

SAS/QC® 14.2 User's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS/QC® 14.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/QC® 14.2 User's Guide

Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

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

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

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2016

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

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

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to http://support.sas.com/thirdpartylicenses.

Contents

Chapter 1. What's New in SAS/QC 14.2 . . . 1
Chapter 2. Using This Book . . . 3
Chapter 3. Components of SAS/QC Software . . . 7
Chapter 4. SAS/QC Graphics . . . 19
Chapter 5. The ANOM Procedure . . . 35
Chapter 6. The CAPABILITY Procedure . . . 189
Chapter 7. The CUSUM Procedure . . . 547
Chapter 8. The FACTEX Procedure . . . 617
Chapter 9. The ISHIKAWA Procedure . . . 703
Chapter 10. The MACONTROL Procedure . . . 787
Chapter 11. Introduction to Multivariate Process Monitoring Procedures . . . 897
Chapter 12. The MVPDIAGNOSE Procedure . . . 901
Chapter 13. The MVPMODEL Procedure . . . 925
Chapter 14. The MVPMONITOR Procedure . . . 957
Chapter 15. The OPTEX Procedure . . . 997
Chapter 16. The PARETO Procedure . . . 1067
Chapter 17. The RAREEVENTS Procedure . . . 1169
Chapter 18. The RELIABILITY Procedure . . . 1203
Chapter 19. The SHEWHART Procedure . . . 1395
Appendix A. Measurement Systems Analysis . . . 2193
Appendix B. The RELIABILITY Graphical Interface . . . 2209
Appendix C. Functions . . . 2215
Appendix D. Special Fonts in SAS/QC Software . . . 2241

Subject Index **2245**

Syntax Index **2271**

Credits and Acknowledgments

Credits

Documentation

Writing	Michael J. Cybrynski, Bobby Gutierrez, Gordon Johnston, Julie LaBarr, Sharad S. Prabhu, Bucky Ransdell, Robert N. Rodriguez, Elizabeth Shamseldin, Randall D. Tobias
Editing	Anne Baxter, Virginia Clark, Ed Huddleston, Sharad S. Prabhu, Robert N. Rodriguez, Donna Sawyer, Julie Simmons
Document Management and Production	Tim Arnold

Software

The procedures in SAS/QC software were implemented by the Statistical Quality Improvement Research and Development Department. Substantial support was given to the project by other members of the Advanced Analytics Division. The Core Development Division, Display Products Division, Graphics Division, and Host Systems Division also contributed to this product.

ANOM	Michael J. Cybrynski, Bucky Ransdell
CAPABILITY	Bucky Ransdell
CUSUM	Michael J. Cybrynski, Bucky Ransdell
FACTEX	Randall D. Tobias
ISHIKAWA	Michael J. Cybrynski
MACONTROL	Michael J. Cybrynski, Bucky Ransdell
MVPDIAGNOSE	Bucky Ransdell
MVPMODEL	Blair Christian, Bucky Ransdell
MVPMONITOR	Bucky Ransdell
OPTEX	Randall D. Tobias
PARETO	Bucky Ransdell
RAREEVENTS	Bucky Ransdell
RELIABILITY	Gordon Johnston, Bucky Ransdell
SHEWHART	Michael J. Cybrynski, Bucky Ransdell

Support Groups

Testing	Jack J. Berry, Brett Chapman, Jeanne Martin Portia Parker, Fouad Younan
Technical Support	Elizabeth Edwards, Kathleen Kiernan, Paul Savarese

Acknowledgments

Many people have been instrumental in the development of SAS/QC software. The individuals acknowledged here have been especially helpful. The organizations listed represent these individuals' affiliations when they made their most significant or most recent contributions.

Melvin T. Alexander	Westinghouse Electric Corporation
Kevin Anderson	Motorola Inc.
Robert V. Baxley	Monsanto Company
Linda W. Blazek	Alcoa Laboratories
James L. Bossert	Eastman Kodak Company
Mike Boyko	Singer Link Flight Simulation Division
Bob Chiverton	G. E. Silicone Products Business Division
Michael L. Cuenco	Kaiser Permanente
Sharon C. Dodson	Kellogg Company
Necip Doganaksoy	General Electric Corporate Research and Development
Melissa Durfee	Wyman-Gordon Company
Luis Escobar	Louisiana State University
Leslie Fowler	Motorola Inc.
Kevin Franklin	Lockheed Aeronautical Systems Company
Paul Hamilton	Boeing
Chris Handorf	Motorola Inc.
Homer Hegedus	Motorola Inc.
Bill Henley	Chrysler Huntsville Electronics Division
Jason C. Hsu	The Ohio State University
Norio Irikura	Nippondenso Co., Ltd.
Bill Kahn	W. L. Gore & Associates
Doug Matlock	Motorola Inc.
William Meeker	Iowa State University
John Mikolaj	Union Carbide
Peter R. Nelson	Clemson University
Wayne Nelson	Consultant
Yasuo Ohashi	University of Tokyo
Joe Perry	Boeing
José Ramirez	W. L. Gore & Associates

Rod Reish	G. E. Silicone Products Business Division
James Sattler	Syntex Research
Robert J. Scharl	LTV Steel Company
Suzanne Scott	Texas Instruments
Mark A. Soboslai	Wheeling-Pittsburgh Steel Corporation
Jan van Schaik	The Upjohn Company
John H. Sheesley	Air Products and Chemicals, Inc.
Wayne E. Stevenson	Dow Corning Corporation
Pat Sullivan	Cameron Iron Works, Inc.
Bob Teasley	Bethlehem Steel Corporation
H. C. M. van der Knaap	Unilever Research Laboratory
Lonnie C. Vance	General Motors Corporation
Teresa Vincel	Bethlehem Steel Corporation
Philip Whittall	Unilever Research Laboratory
Joe Wolkan	General Motors Corporation
Akira Yagi	Takenaka Komuten Co., Ltd.
Kiichiro Yamamura	Japan Air Lines Co., Ltd. (retired)
Jürgen Zeindl	voestalpine Stahl GmbH

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

Chapter 1
What's New in SAS/QC 14.2

Overview

SAS/QC 14.2 includes enhancements to the ANOM, CAPABILITY, CUSUM, MACONTROL, RAREVENTS, and SHEWHART procedures.

ANOM, CUSUM, MACONTROL, and SHEWHART Enhancements

The chart statements in the ANOM, CUSUM, MACONTROL, and SHEWHART procedures support the following new options that enable you to control the markers that are used to plot subgroups in ODS Graphics output when you specify a *symbol-variable*: MARKERDISPLAY=, MARKERDISPLAY2=, MARKERLABEL=, MARKERLABEL2=, and MARKERMISSINGGROUP.

CAPABILITY Enhancements

The new FORCEQN and FORCESN options in the PROC CAPABILITY statement force the calculation of the robust estimates of scale Q_n and S_n. By default, these statistics are not computed for large data sets because the calculations can be very time-consuming.

RAREEVENTS Enhancements

The RAREEVENTS procedure is production in this release and includes the following enhancements:

- The DIST= option now supports the Weibull distribution.
- The COMPARE statement supports the new PROCESS= and REFERENCE= options, which provide different ways to display the process data and the reference probability distribution in a comparison plot.
- When a continuous distribution is in effect, the COMPARE statement now produces a table of goodness-of-fit statistics that are based on the empirical distribution function (EDF).

Chapter 2
Using This Book

Overview

The *SAS/QC User's Guide* provides complete documentation, including introductory examples, syntax, computational details, and advanced examples for the procedures in SAS/QC 14.2. In general, this book can be used for all current releases of SAS/QC software, and it replaces and updates the information provided by *SAS/QC 14.1 User's Guide*.

Point-and-click interfaces for basic statistical quality improvement methods and design of experiments are also included in SAS/QC software. The SQC Menu System for statistical quality control applications is described in *SAS/QC Software: SQC Menu System, Version 6, First Edition*. The ADX Interface for the design and analysis of experiments is described in *Getting Started with the SAS ADX Interface for Design of Experiments*.

NOTE: For releases beginning with SAS/QC 12.1 you must enter the follow statements before invoking the SQC Menu System to ensure its proper operation:

```
ods graphics off;
ods html close;
ods listing;
```

Organization

This book is organized as follows.

Chapter 1, "What's New in SAS/QC 14.2," provides information about the changes and enhancements to SAS/QC software in SAS/QC 14.2.

Chapter 3, "Components of SAS/QC Software," gives an overview of the tools provided by SAS/QC software and their uses.

The majority of SAS/QC procedures produce graphs as an important part of their output. Chapter 4, "SAS/QC Graphics," describes the different approaches available for producing this graphical output.

Each of the remaining chapters describes one SAS/QC procedure. These chapters appear in alphabetical order by procedure name. The following list summarizes the types of information provided for each procedure:

Overview provides a general description of what the procedure does.

Getting Started illustrates simple uses of the procedure using tutorial examples.

Syntax constitutes the major reference section for the syntax of the procedure. First, the statement syntax is summarized. Next, functional summary tables list the options classified by function. Finally, a dictionary of options, listed in alphabetical order, provides details on each option.

Details describes features of the procedure, including equations, computational methods, and input and output data sets.

Examples provides examples that illustrate common and advanced applications of the procedure.

References lists books and journal articles relevant to the procedure.

Several of the SAS/QC procedures are quite large and support several statements that are described independently within the procedure chapter. For example, the chapter describing the CAPABILITY procedure contains a major section for each plot statement (such as the HISTOGRAM statement) supported by the procedure. Each plot statement section contains its own "Overview," "Getting Started," "Syntax," "Details," and "Examples" subsections.

Typographical Conventions

SAS/QC User's Guide uses various type styles, as explained in the following list:

roman	is the standard type style used for most text.
UPPERCASE ROMAN	is used for SAS statements, options, and other SAS language elements when they appear in the text. However, you can enter these elements in your own SAS programs in lowercase, uppercase, or a mixture of the two.
UPPERCASE BOLD	is used in the "Syntax" sections' initial lists of SAS statements and options.
oblique	is used for user-supplied values for options in the syntax definitions. In the text, these values are written in *italic*.
helvetica	is used for the names of variables and data sets when they appear in the text.
bold	is used to refer to matrices and vectors.
italic	is used for terms that are defined in the text, for emphasis, and for references to publications.
`monospace`	is used for example code. In most cases, this book uses lowercase type for SAS code.

Conventions for Examples

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

```
options linesize=80 pagesize=200 nonumber nodate;
```

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

```
ods html style=htmlblue;
```

See Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*), for more information about styles.

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

The following GOPTIONS statement is used to create traditional graphics output (see Chapter 4, "SAS/QC Graphics").

```
filename GSASFILE 'file-specification';
goptions gsfname = GSASFILE
         gsfmode = replace
         fileonly
         dev     = png
         htext   = 2.6pct
         htitle  = 3.5pct
         hsize   = 6.4in
         border
         horigin = 0in
         vorigin = 0in ;
```

Accessing the SAS/QC Sample Library

The SAS/QC sample library includes many examples that illustrate the use of SAS/QC software, including the examples used in this documentation. To access these sample programs, select the **Help** pull-down menu and then select **SAS Help and Documentation**. From the **Contents** list, choose **Learning to Use SAS** and **Sample SAS Programs**. Choose **SAS/QC** to bring up a list of sample programs.

Chapter 3
Components of SAS/QC Software

Contents

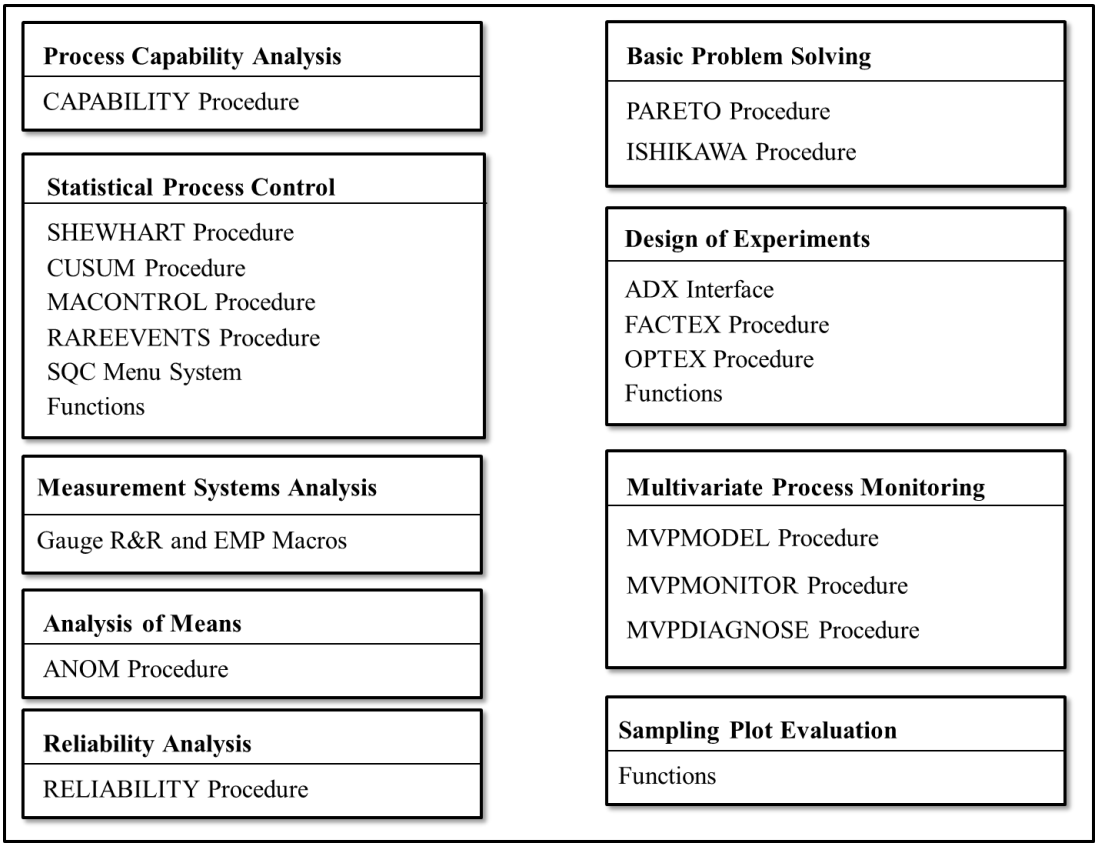
Overview 7
ADX Interface for Design of Experiments 9
SQC Menu System for Statistical Quality Control 10
Procedures for Design of Experiments 11
Procedures for Control Chart Analysis 12
Procedure for Process Capability Analysis 13
Procedures for Basic Quality Problem Solving 14
Procedure for Reliability Analysis 15
Procedure for Analysis of Means 16
Procedures for Multivariate Process Monitoring 17

Overview

SAS/QC software, a component of the SAS System, provides a comprehensive set of tools for statistical quality improvement. You can use these tools to

- organize quality improvement efforts
- design industrial experiments for product and process improvement
- apply Taguchi methods for quality engineering
- establish statistical control of a process
- maintain statistical control and reduce variation
- analyze process capability
- develop and evaluate acceptance sampling plans

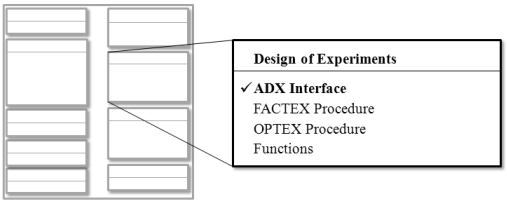
Figure 3.1 Components of SAS/QC Software



There are two types of tools in SAS/QC software: interfaces and procedures.

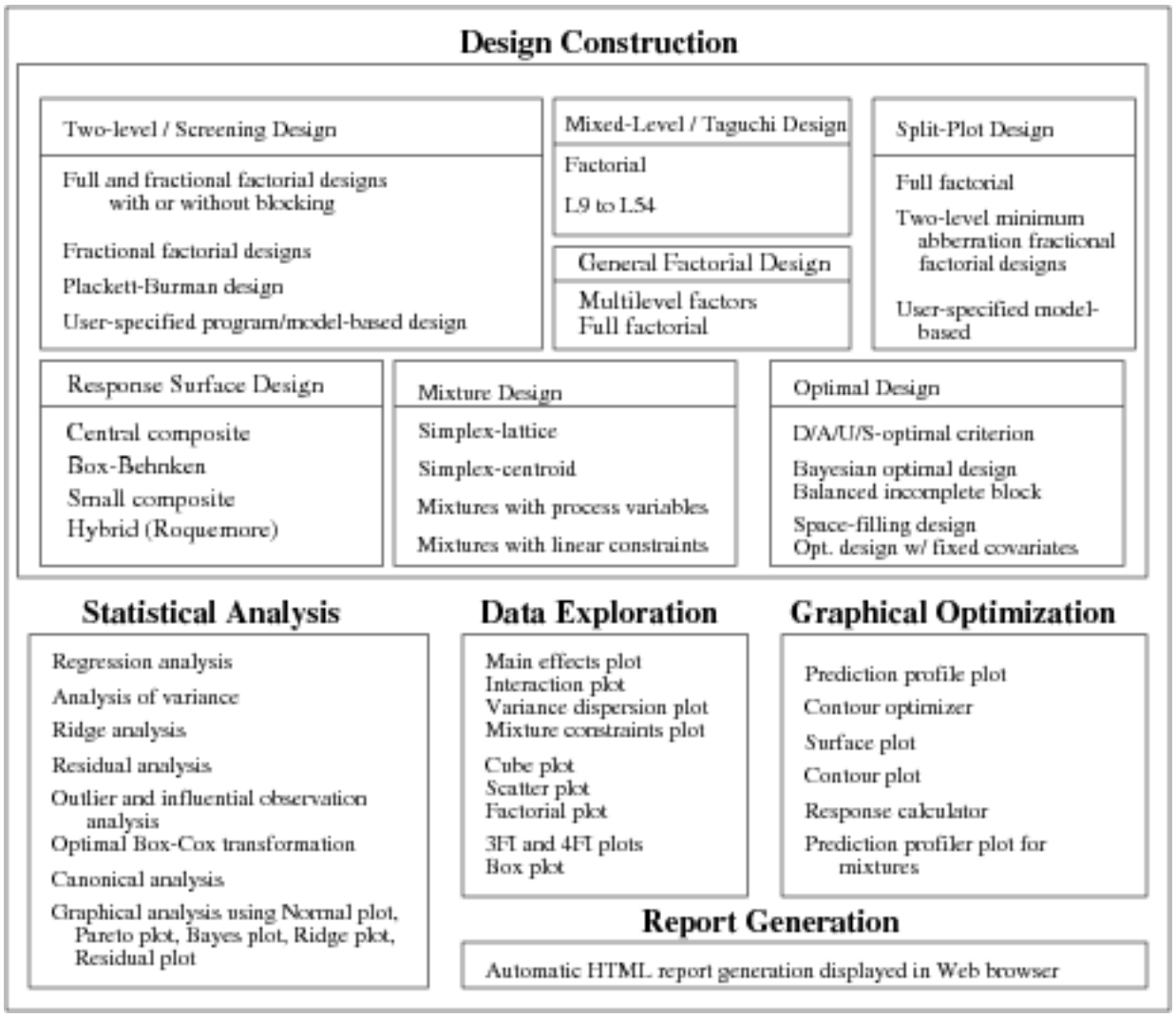
- The interfaces are complete, full-screen-oriented environments for statistical quality-improvement applications. Unlike with the procedures, using the interfaces requires no knowledge of SAS programming. These include the SQC menu system and the ADX interfaces for statistical quality-control applications.
- The procedures in SAS/QC software offer greater flexibility and power than the interface. To use a procedure, you must have a basic knowledge of the SAS language and the syntax of the procedure.

ADX Interface for Design of Experiments



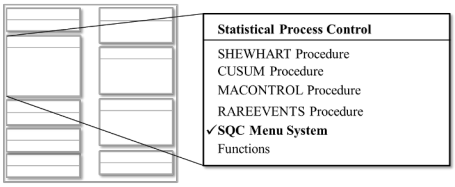
The ADX Interface provides a solution for engineers and researchers who require a point-and-click interface for designing and analyzing experimental designs.

Figure 3.2 General Design and Analysis Facilities



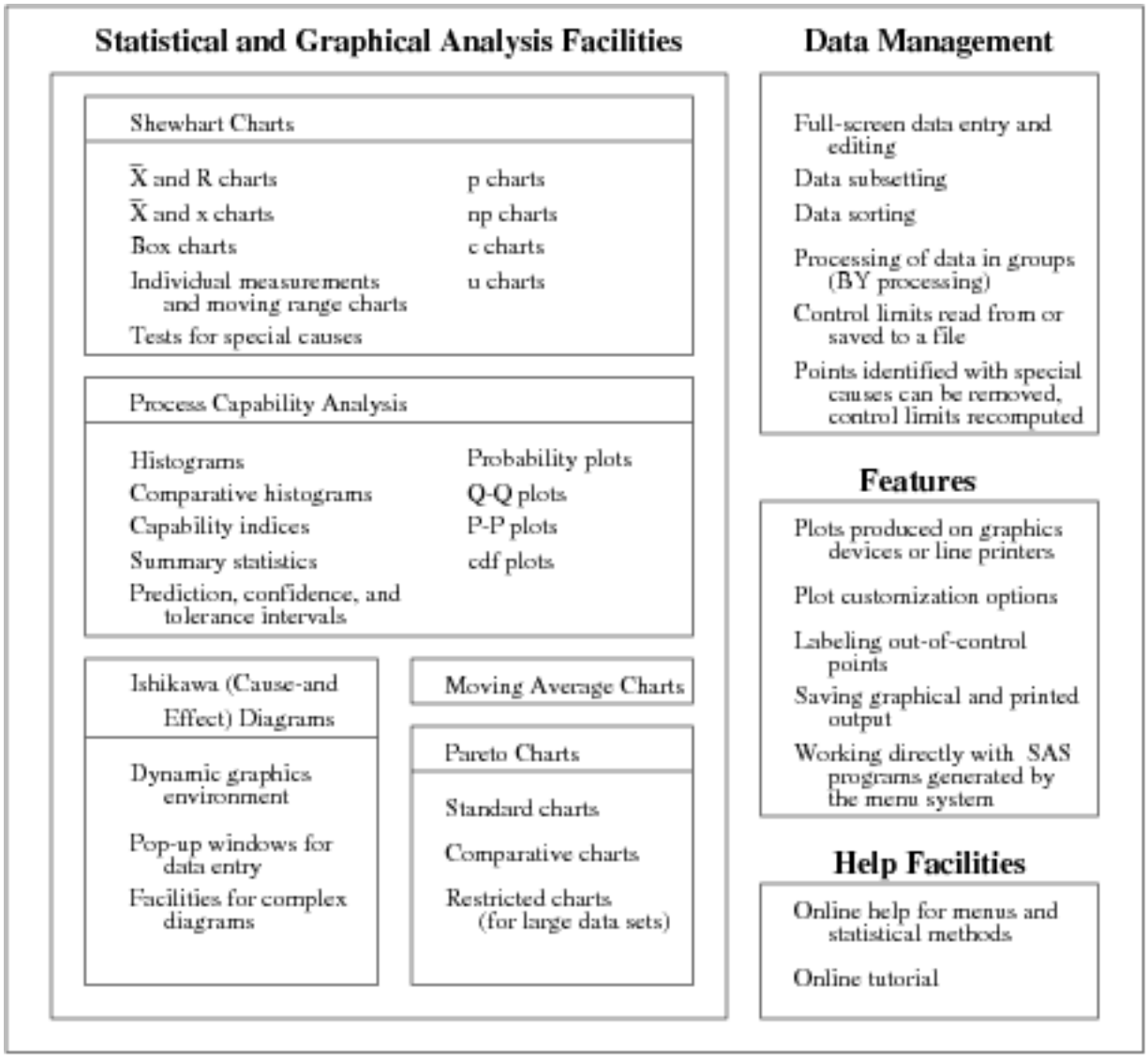
NOTE: For more information about the ADX Interface, see *Getting Started with the SAS ADX Interface for Design of Experiments*

SQC Menu System for Statistical Quality Control



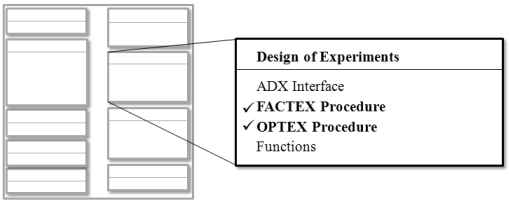
The SQC Menu System provides facilities for standard statistical quality-control applications and is intended for quality analysts and quality-control managers, rather than for statisticians.

Figure 3.3 Overview of the SQC Menu System



NOTE: The SQC Menu System is documented in *SAS/QC Software: SQC Menu System for Quality Improvement, Version 6, Second Edition*.

Procedures for Design of Experiments



The FACTEX procedure constructs factorial experimental designs, which are useful for studying the effects of various factors on a response. The OPTEX procedure searches for optimal designs in situations in which standard designs are not available.

Figure 3.4 Overview of the Experimental Design Procedures

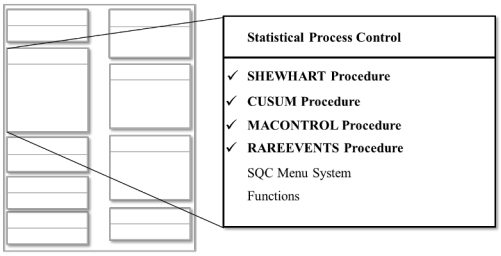
FACTEX Procedure

Overview	Features
Constructs orthogonally confounded full and fractional factorial designs, with and without blocking, including 2^k orthogonal array designs Constructs q^k factorial designs, where q is prime. These can be used with the SAS DATA step to construct many complex desgins, such as mixed-level factorial and balanced incomplete block designs.	No limitation on the number of factors or factor levels Approximate confounding rules are given for model constraints Designs can be chosen to have minimum size, maximum resolution, maximum block size, minimum aberration

OPTEX Procedure

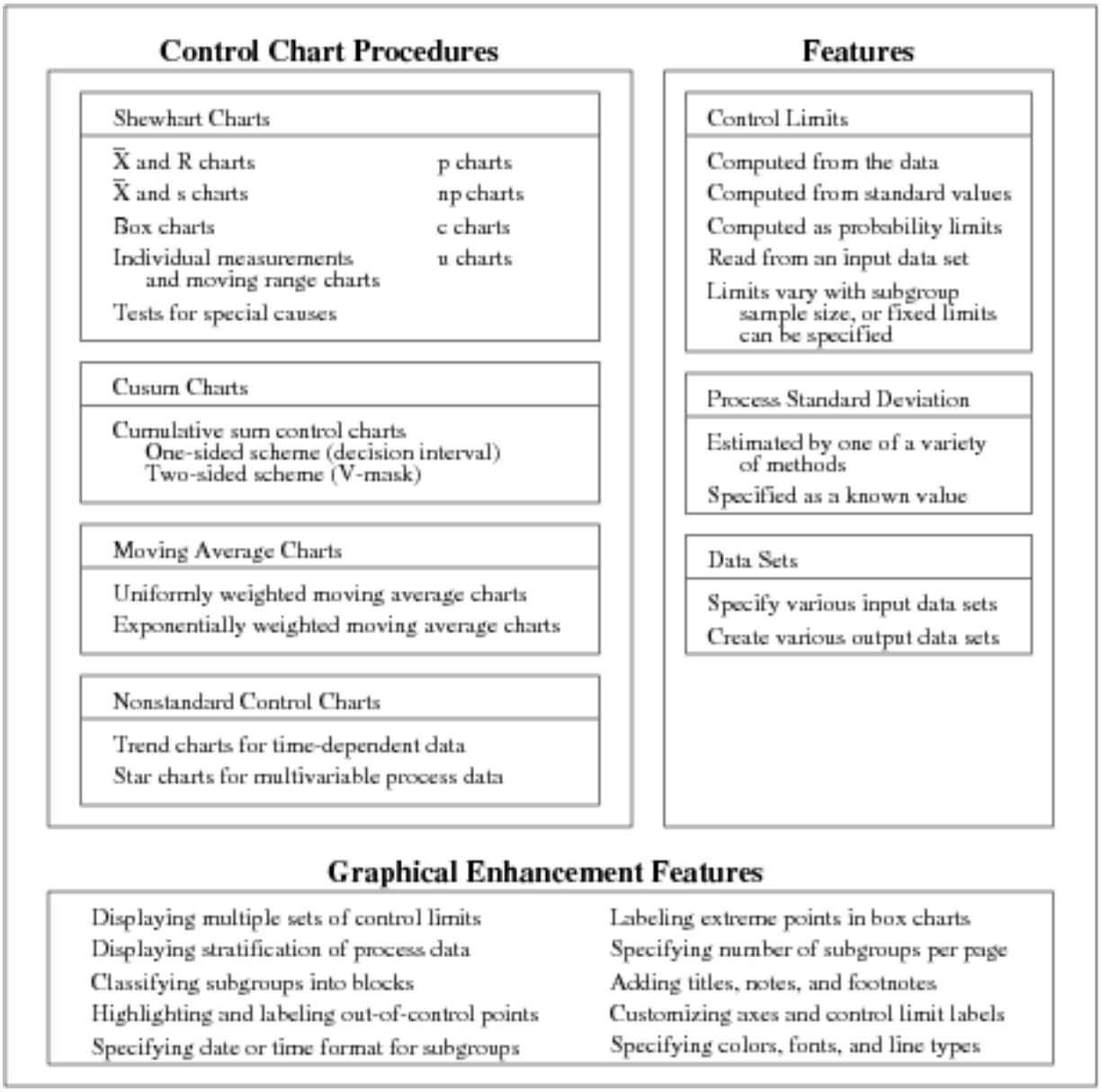
Overview	Features
Searches for D-optimal and A-optimal designs Intended for situations in which a standard design is not appropriate. For example, Not all combinations of factor levels are feasible Resources restrict the number of runs The experimentation region is irregular The model is nonlinear or nonstandard linear	Add points to an existing design Augment preexisting covariates Evaluate an existing design Improve an initial design Various search algorithms and selection criteria are available Flexible GLM-type models are available

Procedures for Control Chart Analysis

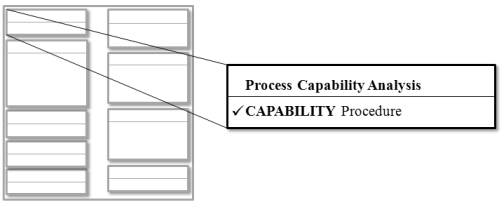


The SHEWHART procedure creates all commonly encountered Shewhart charts for variables and attributes. The CUSUM procedure creates cumulative sum control charts. The MACONTROL procedure creates moving average charts. The RAREEVENTS procedure creates specialized control charts for rare events.

Figure 3.5 Overview of Control Chart Analysis Procedures



Procedure for Process Capability Analysis



The CAPABILITY procedure compares the distribution of output from an in-control process to the specification limits of the process to determine the consistency with which the specification limits can be met.

Figure 3.6 Overview of Process Capability Analysis Procedure

CAPABILITY Procedure

Overview

Provides graphical and statistical tools for process capability analysis

Statistical Methods

Descriptive Statistics

Capability Indices: Cp, Cpk, CPL, CPU, Cpm, k

Confidence, tolerance, and prediction intervals

Goodness-of-fit based on ecdf

Estimation of areas outside specification limits

Other Graphical Features

Inset boxes with summary statistics

Customized axes and legends

Titles, footnotes, notes

Colors, fonts, symbol markers

Graphical Methods

Histograms, superimposed with
- specification limits
- fitted pdf curves
- kernel density estimates
- summary statistics

Comparative histograms, superimposed with
- specification limits
- fitted pdf curves
- kernel density estimates
- summary statistics

cdf plots, superimposed with
- specification limits
- fitted theoretical distribution curves

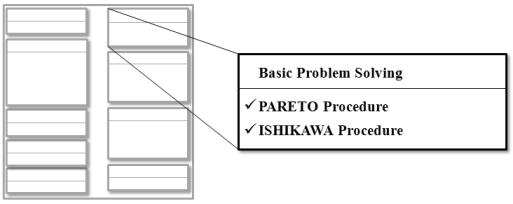
Probability plots, superimposed with
- specification limits

Quantile-quantile (Q-Q) plots, superimposed with specification limits

Probability-probability (P-P) plots

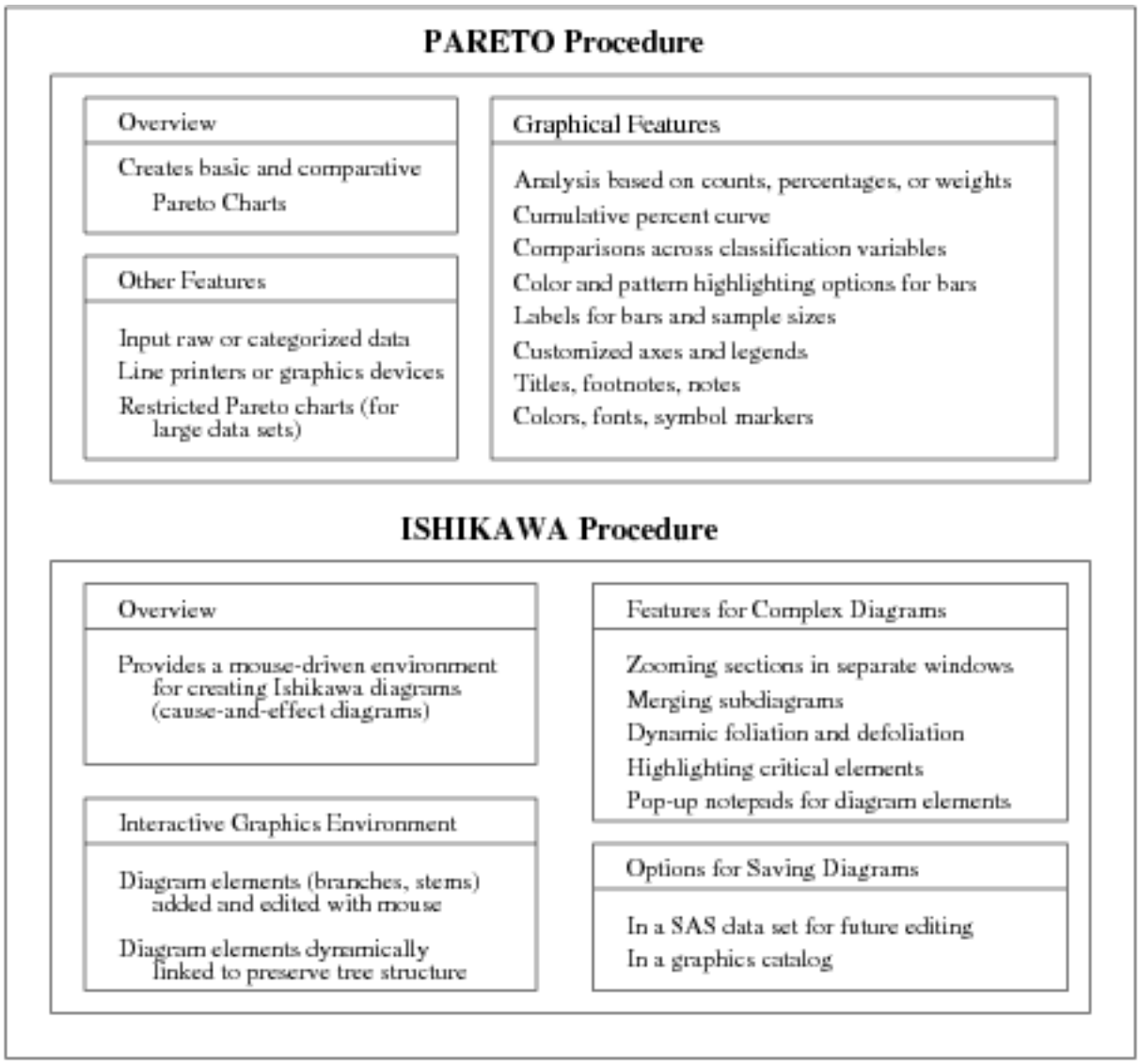
Available distributions
- normal
- lognormal
- exponential
- Weibull
- gamma
- beta

Procedures for Basic Quality Problem Solving

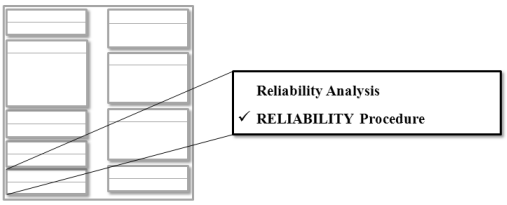


The PARETO procedure creates charts that display the relative frequency of problems in a process or operation. The ISHIKAWA procedure creates a cause-and-effect or fishbone diagram, which displays factors that affect a quality characteristic or problem.

Figure 3.7 Overview of Quality Problem-Solving Procedures



Procedure for Reliability Analysis



The RELIABILITY procedure provides tools for reliability and survival data analysis and for recurrence data analysis.

Figure 3.8 Overview of Reliability Analysis Procedure

RELIABILITY Procedure

Overview

Provides graphical and statistical tools for the analysis of reliability data

Statistical Methods

Fitting of lifetime distributions:
- Weibull
- Lognormal
- Loglogistic
- Extreme value
- Normal
- Logistic
- 3 parameter gamma

Right, left, interval censored lifetimes

Parameter estimation by maximum likelihood and least squares

Confidence intervals for distribution parameters, percentiles, survival function

Analysis of Poisson and binomial data

Regression models

Accelerated life test models

Graphical Features

Probability plots for lifetime distributions:
- Kaplan-Meier, median ranks, expected ranks methods for right censored data
- Turnball algorithm for arbitrarily censored data

Available distributions for probability plots:
- Weibull
- Exponential
- Lognormal
- Loglogistic
- Extreme value
- Normal
- Logistic

Lifetime-stress relationship plots for accelerated lifetest data

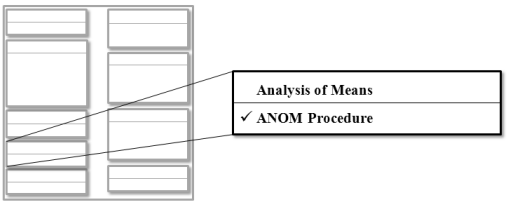
Plots of mean cumulative function and confidence intervals for repairable systems data

Plots for multiple populations

Other Graphical Features

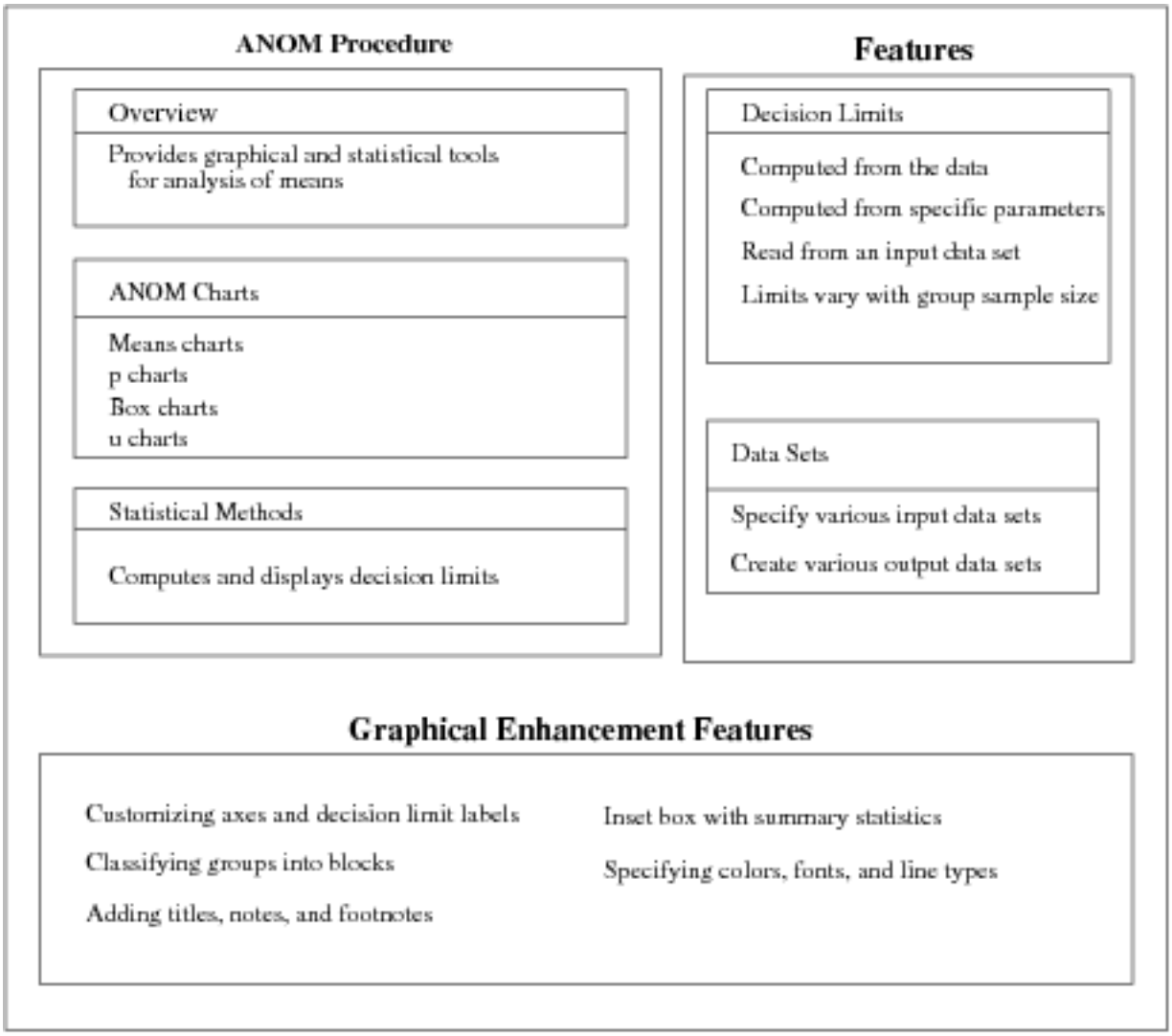
Inset box with summary statistics
Customized axes and legends
Title, footnotes, notes
Colors, fonts, symbol markers

Procedure for Analysis of Means

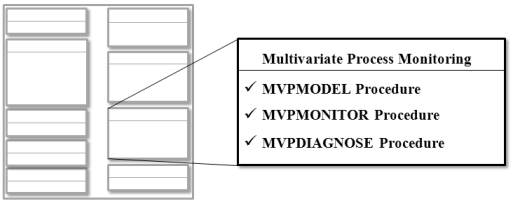


The ANOM procedure provides tools for simultaneously comparing a group of k treatment means with their overall mean at a specified significance level α. The procedure creates ANOM charts for various types of response data, including continuous measurements, proportions, and rates.

Figure 3.9 Overview of Analysis of Means Procedure



Procedures for Multivariate Process Monitoring



The MVPMODEL procedure builds a principal component model from multivariate process data. The MVPMONITOR procedure creates multivariate control charts that are used to detect unusual variation in the process. The MVPDIAGNOSE procedure produces plots that can be used to investigate unusual variation.

Figure 3.10 Overview of Multivariate Process Monitoring Procedures

MVPMODEL Procedure

Builds a principal component model from multivariate process data in which the measured variables are continuous and correlated

Supports cross validation for automatically selecting the number of principal components

Output data sets are inputs for MVPMONITOR and MVPDIAGNOSE

MVPMONITOR Procedure

Produces T^2 and Squared Prediction Error (SPE) control charts that are used to detect unusual variation in the process

T^2 charts detect unusual variation within the model hyperplane

SPE charts detect unusual variation from the model hyperplane

MVPDIAGNOSE Procedure

Produces score plots and contribution plots that can provide insight into unusual variation

Score plots can be displayed in a matrix or individually

Contribution plots can be displayed in a paneled layout or individually

Chapter 4
SAS/QC Graphics

Contents

Overview 19
Which Graphics Approach Should You Use? 20
Traditional Graphics 21
ODS Graphics 26
Legacy Line Printer Displays 32

This chapter describes the alternatives available for producing graphical output with the SAS/QC procedures. The statements you use to produce graphical output and the options you specify to control its appearance are described in the chapters of *SAS/QC User's Guide* devoted to the various procedures.

Overview

The following SAS/QC procedures produce graphical output:

- ANOM
- CAPABILITY
- CUSUM
- ISHIKAWA
- MACONTROL
- MVPDIAGNOSE
- MVPMODEL
- MVPMONITOR
- PARETO
- RAREEVENTS
- RELIABILITY
- SHEWHART

The ISHIKAWA procedure (Chapter 9, "The ISHIKAWA Procedure") provides an interactive environment for creating Ishikawa diagrams. This discussion applies to the other procedures, which generate graphical displays when you specify appropriate statements and options.

SAS/QC procedures can produce two types of graphical output[1]:

- traditional graphics
- ODS Statistical Graphics output

Traditional graphics are saved in a graphics catalogs with entry type GRSEG, and their appearance is controlled by global statements such as the GOPTIONS, AXIS, and SYMBOL statements, which are described in *SAS/GRAPH: Reference*. In addition, SAS/QC procedures provide numerous options for controlling the appearance of traditional graphics. You must have a SAS/GRAPH® license to produce traditional graphics.

Beginning with SAS 9.2, SAS/QC procedures can use ODS Statistical Graphics to create graphs. ODS Statistical Graphics (or ODS Graphics for short) is an extension to the Output Delivery System (ODS). Graphs are produced in standard image file formats (such as PNG) instead of graphics catalogs, and the details of their appearance and layout are controlled by ODS styles and templates rather than global statements and procedure options. Graphical output produced by SAS/QC procedures using ODS Graphics is consistent in appearance with graphical output produced by statistical procedures using ODS Graphics.

When ODS Graphics is enabled (for example, with the ODS GRAPHICS ON statement) SAS/QC procedures produce ODS Graphics output. Otherwise, they produce traditional graphics by default. **NOTE:** The following procedures do not support traditional graphics:

- MVPDIAGNOSE
- MVPMODEL
- MVPMONITOR
- RAREEVENTS

Which Graphics Approach Should You Use?

Beginning with SAS 9.2, SAS/QC procedures can produce ODS Graphics output as an alternative to traditional graphics. Also beginning with SAS 9.2, the default appearance of traditional graphics is determined by the ODS style that is in effect for the ODS destination that you are using. You can prevent the ODS style from affecting the appearance of traditional graphics by specifying the NOGSTYLE system option. Therefore, you have three alternatives for producing graphical output with SAS/QC procedures:

- traditional graphics without ODS styles
- traditional graphics using ODS styles

[1] Some SAS/QC procedures can also produce legacy line printer charts. See "Legacy Line Printer Displays" on page 32.

- ODS Graphics

The appropriate approach depends on your objective, as follows:

- If you are working with a SAS program written prior to SAS 9.2, and your priority is to preserve the appearance of traditional graphics produced with SAS/QC procedures, you should specify the NOGSTYLE system option.
- If you are writing a new SAS program, consider using traditional graphics with ODS style-dependent defaults to take advantage of their improved appearance while retaining control over every detail of your graphs with procedure options.
- If you are writing a new SAS program, consider using ODS Graphics for the highest-quality graphics output and consistency with output from SAS/STAT procedures and other procedures that are enabled to use ODS Graphics.

The next two sections provide more details and examples of these approaches.

Traditional Graphics

The following SAS/QC procedures support traditional graphics:

- ANOM
- CAPABILITY
- CUSUM
- MACONTROL
- PARETO
- RELIABILITY
- SHEWHART

These procedures support global SAS statements (such as GOPTIONS, AXIS, and SYMBOL statements) used to control the appearance of traditional graphics. Each procedure also supports a rich set of options providing detailed control of features specific to its graphs.

Prior to SAS 9.2 the default appearance of SAS/QC graphs was primitive. To produce attractive graphical output, careful selection of colors, fonts, and other attributes specified with global statements and procedure options was required. Beginning with SAS 9.2, the default appearance of traditional graphs is determined by the ODS style that is in effect.

An example taken from the "Getting Started: HISTOGRAM Statement" on page 300 section of Chapter 6, "The CAPABILITY Procedure," demonstrates the alternatives for producing SAS/QC graphics. The following statements create a data set named Trans containing measurements of the thickness of copper plating on 100 printed circuit boards:

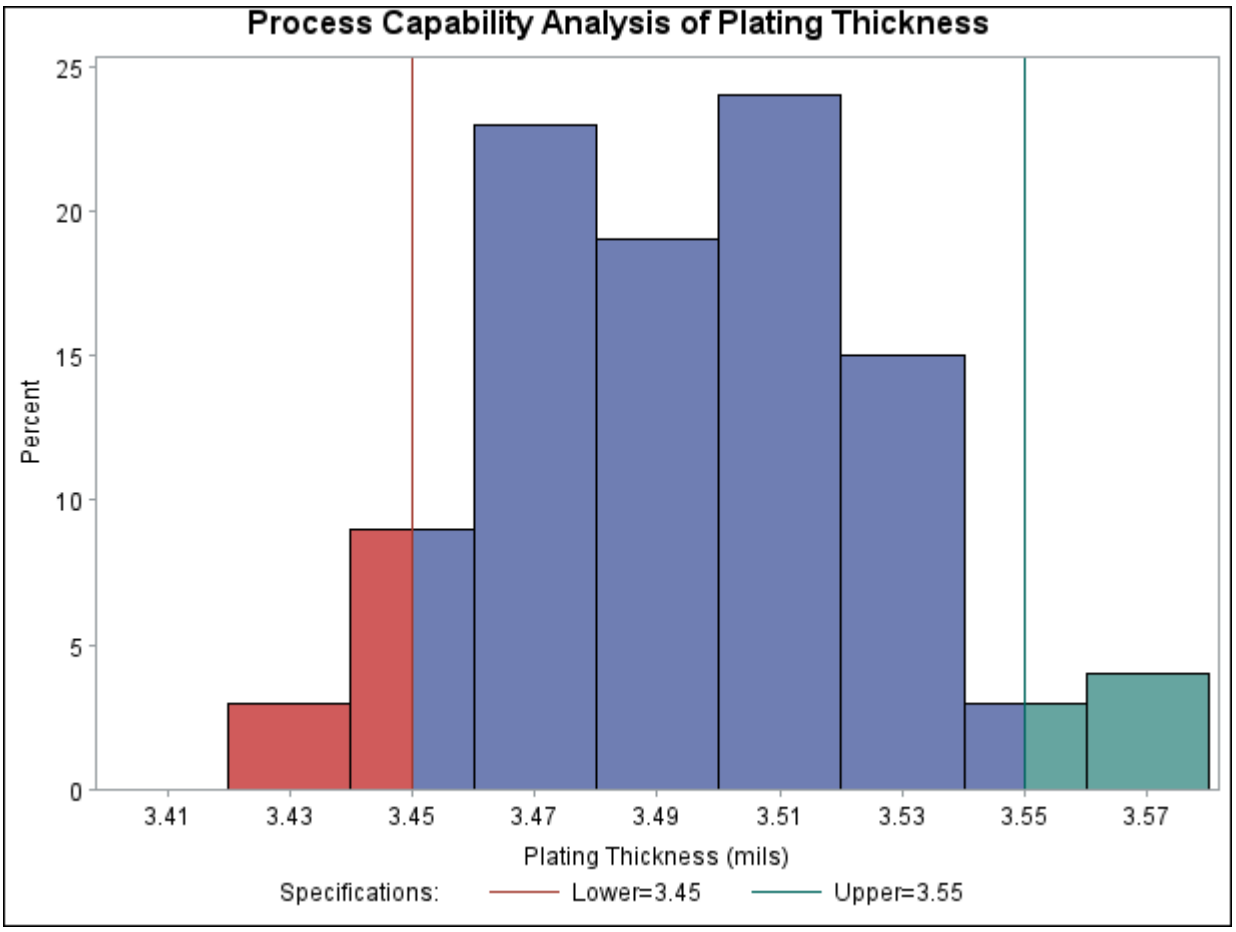
```
data Trans;
   input Thickness @@;
   label Thickness='Plating Thickness (mils)';
   datalines;
3.468 3.428 3.509 3.516 3.461 3.492 3.478 3.556 3.482 3.512
3.490 3.467 3.498 3.519 3.504 3.469 3.497 3.495 3.518 3.523
3.458 3.478 3.443 3.500 3.449 3.525 3.461 3.489 3.514 3.470
3.561 3.506 3.444 3.479 3.524 3.531 3.501 3.495 3.443 3.458
3.481 3.497 3.461 3.513 3.528 3.496 3.533 3.450 3.516 3.476
3.512 3.550 3.441 3.541 3.569 3.531 3.468 3.564 3.522 3.520
3.505 3.523 3.475 3.470 3.457 3.536 3.528 3.477 3.536 3.491
3.510 3.461 3.431 3.502 3.491 3.506 3.439 3.513 3.496 3.539
3.469 3.481 3.515 3.535 3.460 3.575 3.488 3.515 3.484 3.482
3.517 3.483 3.467 3.467 3.502 3.471 3.516 3.474 3.500 3.466
;
```

The following statements produce a histogram of the variable Thickness using traditional graphics whose default appearance is determined by the ODS style.

```
ods graphics off;
title 'Process Capability Analysis of Plating Thickness';
proc capability data=Trans noprint;
   spec lsl=3.45 usl=3.55 cleft cright;
   histogram Thickness;
run;
```

The SPEC statement LSL= and USL= options specify the lower specification limit (LSL) and upper specification limit (USL) for Thickness. The CLEFT and CRIGHT options request that histogram bars (and portions of bars) below the LSL and above the USL be filled with contrasting colors. Figure 4.1 shows the resulting histogram.

Figure 4.1 Traditional Graphics with Default Appearance Determined by ODS Style



The attributes determining the appearance of the graph elements—including text fonts and heights, line styles and thicknesses, and fill colors—all come from the ODS style associated with the output destination. In this case the HTMLBLUE style is in effect.

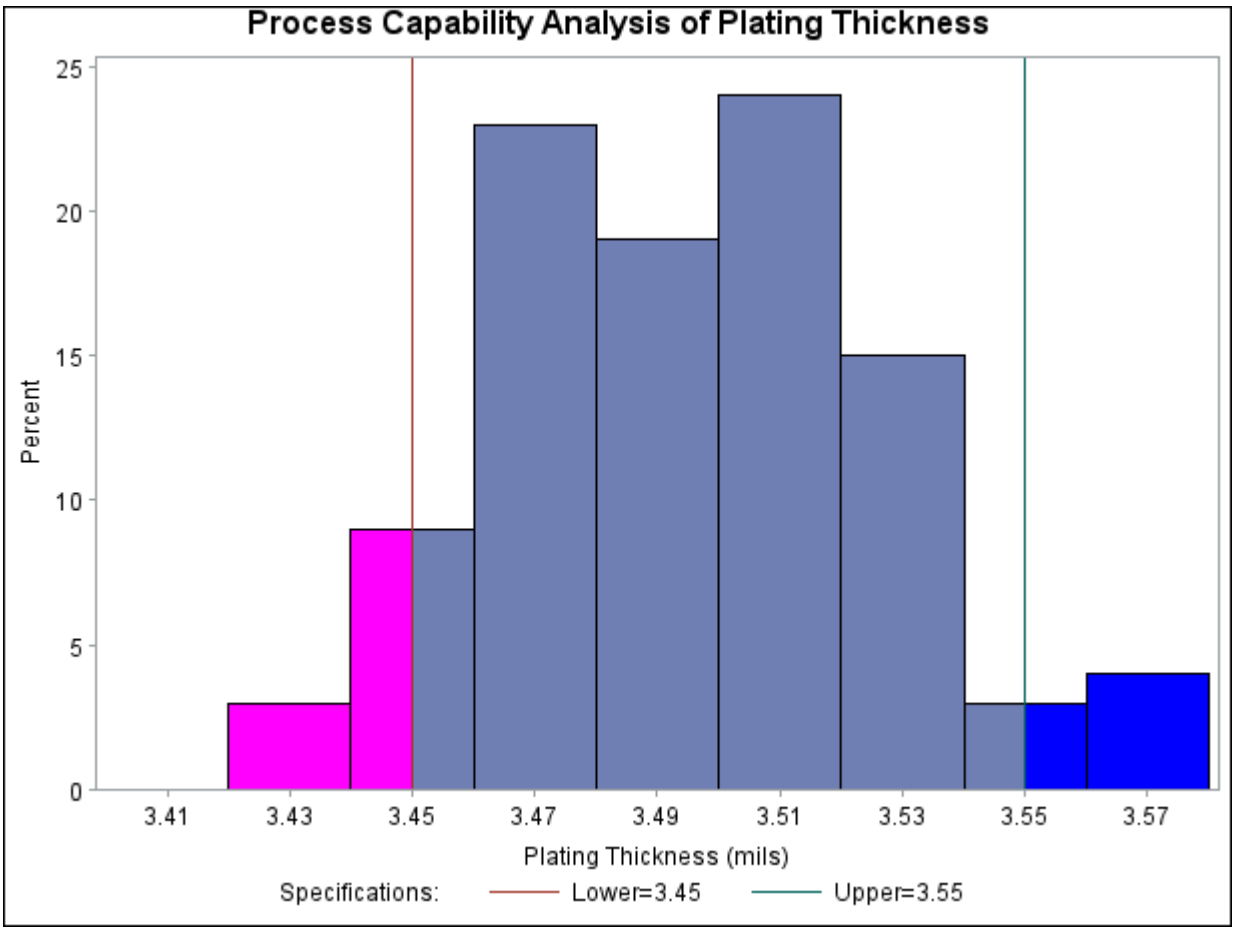
Prior to SAS 9.2, in order to fill histogram bars outside the specification limits with contrasting colors, it was necessary to specify CLEFT= and CRIGHT= colors explicitly. SAS/QC procedures now support options such as CLEFT and CRIGHT that enable optional graph features *without* explicitly specifying colors.

In SAS 9.2, you can still specify colors "on top of" the ODS style, as the following statements demonstrate:

```
proc capability data=Trans noprint;
   spec lsl=3.45 usl=3.55 cleft=magenta cright=blue;
   histogram Thickness;
run;
```

The colors explicitly specified with the CLEFT= and CRIGHT= options replace the CLEFT and CRIGHT colors used in Figure 4.1. The resulting histogram is shown in Figure 4.2.

Figure 4.2 Traditional Graphics Using ODS Style and Appearance Options

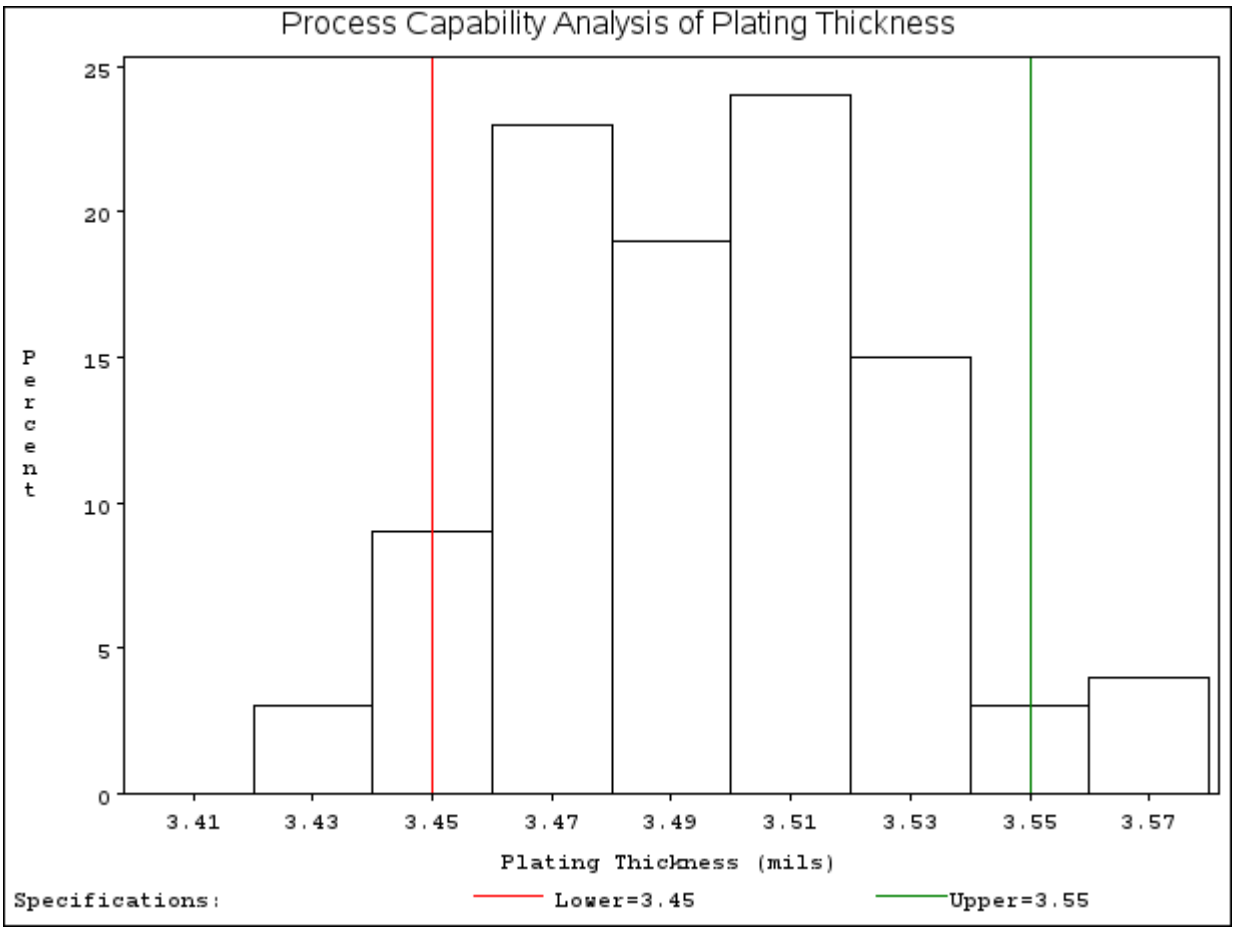


Graphical attributes, such as colors or fonts, specified with global graphics statements or procedure options take precedence over default attributes from the ODS style. To avoid using style-based defaults or to revert to the defaults used prior to SAS 9.2, you can specify the NOGSTYLE system option:

```
options nogstyle;
proc capability data=Trans noprint;
   spec lsl=3.45 usl=3.55;
   histogram Thickness;
run;
```

The appearance of the resulting histogram, shown in Figure 4.3, is very similar to that of a histogram produced by the same CAPABILITY procedure statements in SAS 9.1.

Figure 4.3 Traditional Graphics with NOGSTYLE

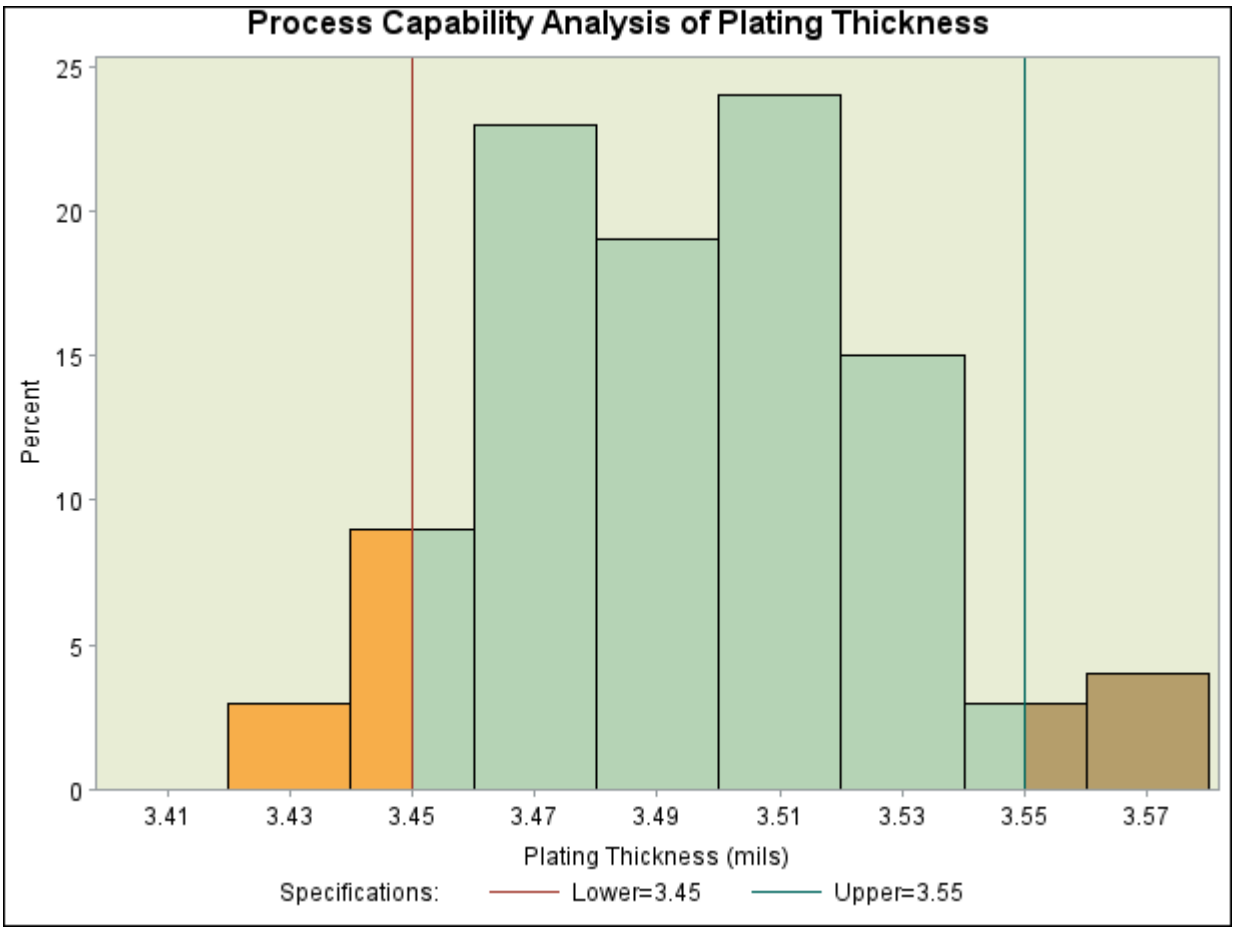


When you specify the NOGSTYLE option, traditional graphics output remains unaffected by ODS styles until the default behavior is restored with the GSTYLE option:

```
options gstyle;
proc capability data=Trans noprint;
   spec lsl=3.45 usl=3.55 cleft=CXf7AE4A cright=CXB59E6B;
   histogram Thickness / cframe = ywh
                         cfill  = CXB5D3B5;
run;
```

The preceding statements produce the histogram shown in Figure 4.4. Here, the default attributes provided by the HTMLBLUE style are overridden by the CLEFT=, CRIGHT=, CFRAME=, and CFILL= options.

Figure 4.4 Traditional Graphics Using ODS Style and Attribute Options



ODS Graphics

The following SAS/QC procedures support ODS Graphics:

- ANOM
- CAPABILITY
- CUSUM
- MACONTROL
- MVPDIAGNOSE
- MVPMODEL
- MVPMONITOR
- PARETO

- RAREEVENTS
- RELIABILITY
- SHEWHART

ODS Graphics provides the highest quality graphical output available from SAS/QC procedures. It is unaffected by global graphics statements, procedure options for controlling traditional graphics, and the GSTYLE system option. See Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*), for a thorough discussion of ODS Graphics.

The following statements produce a histogram of the variable Thickness, which is discussed in the previous section. Here, the ODS GRAPHICS statement is specified to request ODS Graphics.

```
ods listing style=htmlblue;
ods graphics on;
proc capability data=Trans noprint;
   spec lsl = 3.45 usl = 3.55 cleft cright;
   histogram Thickness;
run;
```

Figure 4.5 shows the ODS Graphics version of the histogram. Note that fonts, colors, and other attributes are determined by the HTMLBLUE style. Options for specifying these attributes (as used, for example, in the statements that produced Figure 4.4) are not applicable with ODS Graphics and are ignored when you use the ODS GRAPHICS statement.

Figure 4.5 ODS Graphics Using HTMLBLUE Style

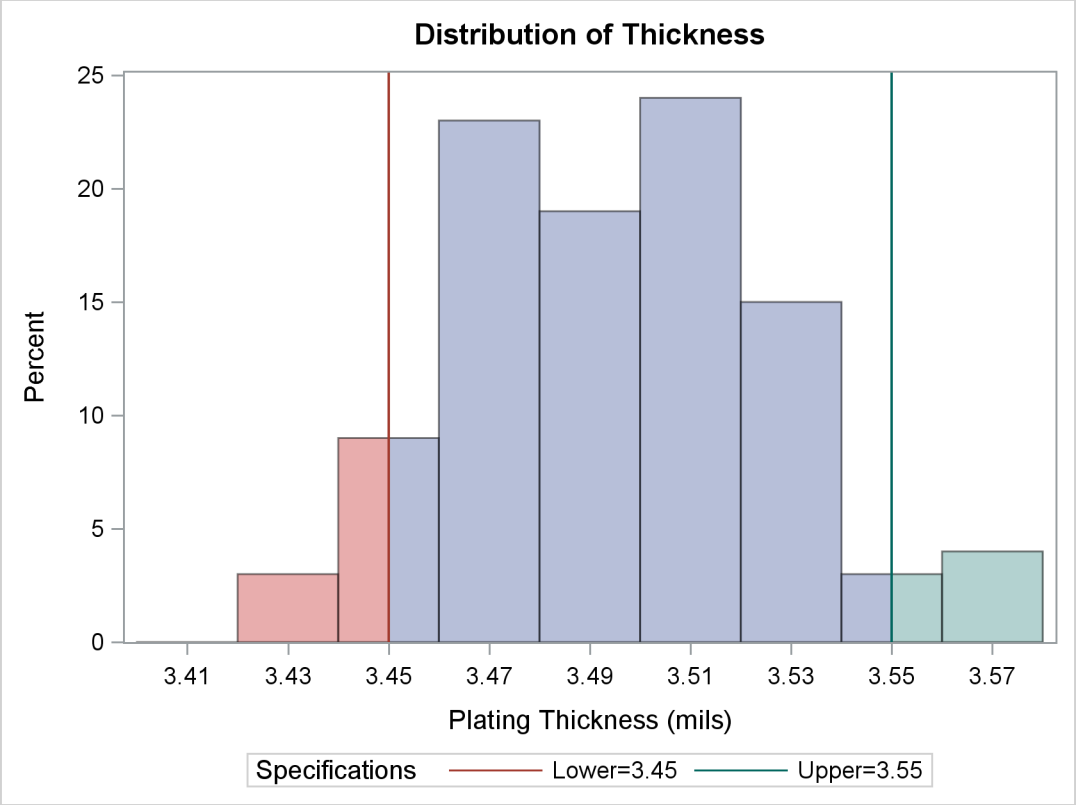


Figure 4.6 through Figure 4.13 show examples of ODS Graphics output produced by SAS/QC procedures. These graphs were all created with the HTMLBLUE style.

Figure 4.6 ANOM Chart

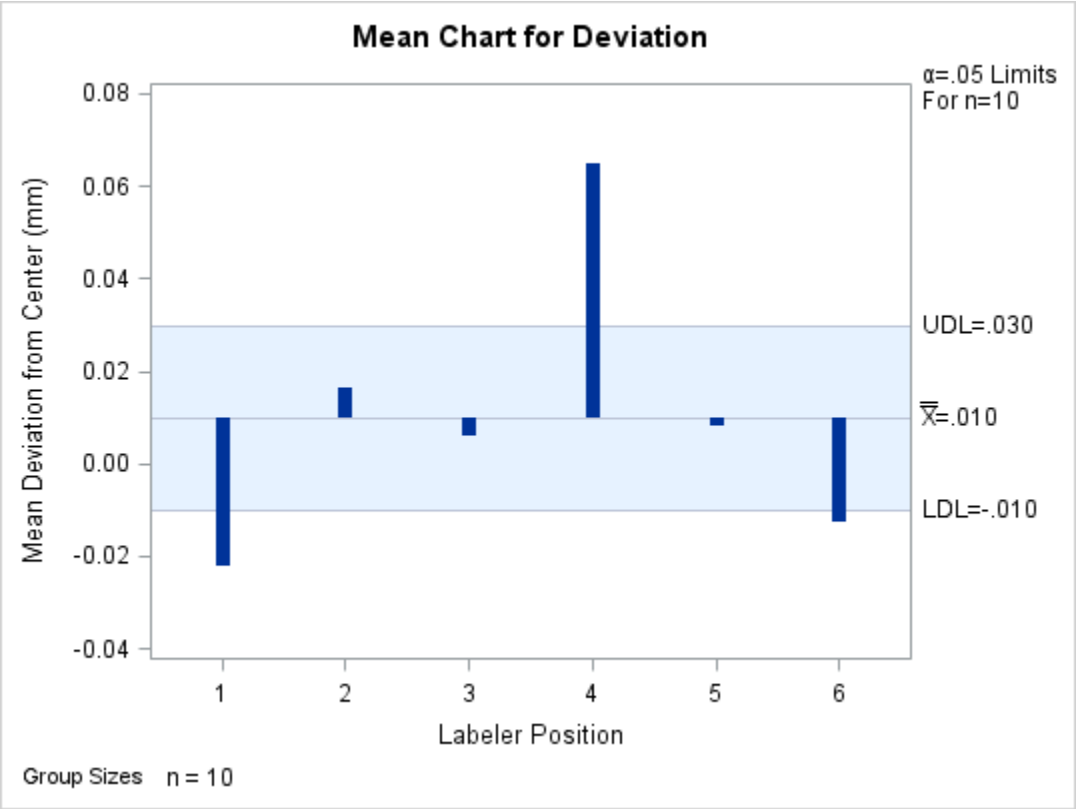


Figure 4.7 Box Chart (SHEWHART)

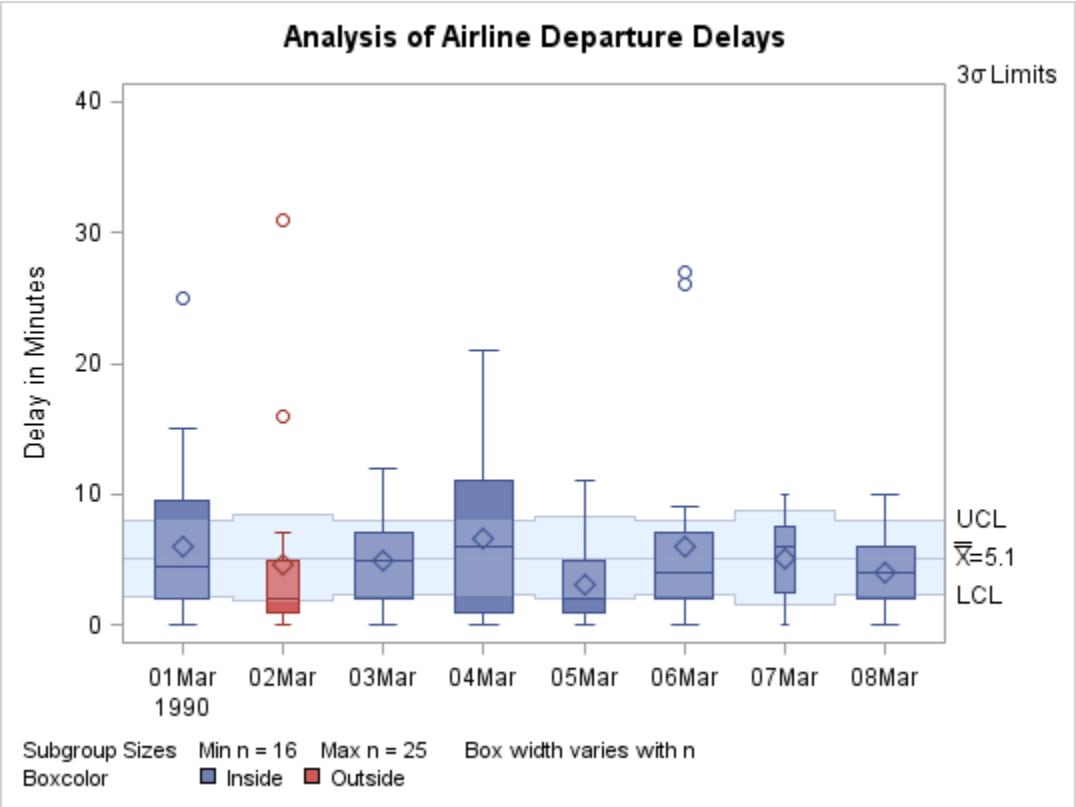


Figure 4.8 Comparative Histogram (CAPABILITY)

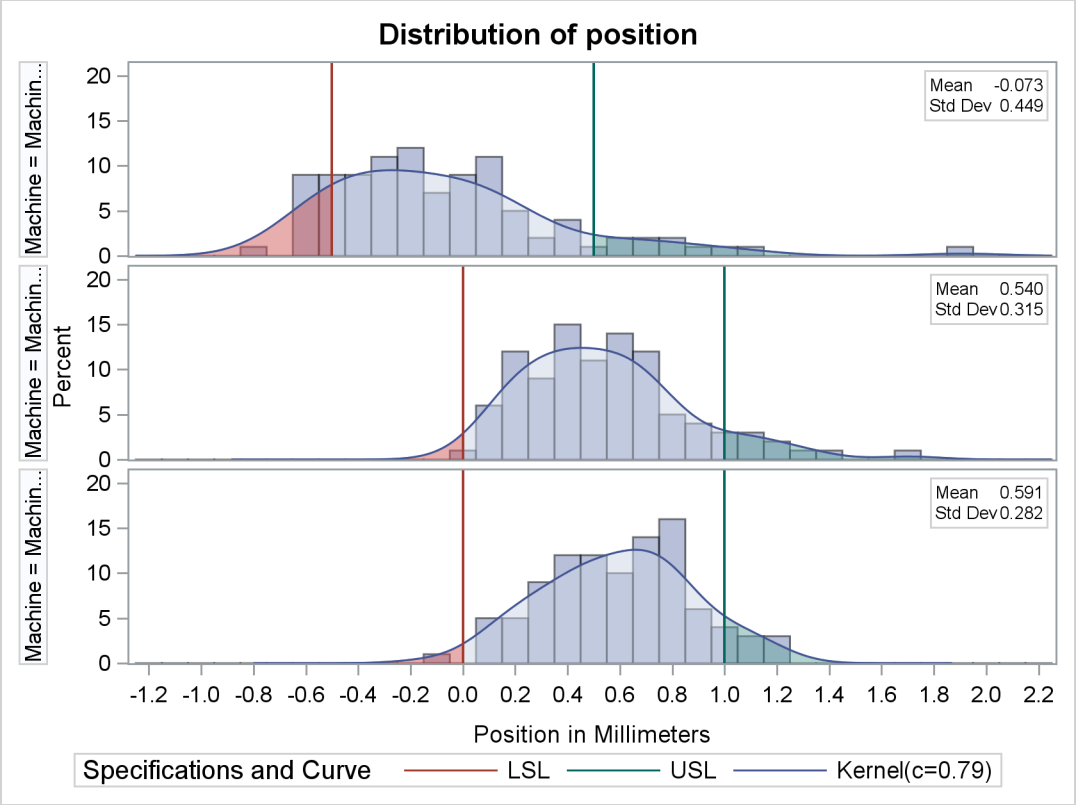


Figure 4.9 Moving Average Chart (MACONTROL)

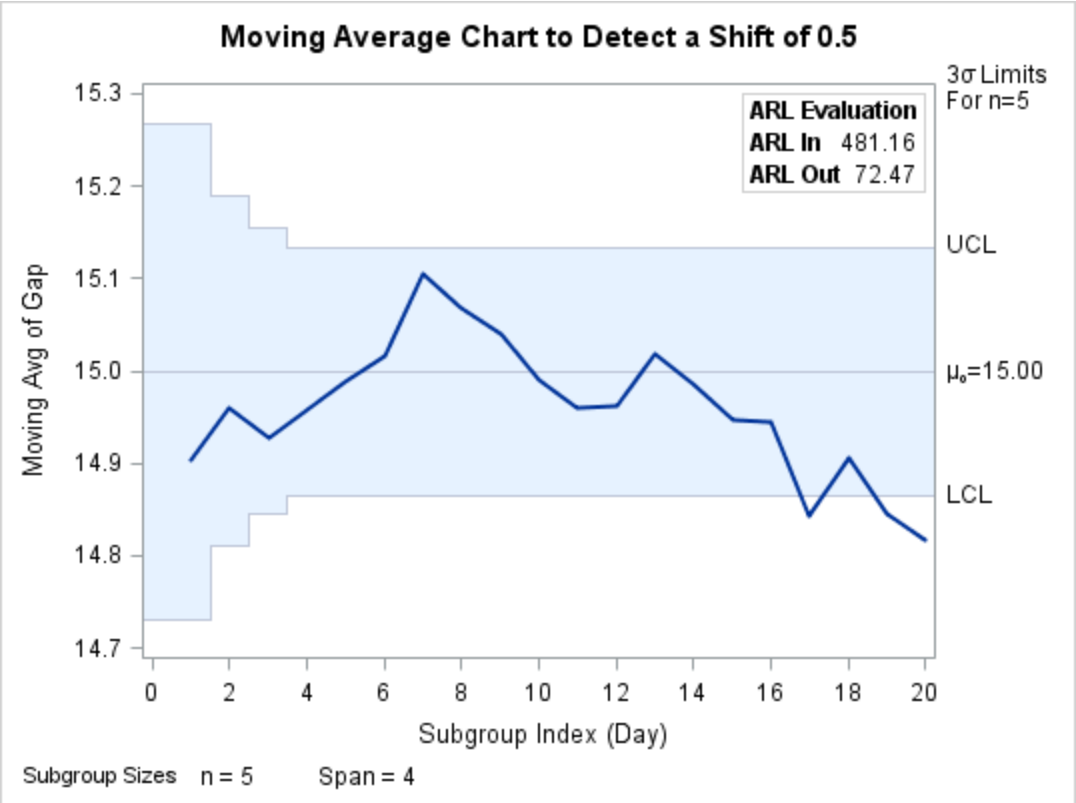


Figure 4.10 Pareto Chart

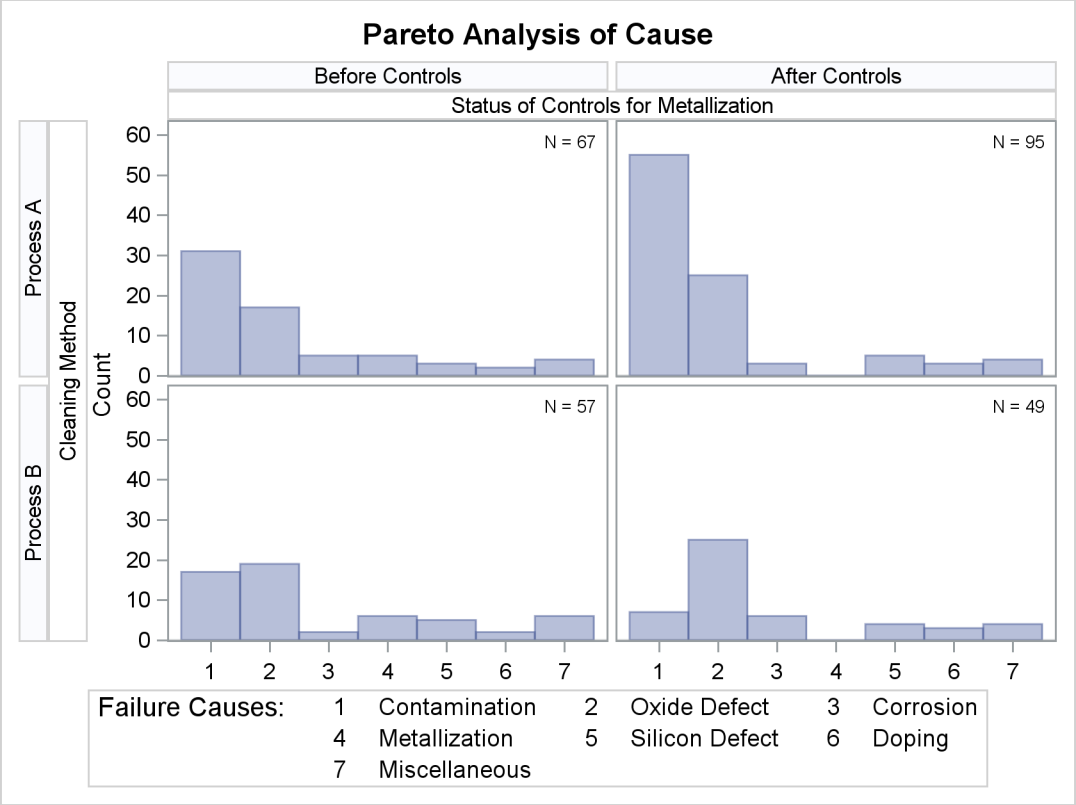


Figure 4.11 Probability Plot (RELIABILITY)

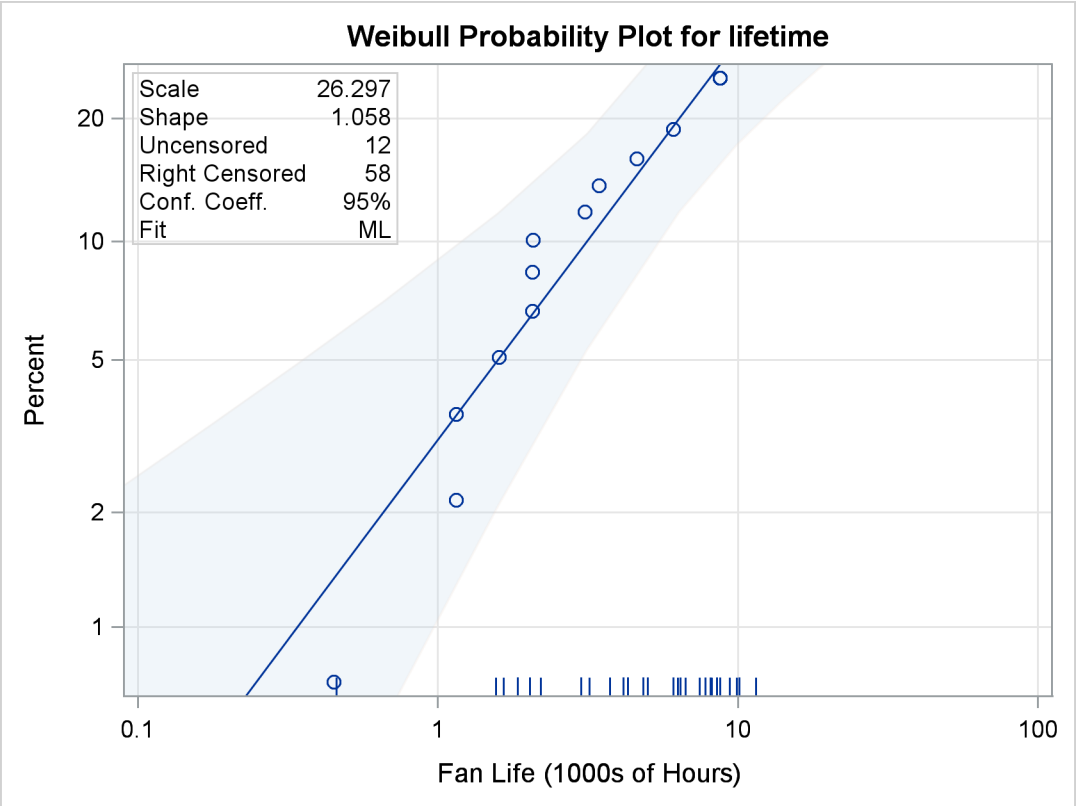


Figure 4.12 Q-Q Plot (CAPABILITY)

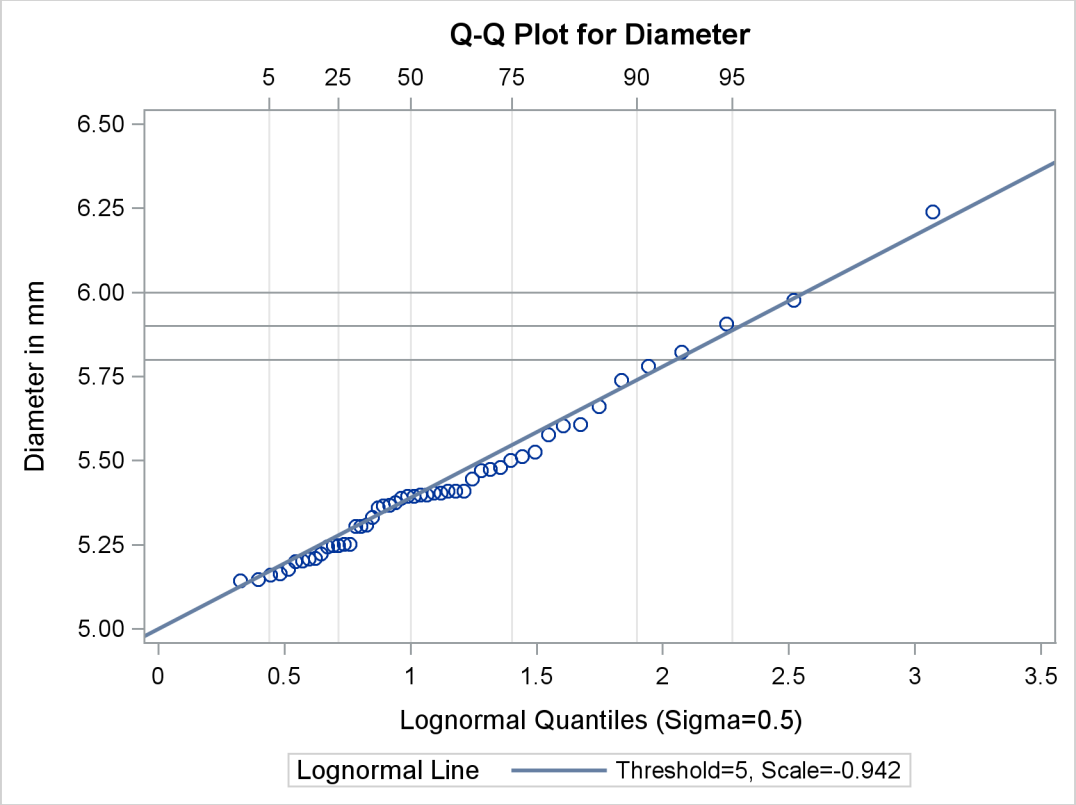
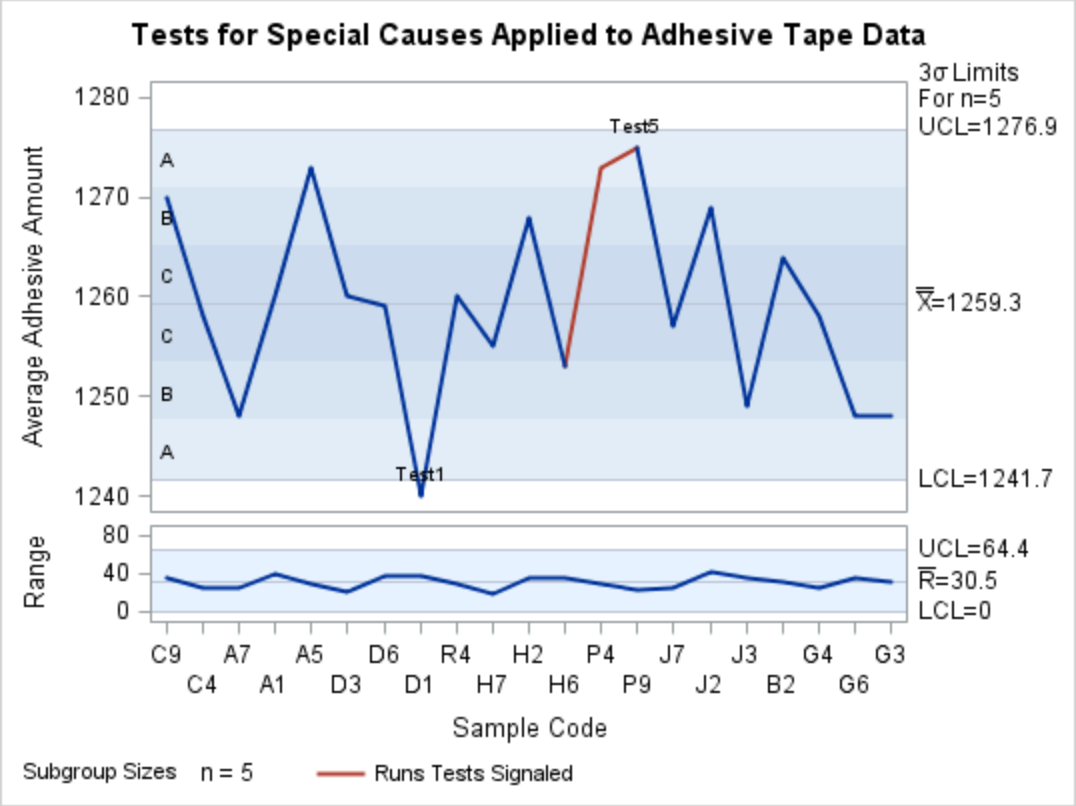


Figure 4.13 $\bar{X}$ and R Chart (SHEWHART)



Legacy Line Printer Displays

The following SAS/QC procedures continue to support legacy line printer charts and plots, drawn with characters, which are produced in the SAS output listing:

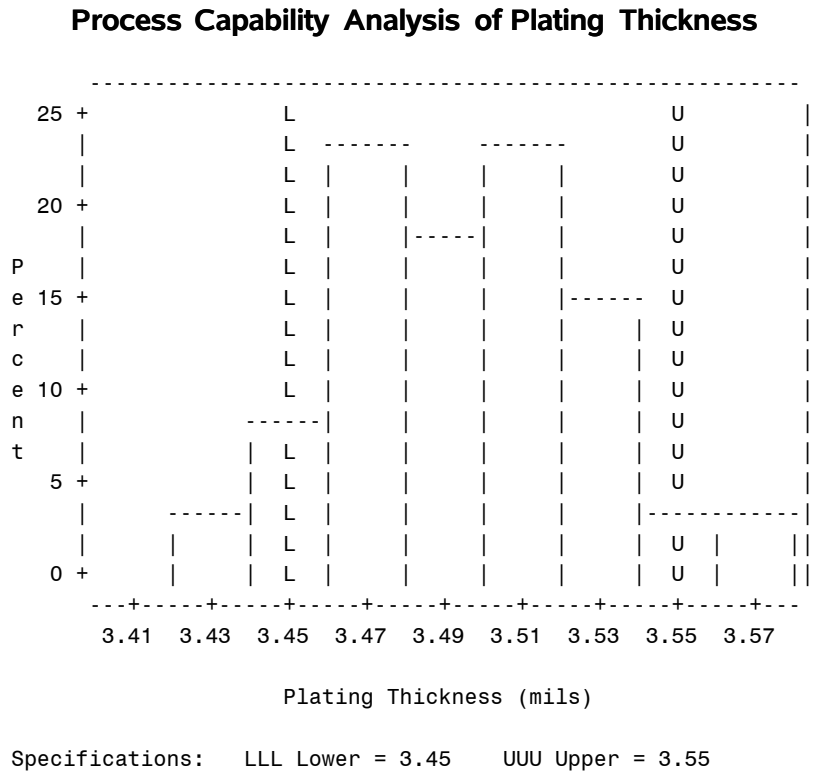
- CAPABILITY
- CUSUM
- MACONTROL
- PARETO
- SHEWHART

Beginning with SAS 7, these procedures produce high-resolution graphics by default, and you must specify the LINEPRINTER option in the PROC statement to create line printer charts, as illustrated by the following statements.

```
title 'Process Capability Analysis of Plating Thickness';
proc capability data=Trans noprint lineprinter;
   spec lsl=3.45 usl=3.55;
   histogram Thickness;
run;
```

The resulting histogram is shown in Figure 4.14.

Figure 4.14 Legacy Line Printer Display



In SAS/QC, some charts and plots are not supported with the LINEPRINTER option. For example, line printer displays are not available with the CAPABILITY procedure's COMPHISTOGRAM statement or the PARETO procedure's HBAR statement.

Chapter 5
The ANOM Procedure

Contents

PROC ANOM and General Statements **38**
Overview: ANOM Procedure 38
Uses of Analysis of Means 38
Terminology 39
History 40
Using the ANOM Procedure 40
Syntax: ANOM Procedure 41
BY Statement 41
ID Statement 41
Graphical Enhancement Statements 42
PROC ANOM Statement 42
BOXCHART Statement: ANOM Procedure **44**
Overview: BOXCHART Statement 44
Getting Started: BOXCHART Statement 45
Creating ANOM Boxcharts from Response Values 45
Creating ANOM Boxcharts from Group Summary Data 47
Saving Summary Statistics for Groups 50
Saving Decision Limits 51
Syntax: BOXCHART Statement 53
Summary of Options 54
Details: BOXCHART Statement 62
Constructing ANOM Boxcharts 62
Output Data Sets 64
ODS Tables 68
ODS Graphics 68
Input Data Sets 69
Axis Labels 74
Missing Values 74
Examples: BOXCHART Statement 75
Example 5.1: ANOM Boxcharts with Unequal Group Sizes 75
PCHART Statement: ANOM Procedure **77**
Overview: PCHART Statement 77
Getting Started: PCHART Statement 78
Creating ANOM Charts for Proportions from Group Counts 78
Creating ANOM Charts for Proportions from Group Summary Data 80
Saving Group Proportions 83

Saving Decision Limits . . . 84
Syntax: PCHART Statement . . . 85
Summary of Options . . . 86
Details: PCHART Statement . . . 94
Constructing ANOM Charts for Proportions . . . 94
Output Data Sets . . . 96
ODS Tables . . . 98
ODS Graphics . . . 99
Input Data Sets . . . 99
Axis Labels . . . 102
Missing Values . . . 103
Examples: PCHART Statement . . . 103
Example 5.2: ANOM p Charts with Angled Axis Labels . . . 103
UCHART Statement: ANOM Procedure . . . 105
Overview: UCHART Statement . . . 105
Getting Started: UCHART Statement . . . 106
Creating ANOM Charts for Rates from Group Counts . . . 106
Saving Decision Limits . . . 108
Syntax: UCHART Statement . . . 110
Summary of Options . . . 111
Details: UCHART Statement . . . 119
Constructing ANOM Charts for Rates . . . 119
Output Data Sets . . . 121
ODS Tables . . . 123
ODS Graphics . . . 123
Input Data Sets . . . 123
Axis Labels . . . 126
Missing Values . . . 127
Examples: UCHART Statement . . . 127
Example 5.3: ANOM u Charts with Angled Axis Labels . . . 127
XCHART Statement: ANOM Procedure . . . 129
Overview: XCHART Statement . . . 129
Getting Started: XCHART Statement . . . 130
Creating ANOM Charts for Means from Response Values . . . 130
Creating ANOM Charts for Means from Group Summary Data . . . 133
Saving Summary Statistics for Groups . . . 135
Saving Decision Limits . . . 136
Syntax: XCHART Statement . . . 137
Summary of Options . . . 138
Details: XCHART Statement . . . 146
Constructing ANOM Charts for Means . . . 146
Constructing ANOM Charts for Two-Way Layouts . . . 148
Output Data Sets . . . 150
ODS Tables . . . 152

ODS Graphics . . . 153
Input Data Sets . . . 153
Axis Labels . . . 156
Missing Values . . . 157
Examples: XCHART Statement . . . 157
Example 5.4: ANOM Charts with Unequal Group Sizes . . . 157
Example 5.5: ANOM for a Two-Way Classification . . . 159
Example 5.6: ANOM Charts Using LIMITS= Data Set . . . 163
Example 5.7: ANOM for Cell Means in Presence of Interaction . . . 164
INSET Statement: ANOM Procedure . . . 168
Overview: INSET Statement . . . 168
Getting Started: INSET Statement . . . 168
Displaying Summary Statistics on an ANOM Chart . . . 168
Formatting Values and Customizing Labels . . . 170
Adding a Header and Positioning the Inset . . . 172
Syntax: INSET Statement . . . 173
Summary of INSET Keywords . . . 174
Summary of Options . . . 175
Dictionary of Options . . . 176
Details: INSET Statement . . . 178
Positioning the Inset Using Compass Points . . . 179
Positioning the Inset in the Margins . . . 180
Positioning the Inset Using Coordinates . . . 180
Dictionary of ANOM Chart Statement Options . . . 183
References . . . 186

PROC ANOM and General Statements

Overview: ANOM Procedure

Analysis of means (ANOM) is a graphical and statistical method for simultaneously comparing k treatment means with their overall mean at a specified significance level α. You can use the ANOM procedure to create ANOM charts for various types of response data, including continuous measurements, proportions, and rates.

In addition, you can use the ANOM procedure to do the following:

- create charts from either response values or summarized data
- analyze multiple response variables
- specify decision limits in terms of the significance level (α)
- compute decision limits from the data and automatically adjust decision limits for unequal sample sizes
- save chart statistics and decision limits in output data sets
- tabulate chart statistics and decision limits.

See Chapter 4, "SAS/QC Graphics," for a detailed discussion of the alternatives available for producing charts with SAS/QC procedures.

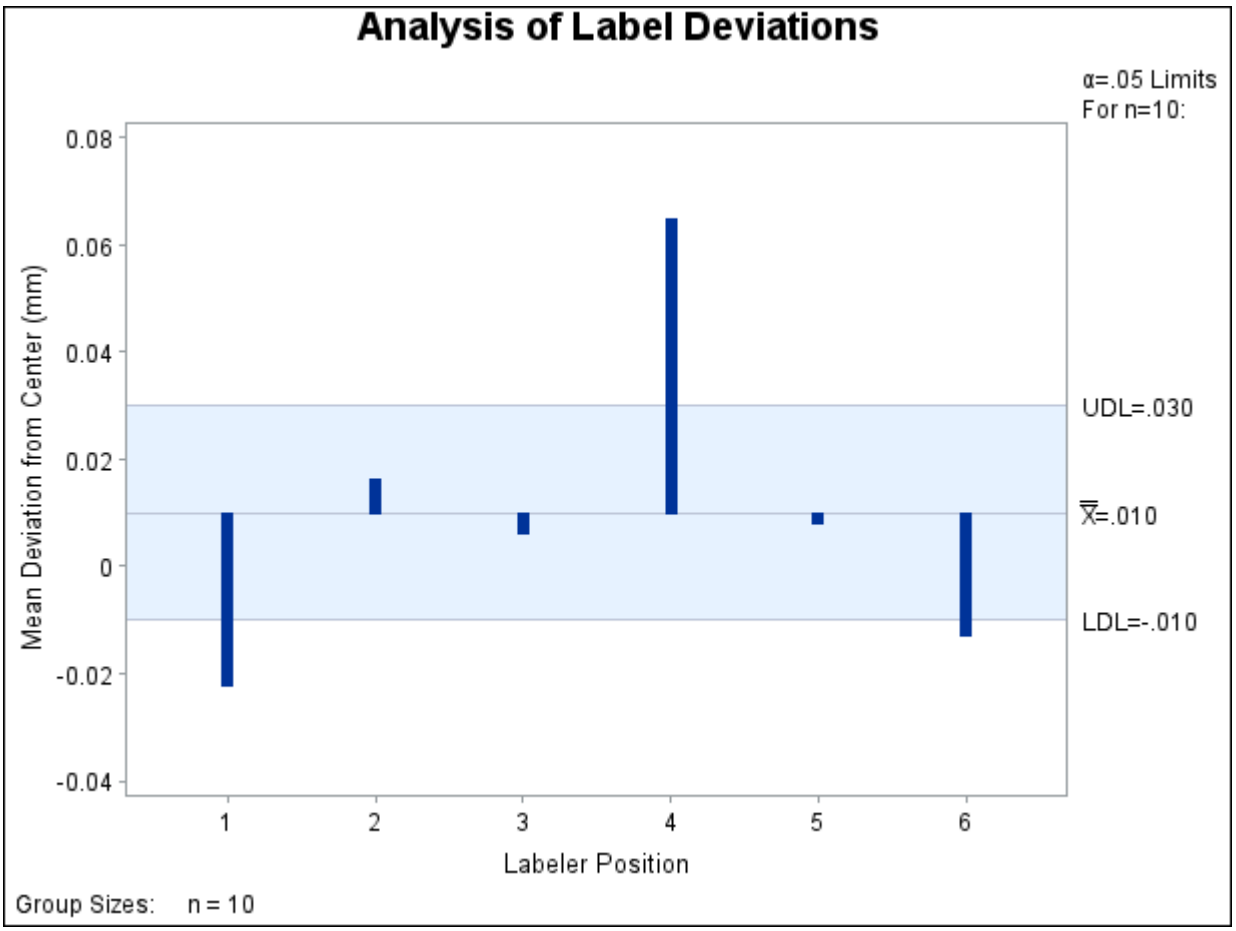
Uses of Analysis of Means

Many statistical quality improvement applications involve a comparison of treatment means to determine which are significantly different from the overall average. For example, a manufacturing engineer might run an experiment to investigate which of six positions on a machine are producing different output, in the sense that the average measurement for each position differs from the overall average. Likewise, a health care system administrator might ask which clinics in the system have a higher or lower rate of admissions than the average for all clinics.

Questions of this type can be answered with *analysis of means*, which is an alternative to one-way analysis of variance (ANOVA) for a fixed effects model. However, unlike ANOVA, which simply determines whether there is a statistically significant difference in the treatment means, ANOM identifies the means that are significantly different from the overall mean. As a statistical technique, ANOM is a method for making multiple comparisons that is sometimes referred to as a "multiple comparison with the weighted mean." Analysis of means lends itself to quality improvement applications because it has a simple graphical representation that is similar to a Shewhart chart and requires little training to interpret. This representation is also useful for assessing practical significance.

Figure 5.1 illustrates a typical ANOM chart. The central line represents the overall average. The treatment means, plotted as deviations from the overall average are compared with upper and lower decision limits to identify which are significantly different from the overall mean (in this case, the means corresponding to the first, fourth, and sixth positions).

Figure 5.1 Typical ANOM Chart



Although the term "analysis of means" suggests that the method is intended for means of continuous response measurements, the method is also applicable to means of attributes data, including proportions and rates.

Analysis of means was introduced as a tool for statistical quality control by Ellis Ott in 1967, and it became popular during the early 1980s, when it was applied to experimental data in manufacturing. In this setting, measurements are taken at a number of treatment levels (factor levels). During the 1990s, the use of ANOM spread to service industry applications and, in particular, health care quality improvement. In these settings, data (such as utilization rates) are observed for a number of groups (such as hospitals or clinics).

Terminology

In order to accommodate the growing variety of modern applications for analysis of means, the term *group* is used instead of treatment level throughout the documentation for the ANOM procedure. Likewise, the term *group-variable* is used to refer to the variable in the input data set that classifies the observations into treatment levels. In the ANOM procedure, a *group-variable* plays the same role as a CLASS variable in the GLM and ANOVA procedures, and it is syntactically the same as a *subgroup-variable* in the SHEWHART procedure.

The nomenclature for ANOM charts is the same as that for Shewhart charts: $\bar{X}$ charts for means, p charts for proportions, and u charts for rates. Consequently, the syntax for the ANOM procedure is patterned after

the syntax for the SHEWHART procedure. However, there are some important differences between ANOM charts and Shewhart charts:

- Analysis of means is formally a test of hypothesis, whereas a Shewhart chart is used to distinguish between special and common causes of variation.
- In an ANOM chart, the horizontal axis corresponds to the *group-variable*, and it identifies the groups, which can be displayed in any order. In a Shewhart chart, the horizontal axis corresponds to the *subgroup-variable*, and it identifies the order in which the subgroup measurements were taken.
- An ANOM chart displays response summary statistics for a set of groups (treatments) at a specific time. A Shewhart chart displays subgroup summary statistics for a specific process where the subgroups are made up of measurements taken over successive points in time.
- In an ANOM chart, the decision limits are determined by a specified significance level (α), which is the probability that under the null hypothesis of no treatment differences, at least one of the response summary statistics will exceed the decision limits. In a Shewhart chart, control limits are typically computed as 3σ limits.

History

Analysis of means compares the absolute deviations of group means from their overall mean, an approach that was initially studied by Laplace in 1827. Halperin et al. derived a version of this method in the form of a multiple significance test in 1955. Ott developed a graphical representation for the test and introduced the term "analysis of means" in 1967. Refer to Ott (1967) and Ott (1975).

P. R. Nelson (1982a) and L. S. Nelson (1983) provided exact critical values for ANOM when the groups have equal sample sizes. P. R. Nelson (1991) developed a method for computing exact critical values for ANOM when the group sample sizes are not equal. Refer to Nelson, Coffin, and Copeland (2003) for more information on the use of ANOM in engineering experimentation.

Using the ANOM Procedure

The PROC ANOM statement invokes the ANOM procedure and it optionally identifies various data sets.

To create an ANOM chart, you specify a chart statement (after the PROC ANOM statement) that specifies the type of ANOM chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request a basic ANOM chart for treatment means:

```
proc anom data=Values;
   xchart Weight*Treatment;
run;
```

Here, the DATA= option specifies an input data set (Values) that contains the *response* measurement variable (Weight) and the *group-variable* (Treatment). You can use options in the PROC ANOM statement to

- specify input data sets containing variables to be analyzed, decision limits, and annotation information
- specify a graphics catalog for saving graphical output

NOTE: If you are learning to use the ANOM procedure, you should read both this section and the "Getting Started" subsection in the section for the chart statement that corresponds to the chart you want to create.

Syntax: ANOM Procedure

The following are the primary statements that control the ANOM procedure:

PROC ANOM < *options* > **;**
BOXCHART (*responses*) ∗ *group-variable* < (*block-variables*) >
< **=***symbol-variable* > < / *options* > **;**
PCHART (*responses*) ∗ *group-variable*
< (*block-variables*) >
< **=***symbol-variable* > < / *options* > **;**
UCHART (*responses*) ∗ *group-variable*
< (*block-variables*) >
< **=***symbol-variable* > < / *options* > **;**
XCHART (*responses*) ∗ *group-variable* < (*block-variables*) >
< **=***symbol-variable* > < / *options* > **;**
INSET *keyword-list* < / *options* > **;**

The PROC ANOM statement invokes the procedure and specifies the input data set. The chart statements create different types of charts. You can specify one or more of each of the chart statements. For details, read the section on the chart statement that corresponds to the type of chart that you want to produce.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ANOM to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the ANOM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

ID Statement

In addition, you can optionally specify the following statement:

ID *variables* **;**

The ID statement specifies variables used to identify observations. The ID variables must be variables in the DATA= or SUMMARY= input data sets.

The ID variables are used in the following ways:

- If you create an OUTSUMMARY= or OUTTABLE= data set, the ID variables are included. If the input data set is a DATA= data set, only the values of the ID variables from the first observation in each group are passed to the output data set.
- If you specify the TABLEID or TABLEALL options in a chart statement, the table produced is augmented by a column for each of the ID variables. Only the values of the ID variables from the first observation in each group are tabulated.
- If you specify the BOXSTYLE= SCHEMATICID option or the BOXSTYLE= SCHEMATICIDFAR option in the BOXCHART statement, the value of the first variable listed in the ID statement is used to label each extreme observation.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance graphical and printed output. You can also use AXIS, LEGEND, and SYMBOL statements to enhance traditional graphics. For details, refer to *SAS/GRAPH: Reference* and see the section for the chart statement that you are using.

PROC ANOM Statement

The syntax for the PROC ANOM statement is as follows:

PROC ANOM *options* **;**

The PROC ANOM statement starts the ANOM procedure and optionally identifies various data sets. The following options can appear in the PROC ANOM statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set containing ANNOTATE= variables as described in *SAS/GRAPH: Reference*. You can use this data set to add features to ANOM charts produced as traditional graphics. Features provided in this data set are displayed on every chart produced in the current run of the ANOM procedure. This option is ignored if you are not producing traditional graphics.

BOX=*SAS-data-set*

names an input data set that contains group summary statistics, decision limits, and outlier values in "strung out" form, with more than one observation per group. Each observation corresponds to one feature of one group's box-and-whisker plot. Typically, this data set is created as an OUTBOX= data set in a previous run of the ANOM procedure with a BOXCHART statement. The BOX= data set is the only kind of summary data set you can use to produce schematic box-and-whisker plots. The BOXCHART statement is the only chart statement you can use with a BOX= input data set.

DATA=*SAS-data-set*

names an input data set that contains response values (typically, measurements or counts) as observations. Note that the DATA= data set may need sorting. If the values of the *group-variable* are numeric, you must sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted.

The DATA= data set may contain more than one observation for each value of the *group-variable*. This happens, for example, when you produce a chart for means and ranges with the XCHART statement.

You cannot use a DATA= data set together with a SUMMARY= or a TABLE= data set. If you do not specify one of these three input data sets, the ANOM procedure uses the most recently created data set as a DATA= data set. For more information, see the "DATA= Data Set" subsection in the section for the chart statement you are using.

GOUT=*graphics-catalog*

specifies the graphics catalog for traditional graphics output from the ANOM procedure. This is useful if you want to save the output. This option is ignored if you are not producing traditional graphics.

SUMMARY=*SAS-data-set*

names an input data set that contains group summary statistics. For example, you can read sample sizes, means, and standard deviations for the groups to create an ANOM chart. Typically, this data set is created as an OUTSUMMARY= data set in a previous run of the ANOM procedure, but it can also be created using a SAS summarization procedure such as PROC MEANS.

Note that the SUMMARY= data sets may need sorting. If the values of the *group-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted. The SUMMARY= data set can contain only one observation for each value for the *group-variable*.

You cannot use a SUMMARY= data set with a DATA= or a TABLE= data set. If you do not specify one of these three input data sets, the ANOM procedure uses the most recently created data set as a DATA= data set. For more information, see the "SUMMARY= Data Set" subsection in the section for the chart statement you are using.

LIMITS=*SAS-data-set*

names an input data set that contains preestablished decision limits or the parameters from which decision limits can be computed. Each observation in a LIMITS= data set provides decision limit information for a *response*. Typically, this data set is created as an OUTLIMITS= data set in a previous run of the ANOM procedure.

If you omit the LIMITS= option, then decision limits are computed from the data in the DATA= or SUMMARY= input data sets. For details about the variables needed in a LIMITS= data set, see the "LIMITS= Data Set" subsection in the section for the chart statement you are using.

TABLE=*SAS-data-set*

names an input data set that contains group summary statistics and decision limits. Each observation in a TABLE= data set provides information for a particular group and *response*. Typically, this data set is created as an OUTTABLE= data set in a previous run of the ANOM procedure.

You cannot use a TABLE= data set with a DATA= or a SUMMARY= data set. If you do not specify one of these three input data sets, the ANOM procedure uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" subsection in the section for the chart statement that you are using.

BOXCHART Statement: ANOM Procedure

Overview: BOXCHART Statement

The BOXCHART statement creates an ANOM chart for group (treatment level) means of response values superimposed with box-and-whisker plots of the measurements in each group. Throughout this chapter, a chart of this type is referred to as an *ANOM boxchart*. You can use options in the BOXCHART statement to

- compute decision limits from the data based on a specified parameters, such as the significance level (α)
- tabulate group sample sizes, group means, decision limits, and other information
- save decision limits in an output data set
- save group sample sizes and group means in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups
- specify one of several methods for calculating quantile statistics (percentiles)
- control the style of the box-and-whisker plots
- add block legends and symbol markers to identify special groups
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have two alternatives for producing ANOM boxcharts with the BOXCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced if SAS/GRAPH® is licensed.

See Chapter 4, “SAS/QC Graphics,” for more information about producing these different kinds of graphs.

Getting Started: BOXCHART Statement

This section introduces the BOXCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the BOXCHART statement is presented in the section "Syntax: BOXCHART Statement" on page 53, and advanced examples are given in the section "Examples: BOXCHART Statement" on page 75.

Creating ANOM Boxcharts from Response Values

NOTE: See *Creating ANOM BOXCHARTS from Response Values* in the SAS/QC Sample Library.

A manufacturing engineer carries out a study to determine the source of excessive variation in the positioning of labels on shampoo bottles.[1] A labeling machine removes bottles from the line, attaches the labels, and returns the bottles to the line. There are six positions on the machine, and the engineer suspects that one or more of the position heads might be faulty.

A sample of 60 bottles, 10 per position, is run through the machine. For each bottle, the deviation of each label is measured in millimeters, and the machine position is recorded. The following statements create a SAS data set named LabelDeviations, which contains the deviation measurements for the 60 bottles:

```
data LabelDeviations;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.02386  -0.02853  -0.03001  -0.00428  -0.03623
1  -0.04222  -0.00144  -0.06466   0.00944  -0.00163
2  -0.02014  -0.02725   0.02268  -0.03323   0.03661
2   0.04378   0.05562   0.00977   0.05641   0.01816
3  -0.00728   0.02849  -0.04404  -0.02214  -0.01394
3   0.04855   0.03566   0.02345   0.01339  -0.00203
4   0.06694   0.10729   0.05974   0.06089   0.07551
4   0.03620   0.05614   0.08985   0.04175   0.05298
5   0.03677   0.00361   0.03736   0.01164  -0.00741
5   0.02495  -0.00803   0.03021  -0.00149  -0.04640
6   0.00493  -0.03839  -0.02037  -0.00487  -0.01202
6   0.00710  -0.03075   0.00167  -0.02845  -0.00697
;
```

A partial listing of LabelDeviations is shown in Figure 5.2.

[1] This example is based on a case study described by Hansen (1990).

Figure 5.2 Listing of the Data Set LabelDeviations

The Data Set LabelDeviations

Position	Deviation
1	-0.02386
1	-0.02853
1	-0.03001
1	-0.00428
1	-0.03623
1	-0.04222
1	-0.00144
1	-0.06466
1	0.00944
1	-0.00163
2	-0.02014
2	-0.02725

The data set LabelDeviations is said to be in "strung-out" form, because each observation contains the position and the deviation measurement for a single bottle. The first 10 observations contain the measurements for the first position, the second 10 observations contain the measurements for the second position, and so on. Because the variable Position classifies the observations into groups (treatment levels), it is referred to as the *group-variable*. The variable Deviation contains the deviation measurements and is referred to as the *response variable* (or *response* for short).

The following statements create the ANOM boxchart shown in Figure 5.3:

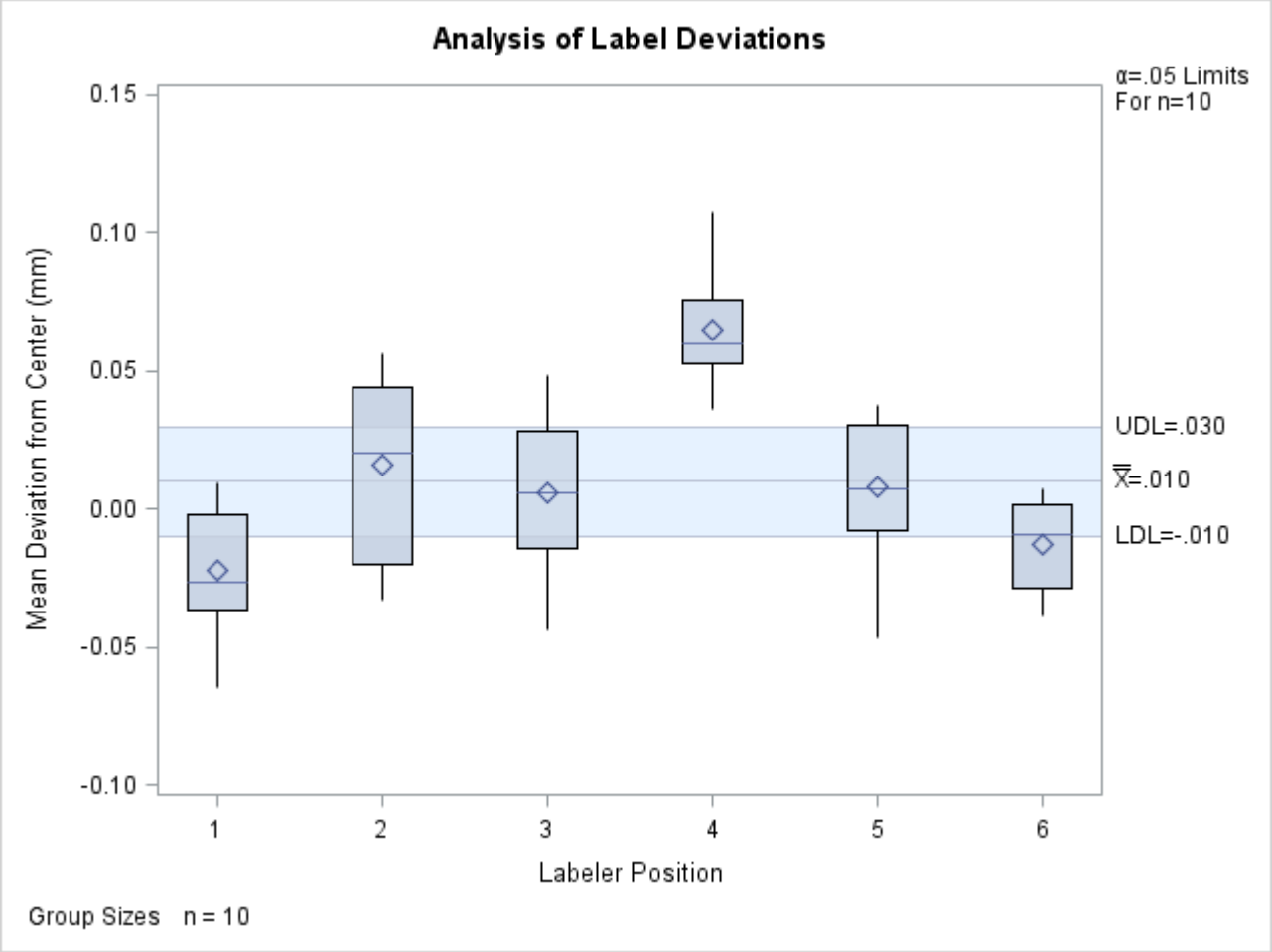
```
ods graphics on;
title 'Analysis of Label Deviations';
proc anom data=LabelDeviations;
   boxchart Deviation*Position / alpha    = 0.05
                                odstitle = title;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

The ODS GRAPHICS ON statement specified before the PROC ANOM statement enables ODS Graphics, so the boxchart is created by using ODS Graphics instead of traditional graphics. This example illustrates the basic form of the BOXCHART statement. After the keyword BOXCHART, you specify the *response* to analyze (in this case, Deviation) followed by an asterisk and the *group-variable* (Position). Options are specified after the slash (/) in the BOXCHART statement. A complete list of options is presented in the section "Syntax: BOXCHART Statement" on page 53.

The input data set is specified with the DATA= option in the PROC ANOM statement when it contains raw measurements for the *response*.

Each point on the ANOM chart represents the average (mean) of the response measurements for a particular sample.

Figure 5.3 ANOM Chart for Means of Labeler Position Data



The average for Position 1 is below the lower decision limit (LDL), and the average for Position 6 is slightly below the lower decision limit. The average for Position 4 exceeds the upper decision limit (UDL). The conclusion is that Positions 1, 4, and 6 are operating differently.

By default, the decision limits shown correspond to a significance level of $\alpha = 0.05$; the formulas for the limits are given in the section "Decision Limits" on page 63. You can also read decision limits from an input data set.

For computational details, see "Constructing ANOM Boxcharts" on page 62. For details on reading raw measurements, see "DATA= Data Set" on page 70.

Creating ANOM Boxcharts from Group Summary Data

NOTE: See *Creating BOXCHARTS from Group Summary Data* in the SAS/QC Sample Library.

The previous example illustrates how you can create ANOM charts for means using measurement data. However, in many applications, the data are provided as group summary statistics. This example illustrates how you can use the BOXCHART statement with data of this type.

The following data set (Labels) provides the data from the preceding example in summarized form:

```
data Labels;
   input Position DeviationL Deviation1 DeviationX
         DeviationM Deviation3 DeviationH DeviationS;
   DeviationN = 10;
   datalines;
1  -0.0647  -0.0362  -0.02234  -0.02620  -0.0016  0.0094  0.02281
2  -0.0332  -0.0201   0.01625   0.02045   0.0438  0.0564  0.03347
3  -0.0440  -0.0139   0.00604   0.00570   0.0285  0.0486  0.02885
4   0.0362   0.0530   0.06473   0.06030   0.0755  0.1073  0.02150
5  -0.0464  -0.0074   0.00813   0.00760   0.0302  0.0374  0.02593
6  -0.0384  -0.0285  -0.01283  -0.00950   0.0017  0.0071  0.01599
;
```

A listing of Labels is shown in Figure 5.4. There is exactly one observation for each group (note that the groups are still indexed by Position). There are eight summary variables in Labels.

- DeviationL contains the group minimums (low values).
- Deviation1 contains the 25th percentile (first quartile) of each group.
- DeviationX contains the group means.
- DeviationM contains the group medians.
- Deviation3 contains the 75th percentile (third quartile) of each group.
- DeviationH contains the group maximums (high values).
- DeviationS contains the group standard deviations.
- DeviationN contains the group sample sizes (these are all 10 in this case).

Figure 5.4 The Summary Data Set Labels

The Data Set Labels

Position	DeviationL	Deviation1	DeviationX	DeviationM	Deviation3	DeviationH	DeviationS	DeviationN
1	-0.0647	-0.0362	-0.02234	-0.02620	-0.0016	0.0094	0.02281	10
2	-0.0332	-0.0201	0.01625	0.02045	0.0438	0.0564	0.03347	10
3	-0.0440	-0.0139	0.00604	0.00570	0.0285	0.0486	0.02885	10
4	0.0362	0.0530	0.06473	0.06030	0.0755	0.1073	0.02150	10
5	-0.0464	-0.0074	0.00813	0.00760	0.0302	0.0374	0.02593	10
6	-0.0384	-0.0285	-0.01283	-0.00950	0.0017	0.0071	0.01599	10

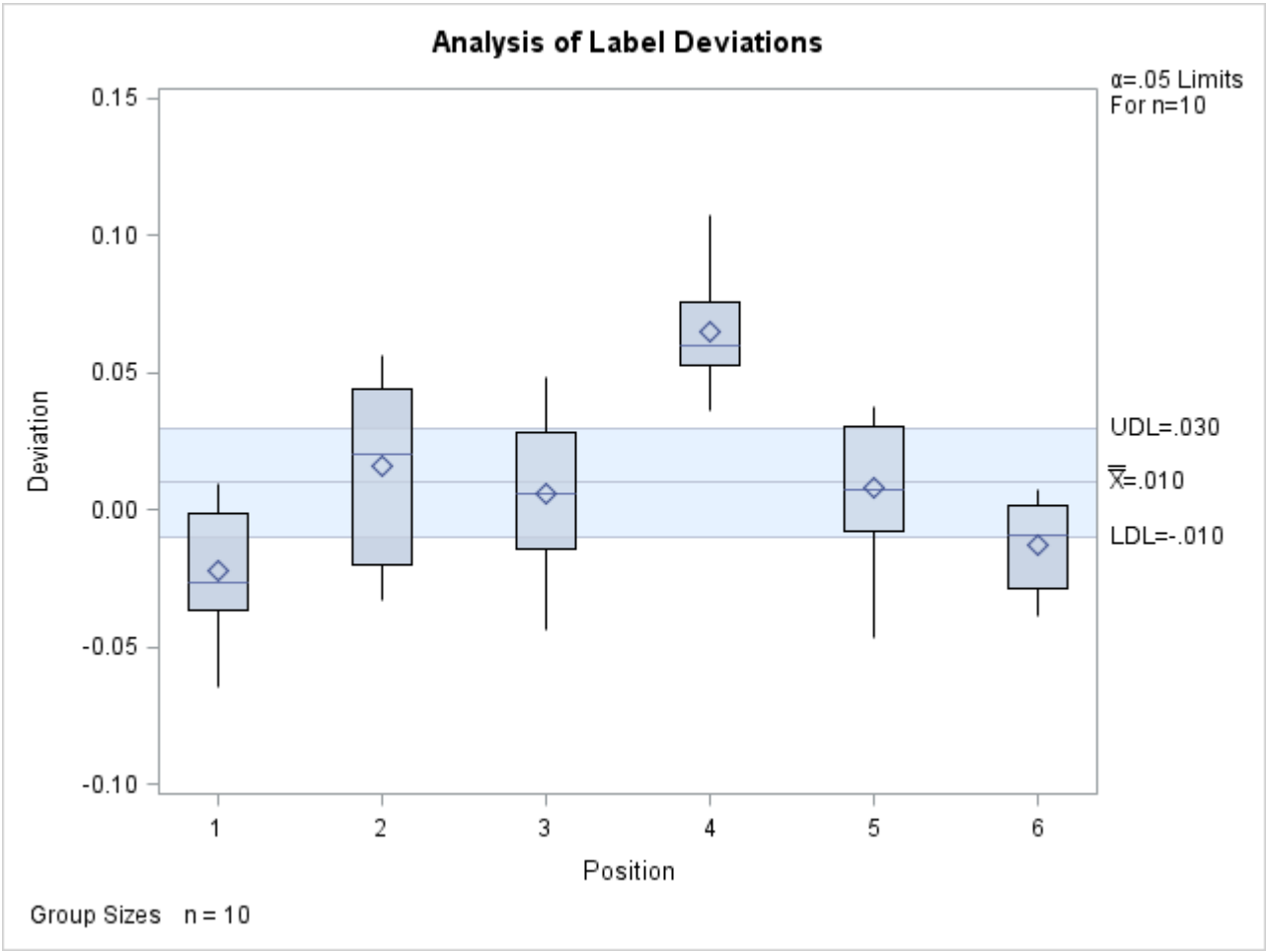
You can read this data set by specifying it as a SUMMARY= data set in the PROC ANOM statement, as follows:

```
ods graphics on;
title 'Analysis of Label Deviations';
proc anom summary=Labels;
   boxchart Deviation*Position / odstitle=title1;
run;
```

The resulting ANOM boxchart is shown in Figure 5.5.

Note that Deviation is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the eight summary variables. The suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* indicate the contents of the variable. For example, the suffix characters *1* and *3* indicate first and third quartiles. Thus, you can specify three group summary variables in a SUMMARY= data set with a single name (Deviation), which is referred to as the *response*. The name Position specified after the asterisk is the name of the *group-variable*.

Figure 5.5 ANOM Chart for Means in Data Set Labels



In general, a SUMMARY= input data set used with the BOXCHART statement must contain the following variables:

- group variable
- group minimum variable
- group first quartile variable
- group mean variable
- group median variable
- group third quartile variable
- group maximum variable
- group standard deviation variable
- group sample size variable

Furthermore, the names of the summary variables must begin with the *response* name specified in the BOXCHART statement and end with the appropriate suffix characters. If the names do not follow this convention, you can use the RENAME option in the PROC ANOM statement to rename the variables for the duration of the ANOM procedure step. If a label is associated with the group mean variable, it is used to label the vertical axis.

In summary, the interpretation of *response* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *response* is the name of the SAS variable containing the response measurements.
- If summary data are read using the SUMMARY= option (as in this example), *response* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "SUMMARY= Data Set" on page 72.

Saving Summary Statistics for Groups

NOTE: See *Saving Summary Statistics for Groups* in the SAS/QC Sample Library.

In this example, the BOXCHART statement is used to create a data set containing group summary statistics that can be read later by the ANOM procedure (as in the preceding example). The following statements read measurements from the data set LabelDeviations and create a summary data set named LabelSummary:

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position / outsummary=LabelSummary
                                nochart;
run;
```

The OUTSUMMARY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to Figure 5.3.

Figure 5.6 contains a listing of LabelSummary.

Figure 5.6 The Summary Data Set LabelSummary

The Data Set LabelSummary

Position	DeviationL	Deviation1	DeviationX	DeviationM	Deviation3	DeviationH	DeviationS	DeviationN
1	-0.06466	-0.03623	-0.022342	-0.026195	-0.00163	0.00944	0.022805	10
2	-0.03323	-0.02014	0.016241	0.020420	0.04378	0.05641	0.033478	10
3	-0.04404	-0.01394	0.006011	0.005680	0.02849	0.04855	0.028847	10
4	0.03620	0.05298	0.064729	0.060315	0.07551	0.10729	0.021492	10
5	-0.04640	-0.00741	0.008121	0.007625	0.03021	0.03736	0.025920	10
6	-0.03839	-0.02845	-0.012812	-0.009495	0.00167	0.00710	0.015974	10

There are nine variables in the data set LabelSummary.

- Position identifies the group.
- DeviationL contains the group minimums.
- Deviation1 contains the first quartile for each group.
- DeviationX contains the group means.
- DeviationM contains the group medians.
- Deviation3 contains the third quartile for each group.
- DeviationH contains the group maximums.
- DeviationS contains the group standard deviations.
- DeviationN contains the group sizes.

Note that the summary statistic variables are named by adding the suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* to the *response* Deviation specified in the BOXCHART statement. In other words, the variable naming convention for OUTSUMMARY= data sets is the same as that for SUMMARY= data sets.

For more information, see "OUTSUMMARY= Data Set" on page 66.

Saving Decision Limits

NOTE: See *Saving Decision Limits Using ANOM BOXCHART* in the SAS/QC Sample Library.

You can save the decision limits for an ANOM chart, together with the parameters used to compute the limits, in a SAS data set.

The following statements read measurements from the data set LabelDeviations (see "Creating ANOM Boxcharts from Response Values" on page 45.) and save the decision limits displayed in Figure 5.3 in a data set named LabelLimits:

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position / outlimits=LabelLimits
                                nochart;
run;
```

The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set LabelLimits is listed in Figure 5.7.

Figure 5.7 The Data Set LabelLimits Containing Decision Limit Information

Decision Limits for Labler Position Deviations

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLX_	_MEAN_	_UDLX_	_MSE_	_DFE_	_LIMITK_
Deviation	Position	ESTIMATE	10	0.05	-.009878975	.009991333	0.029862	.000643646	54	6

The data set LabelLimits contains one observation with the limits for *response* Deviation. The values of _LDLX_ and _UDLX_ are the lower and upper decision limits for the means, and the value of _MEAN_ is the weighted average of the group means, which is represented by the central line.

The values of _MEAN_, _MSE_, _DFE_, _LIMITK_, _LIMITN_, and _ALPHA_ are the parameters used to compute the decision limits. The value of _MSE_ is the mean square error, and the value of _DFE_ is the associated degrees of freedom. The value of _LIMITK_ is the group size (k), the value of _LIMITN_ is the nominal sample size associated with the decision limits, and the value of _ALPHA_ is the value of the significance level (α). The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _MSE_ are estimates computed from the data or standard (known) values specified with procedure options. In most applications, the value of _TYPE_ will be 'ESTIMATE.'

NOTE: See *Saving Decision Limits and Summary Statistics* in the SAS/QC Sample Library.

You can create an output data set containing both decision limits and group summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position / outtable=LabelTab
                                nochart;
run;
```

The data set LabelTab is listed in Figure 5.8.

Figure 5.8 The Data Set LabelTab

Summary Statistics and Decision Limits

VAR	Position	_ALPHA_	_LIMITN_	_SUBN_	_LDLX_	_SUBX_	_MEAN_	_UDLX_
Deviation	1	0.05	10	10	-.009878975	-0.022342	.009991333	0.029862
Deviation	2	0.05	10	10	-.009878975	0.016241	.009991333	0.029862
Deviation	3	0.05	10	10	-.009878975	0.006011	.009991333	0.029862
Deviation	4	0.05	10	10	-.009878975	0.064729	.009991333	0.029862
Deviation	5	0.05	10	10	-.009878975	0.008121	.009991333	0.029862
Deviation	6	0.05	10	10	-.009878975	-0.012812	.009991333	0.029862

EXLIM	_SUBMIN_	_SUBQ1_	_SUBMED_	_SUBQ3_	_SUBMAX_
LOWER	-0.06466	-0.03623	-0.026195	-0.00163	0.00944
	-0.03323	-0.02014	0.020420	0.04378	0.05641
	-0.04404	-0.01394	0.005680	0.02849	0.04855
UPPER	0.03620	0.05298	0.060315	0.07551	0.10729
	-0.04640	-0.00741	0.007625	0.03021	0.03736
LOWER	-0.03839	-0.02845	-0.009495	0.00167	0.00710

This data set contains one observation for each group sample. The variable _SUBMIN_ contains the group minimums, and the variable _SUBQ1_ contains the first quartile for each group. The variables _SUBX_ and _SUBMED_ contain the group means and medians. The variable _SUBQ3_ contains the third quartiles, _SUBMAX_ contains the group maximums, and _SUBN_ contains the group sample sizes. The variables _LDLX_ and _UDLX_ contain the lower and upper decision limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Position contain the *response* name and values of the *group-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 67.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read LabelTab and display an ANOM boxchart (not shown here) identical to the chart in Figure 5.3:

```
title 'Analysis of Label Deviations';
proc anom table=LabelTab;
   boxchart Deviation*Position / odstitle=title;
label _SUBX_ = 'Mean Deviation from Center (mm)';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM boxcharts.

For more information, see "TABLE= Data Set" on page 73.

Syntax: BOXCHART Statement

The basic syntax for the BOXCHART statement is as follows:

BOXCHART *response* ∗ *group-variable* **;**

The general form of this syntax is as follows:

BOXCHART *(responses)* ∗ *group-variable* < **(***block-variables***)** >
< **=***symbol-variable* > < *options* > **;**

You can use any number of BOXCHART statements in the ANOM procedure. The components of the BOXCHART statement are described as follows.

responses

identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If response values (raw data) are read from a DATA= data set, *response* must be the name of the variable containing the values.For an example, see "Creating ANOM Boxcharts from Response Values" on page 45.
- If summary data are read from a SUMMARY= data set, *response* must be the common prefix of the summary variables in the SUMMARY= data set. For an example, see "Creating ANOM Boxcharts from Group Summary Data" on page 47.
- If summary data and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Decision Limits" on page 51.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM charts for the means of Weight, Length, and Width:

```
proc anom data=Measures;
   xchart (Weight Length Width)*Day;
run;
```

group-variable
is the variable that identifies groups in the data. The *group-variable* is required. In the preceding BOXCHART statement, Day is the group variable.

block-variables
are optional variables that group the data into blocks of consecutive groups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker used to plot the means. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements.

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function.

Summary of Options

The following tables list the BOXCHART statement options by function. Options unique to the ANOM procedure are listed in Table 5.1, and are described in detail in "Dictionary of ANOM Chart Statement Options" on page 183. Options that are common to both the ANOM and SHEWHART procedures are listed in Table 5.2, and are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 5.1 BOXCHART Statement Special Options

Option	Description
Options for Specifying Parameters for Decision Limits	
ALPHA=	Specifies the probability of a Type I error
DFE=	Specifies the degrees of freedom associated with the root mean square error
LIMITK=	Specifies number of groups for decision limits
LIMITN=	Specifies either nominal sample size for fixed decision limits or varying limits
MEAN=	Specifies the mean
MSE=	Specifies the mean square error
NOREADLIMITS	Computes decision limits for each *response* from the data rather than a LIMITS= data set
READINDEXES=	Reads multiple sets of decision limits for each *response* from a LIMITS= data set

Table 5.1 *continued*

Option	Description
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Displaying Decision Limits	
CINFILL=	Specifies color for area inside decision limits
CLIMITS=	Specifies color of decision limits, central line, and related labels
LDLLABEL=	Specifies label for lower decision limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=	Specifies line type for decision limits
NDECIMAL=	Specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	Suppresses display of central line
NOLDL	Suppresses display of lower decision limit
NOLIMITLABEL	Suppresses labels for decision limits and central line
NOLIMITS	Suppresses display of decision limits
NOLIMITSFRAME	Suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for decision limits
NOUDL	Suppresses display of upper decision limit
UDLLABEL=	Specifies label for upper decision limit
WLIMITS=	Specifies width for decision limits and central line
XSYMBOL=	Specifies label for central line
Output Data Set Option	
OUTSUMMARY=	Creates output data set containing group summary statistics

Table 5.2 BOXCHART Statement General Options

Option	Description
Options for Controlling Box Appearance	
BOXCONNECT=	Connects group means, medians, maximum values, minimum values, or quartiles in box-and-whisker plots
BOXSTYLE=	Specifies style of box-and-whisker plots
BOXWIDTH=	Specifies width of box-and-whisker plots
BOXWIDTHSCALE=	Specifies that widths of box-and-whisker plots vary proportionately to group sample size
CBOXES=	Specifies color for outlines of box-and-whisker plots
CBOXFILL=	Specifies fill color for interior of box-and-whisker plots

Table 5.2 *continued*

Option	Description
IDCOLOR=	Specifies outlier symbol color in schematic box-and-whisker plots
IDCTEXT=	Specifies text color to label outliers or response variable values
IDFONT=	Specifies text font to label outliers or response variable values
IDHEIGHT=	Specifies text height to label outliers or response variable values
IDSYMBOL=	Specifies outlier symbol in schematic box-and-whisker plots
LBOXES=	Specifies line types for outlines of box-and-whisker plots
NOTCHES	Specifies that box-and-whisker plots are to be notched
PCTLDEF=	Specifies percentile definition used for box-and-whisker plots
SERIFS	Adds serifs to the whiskers of skeletal box-and-whisker plots
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on ANOM boxchart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
COUT=	Specifies color for portions of line segments that connect points outside decision limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels
LABELHEIGHT=	Specifies height of labels
OUTLABEL=	Labels points outside decision limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text

Table 5.2 *continued*

Option	Description
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of ANOM boxchart
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for ANOM boxchart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all groups
BILEVEL	Creates ANOM boxchart using half-screens and half-pages
EXCHART	Creates ANOM boxchart for a response only when a group mean exceeds the decision limits
INTERVAL=	Specifies natural time interval between consecutive group positions when time, date, or datetime format is associated with a numeric group variable
MAXPANELS=	Maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for group sample sizes
NPANELPOS=	Specifies number of group positions per panel on each chart
REPEAT	Repeats last group position on panel as first group position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart

Table 5.2 *continued*

Option	Description
ZEROSTD	Displays ANOM boxchart regardless of whether root mean square error is zero
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= option
CVREF=	Specifies color for lines requested by VREF= option
HREF=	Specifies position of reference lines perpendicular to horizontal axis on ANOM boxchart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on ANOM boxchart
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on ANOM boxchart
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	Specifies position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to ANOM boxchart

Table 5.2 *continued*

Option	Description
DESCRIPTION=	Specifies description of ANOM boxchart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on chart
NAME=	Specifies name of ANOM boxchart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
BOXES=	Specifies variables whose values define colors box outlines
BOXFILL=	Specifies variables whose values define colors for filling boxes
CFRAMELAB	Draws a frame around labeled points
CPHASEBOX	Requests boxes enclosing all plotted points for a phase
CPHASEBOXCONNECT	Requests lines connecting adjacent enclosing boxes
CPHASEBOXFILL	Fills boxes enclosing all plotted points for a phase
CPHASEMEANCONNECT	Requests lines connecting phase average value points
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
BOXTRANSPARENCY=	Specifies the box fill transparency for box-and-whisker charts
INFILLTRANSPARENCY=	Specifies the decision limit infill transparency
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOBOXFILLLEGEND	Suppresses legend for levels of a BOXFILL= variable
NOFILLLEGEND	Suppresses legend for levels of a BOXFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEBOXLABELS	draws phase labels as titles along the top of phase boxes
PHASEPOS=	Specifies vertical position of phase legend

Table 5.2 *continued*

Option	Description
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
URL=	Specifies a variable whose values are URLs to be associated with groups
WBOXES=	Specifies width of box outlines for box-and-whisker charts
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTBOX=	Creates output data set containing group summary statistics, decision limits, and outlier values
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing decision limits
OUTTABLE=	Creates output data set containing group summary statistics and decision limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of group means, group sample sizes, and decision limits
TABLEALL	Creates all the tables that are produced by the TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS options
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLEOUTLIM	Augments basic table with columns indicating decision limits exceeded
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend

Table 5.2 *continued*

Option	Description
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTSUMMARY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels decision limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive ANOM Charts	
HTML=	Specifies a variable whose values create links to be associated with groups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Details: BOXCHART Statement

Constructing ANOM Boxcharts

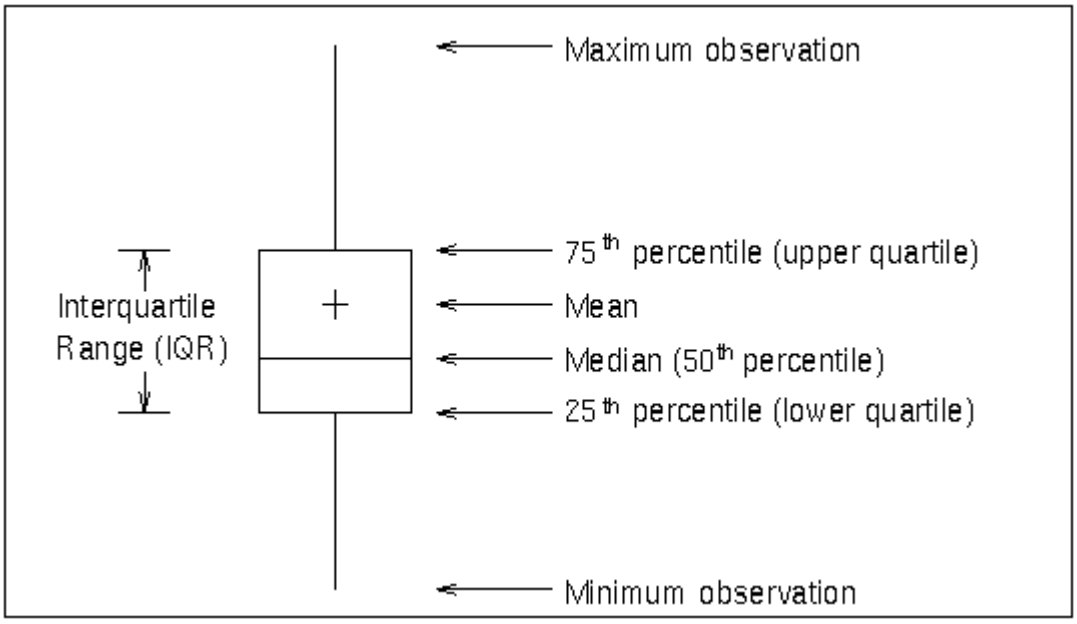
The following notation is used in this section:

X_{ij}	jth response in the ith group
k	Number of groups
n_i	Sample size of ith group
N	Total sample size $= n_1 + \cdots + n_k$
μ_i	Expected value of a response in the ith group
σ	Standard deviation of response
$\bar{X}_i$	Average response in ith group
$\overline{\overline{X}}$	Weighted average of k group means
s_i^2	Sample variance of the responses in the ith group
$\widehat{\sigma^2}$	Mean square error (MSE)
ν	Degrees of freedom associated with the mean square error
α	Significance level
$h(\alpha; k, n, \nu)$	Critical value for analysis of means when the sample sizes n_i are equal ($n_i \equiv n$)
$h(\alpha; k, n_1, \ldots, n_k, \nu)$	Critical value for analysis of means when the sample sizes n_i are not equal

Elements of Box-and-Whisker Plots

A box-and-whisker plot is displayed for the measurements in each group on the ANOM boxchart. Figure 5.9 illustrates the elements of each plot.

Figure 5.9 Box-and-Whisker Plot



The skeletal style of the box-and-whisker plot shown in Figure 5.9 is the default. You can specify alternative styles with the BOXSTYLE= option; see the entry for the BOXSTYLE= option in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Central Line

By default, the central line on an ANOM chart for means represents the weighted average of the group means, which is computed as

$$\overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_k\bar{X}_k}{n_1 + \cdots + n_k}$$

You can specify a value for $\overline{\overline{X}}$ with the MEAN= option in the BOXCHART statement or with the variable _MEAN_ in a LIMITS= data set.

Decision Limits

In the analysis of means for continuous data, it is assumed that the responses in the ith group are at least approximately normally distributed with a constant variance:

$$X_{ij} \sim N(\mu_i, \sigma^2), \qquad j = 1, \ldots, n_i$$

When the group sizes are constant ($n_i \equiv n$), then $\nu = N - k = k(n-1)$ and the decision limits are computed as follows:

$$\begin{aligned} \text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}} \end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{1}{k}\sum_{j=1}^{k} s_j^2$$

For details concerning the function $h(\alpha; k, n, \nu)$, see Nelson (1981, 1982a, 1993).

When the group sizes n_i are not constant (the unbalanced case), $\nu = N - k$ and the decision limits for the ith group are computed as follows:

$$\begin{aligned} \text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}} \end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{(n_1 - 1)s_1^2 + \cdots + (n_k - 1)s_k^2}{n_1 + \cdots + n_k - k}$$

This requires that ν be positive. A chart is not produced if $\nu > 0$ but MSE is equal to zero (unless you specify the ZEROSTD option). For details concerning the function $h(\alpha; k, n_1, \ldots, n_k, \nu)$, see Fritzsch and Hsu (1997), Nelson (1982b, 1991), and Soong and Hsu (1997).

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, $\alpha = 0.05$.
- Specify a constant nominal sample size $n_i \equiv n$ for the decision limits in the balanced case with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. By default, n is the observed sample size in the balanced case.
- Specify k with the LIMITK= option or with the variable _LIMITK_ in a LIMITS= data set. By default, k is the number of groups.
- Specify $\overline{\overline{X}}$ with the MEAN= option or with the variable _MEAN_ in a LIMITS= data set. By default, $\overline{\overline{X}}$ is the weighted average of the responses.
- Specify $\widehat{\sigma^2}$ with the MSE= option or with the variable _MSE_ in a LIMITS= data set. By default, $\widehat{\sigma^2}$ is computed as indicated above.
- Specify ν with the DFE= option or with the variable _DFE_ in a LIMITS= data set. By default, ν is determined as indicated above.

Output Data Sets

OUTBOX= Data Set

The OUTBOX= data set saves group summary statistics, decision limits, and outlier values. The following variables can be saved:

- the *group-variable*
- the variable _VAR_, containing the analysis variable name
- the variable _TYPE_, identifying features of box-and-whisker plots
- the variable _VALUE_, containing values of box-and-whisker plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing links associated with box-and-whisker plot features

_ID_ is included in the OUTBOX= data set only if one of the keywords SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if the HTML= or HTML2= option is specified.

Each observation in an OUTBOX= data set records the value of a single feature of one group's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. Table 5.4 lists the valid _TYPE_ variable values:

Table 5.4 Valid _TYPE_ Values in an OUTBOX= Data Set

TYPE Value	Description
N	Group size
ALPHA	Significance level
LIMITN	Nominal sample size associated with decision limits
LDLX	Lower decision limit for group mean
UDLX	Upper decision limit for group mean
RESPMEAN	Overall response variable mean
MIN	Group minimum value
Q1	Group first quartile
MEDIAN	Group median
MEAN	Group mean
Q3	Group third quartile
MAX	Group maximum value
LOW	Low outlier value
HIGH	High outlier value
LOWHISKR	Low whisker value, if different from MIN
HIWHISKR	High whisker value, if different from MAX
FARLOW	Low far outlier value
FARHIGH	High far outlier value

Additionally, the following variables, if specified, are included:

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 5.5 OUTLIMITS= Data Set

Variable	Description
ALPHA	Significance level
DFE	Degrees of freedom for mean square error
GROUP	*Group-variable* specified in the BOXCHART statement
INDEX	Optional identifier for the decision limits specified with the OUTINDEX= option
LDLX	Lower decision limit for group means
LIMITK	Number of groups
LIMITN	Sample size associated with the decision limits
MEAN	Weighted average of group means ($\overline{\overline{X}}$)
MSE	Mean square error
TYPE	Type (estimate or standard value) of _MEAN_ and _MSE_
UDLX	Upper decision limit for group means
VAR	*Response* specified in the BOXCHART statement

Notes:

1. In the unbalanced case, the special missing value *V* is assigned to the variables _LIMITN_, _LDLX_, and _UDLX_.
2. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the BOXCHART statement. For an example, see "Saving Decision Limits" on page 51.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables can be saved:

- the *group-variable*
- a group minimum variable named by *response* suffixed with *L*
- a group first-quartile variable named by *response* suffixed with *1*
- a group mean variable named by *response* suffixed with *X*
- a group median variable named by *response* suffixed with *M*
- a group third-quartile variable named by *response* suffixed with *3*
- a group maximum variable named by *response* suffixed with *H*
- a group standard deviation variable named by *response* suffixed with *S*
- a group sample size variable named by *response* suffixed with *N*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the BOXCHART statement. For example, consider the following statements:

```
proc anom data=Steel;
   xchart (Width Diameter)*Lot / outsummary=Summary;
run;
```

The data set Summary contains variables named Lot, WidthL, Width1, WidthX, WidthM, Width3, WidthH, WidthS, WidthN, DiameterL, Diameter1, DiameterX, DiameterM, Diameter3, DiameterH, DiameterS, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTSUMMARY= data set, see "Saving Summary Statistics for Groups" on page 50.

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	Significance level
EXLIM	Decision limit exceeded (if any)
Group	Values of the group variable
LDLX	Lower decision limit for group mean
LIMITN	Nominal sample size associated with the decision limits
MEAN	Central line
SUBMAX	Group maximum
SUBMED	Group median
SUBMIN	Group minimum
SUBN	Group sample size
SUBQ1	Group first quartile
SUBQ3	Group third quartile
SUBX	Group mean
UDLX	Upper decision limit for group mean
VAR	*Response* specified in the BOXCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

NOTE: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Decision Limits" on page 51.

ODS Tables

The following table summarizes the ODS tables that you can request with the BOXCHART statement.

Table 5.7 ODS Tables Produced with the BOXCHART Statement

Table Name	Description	Options
BoxChartSummary	ANOM chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. BOXCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the BOXCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 5.8.

Table 5.8 ODS Graphics Produced by the BOXCHART Statement

ODS Graph Name	Plot Description
BoxChart	ANOM boxchart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

BOX= Data Set

You can read summary statistics, decision limits, and outlier values from a BOX= data set specified in the PROC ANOM statement. This enables you to reuse an OUTBOX= data set created in a previous run of the ANOM procedure to display a box chart.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the analysis variable name
- _TYPE_, identifying features of box-and-whisker plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one group's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. Table 5.9 lists valid the _TYPE_ variable values:

Table 5.9 Valid _TYPE_ Values in a BOX= Data Set

TYPE Value	Description
N	Group size
ALPHA	Significance level
LIMITN	Nominal sample size associated with decision limits
LDLX	Lower decision limit for group mean
UDLX	Upper decision limit for group mean
RESPMEAN	Overall response variable mean
MIN	Group minimum value
Q1	Group first quartile
MEDIAN	Group median
MEAN	Group mean
Q3	Group third quartile
MAX	Group maximum value
LOW	Low outlier value
HIGH	High outlier value
LOWHISKR	Low whisker value, if different from MIN
HIWHISKR	High whisker value, if different from MAX
FARLOW	Low far outlier value
FARHIGH	High far outlier value

The features identified by _TYPE_ values N, LDLX, UDLX, RESPMEAN, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each group.

Other variables that can be read from a BOX= data set include:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing links to be associated with features on box plots
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_ does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

DATA= Data Set

You can read raw data (response values) from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the BOXCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into group samples indexed by the *group-variable*. The *group-variable*, which is specified in the BOXCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *response* and a value for the *group-variable*. If the *i*th group contains n_i items, there should be n_i consecutive observations for which the value of the *group-variable* is the index of the *i*th group. For example, if each group contains five items and there are 10 groups, the DATA= data set should contain 50 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option.

For an example of a DATA= data set, see "Creating ANOM Boxcharts from Response Values" on page 45.

LIMITS= Data Set

You can read preestablished decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set Conlims:

```
proc anom data=Info limits=Conlims;
   xchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 5.5. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LDLX_, _MEAN_, and _UDLX_, which specify the decision limits directly
- the variables _MEAN_, _MSE_, and _DFE_, which are used to calculate the decision limits according to the equations in the section "Decision Limits" on page 63.

In addition, note the following:

- The variables _VAR_ and _GROUP_ are required. These must be character variables whose lengths are no greater than 32.
- _DFE_ is optional. The default is $\nu = N - k$, and in the case of equal group sizes, $\nu = k(n-1)$.
- _MSE_ is optional if _LDLX_ and _UDLX_ are specified; otherwise it is required.
- _LDLX_ and _UDLX_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are 'ESTIMATE,' 'STANDARD,' 'STDMEAN,' and 'STDRMS.' The default is 'ESTIMATE.'
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

A SUMMARY= data set used with the BOXCHART statement must contain the following:

- the *group-variable*
- a group minimum variable for each *response*
- a group first-quartile variable for each *response*
- a group mean variable for each *response*
- a group median variable for each *response*
- a group third-quartile variable for each *response*
- a group maximum variable for each *response*
- a group standard deviation variable for each *response*
- a group sample size variable for each *response*

The names of the group summary statistics variables must be the *response* name concatenated with the following special suffix characters:

Group Summary Statistic	Suffix Character
Group minimum	L
Group first-quartile	1
Group median	M
Group mean	X
Group third-quartile	3
Group maximum	H
Group standard deviation	S
Group sample size	N

For example, consider the following statements:

```
proc anom summary=Summary;
   xchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightL, Weight1, WeightX, WeightM, Weight3, WeightH, WeightS, WeightN, YieldstrengthL, Yieldstrength1, YieldstrengthX, YieldstrengthM, Yieldstrength3, YieldstrengthH, YieldstrengthS, and YieldstrengthN. Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a SUMMARY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option.

For an example of a SUMMARY= data set, see "Creating ANOM Boxcharts from Group Summary Data" on page 47.

TABLE= Data Set

You can read summary statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure. Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

Table 5.10 lists the variables required in a TABLE= data set used with the BOXCHART statement:

Table 5.10 Variables Required in a TABLE= Data Set

Variable	Description
Group-variable	Values of the *group-variable*
LDLX	Lower decision limit for mean
LIMITN	Nominal sample size associated with the decision limits
MEAN	Central line
SUBMAX	Group maximum
SUBMED	Group median
SUBMIN	Group minimum
SUBN	Group sample size
SUBQ1	Group first quartile
SUBQ3	Group third quartile
SUBX	Group mean
UDLX	Upper decision limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*

- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Decision Limits" on page 51.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*Group-variable*
Vertical	DATA=	*Response*
Vertical	SUMMARY=	Group mean variable
Vertical	TABLE=	_SUBX_

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. Missing values of response variables generally lead to unequal group sample sizes. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: BOXCHART Statement

This section provides an advanced example of the BOXCHART statement.

Example 5.1: ANOM Boxcharts with Unequal Group Sizes

NOTE: See *ANOM BOXCHARTS With Unequal Group Sizes* in the SAS/QC Sample Library.

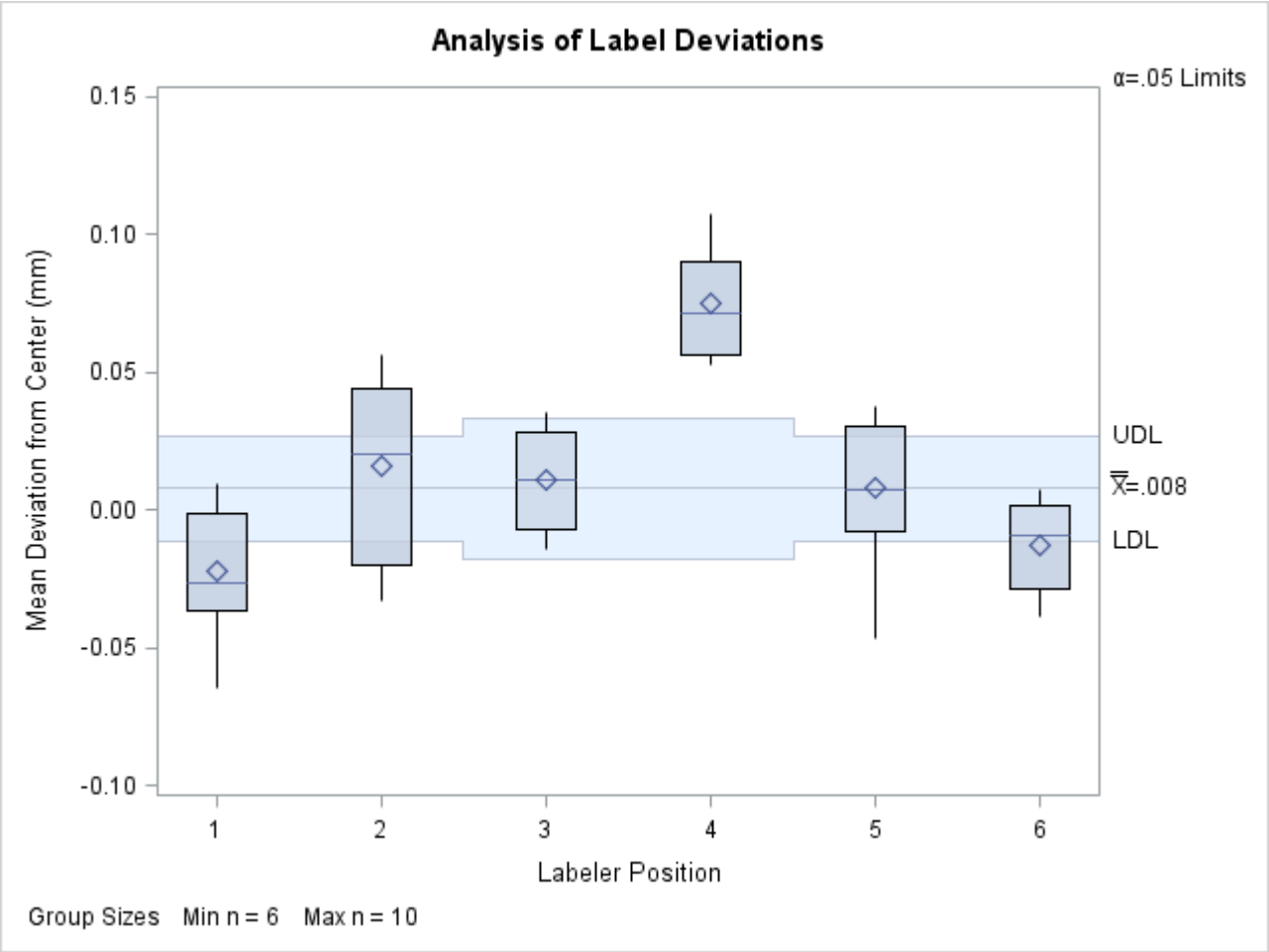
Consider the example described in "Creating ANOM Boxcharts from Response Values" on page 45. Suppose that four of the 10 measurements were missing for the third and fourth labeler positions. The following statements create a SAS data set named LabelDev2, which contains the resulting deviation measurements:

```
data LabelDev2;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.0239  -0.0285  -0.0300  -0.0043  -0.0362
1  -0.0422  -0.0014  -0.0647   0.0094  -0.0016
2  -0.0201  -0.0273   0.0227  -0.0332   0.0366
2   0.0438   0.0556   0.0098   0.0564   0.0182
3  -0.0073   0.0285    .        .      -0.0139
3    .       0.0357   0.0235    .      -0.0020
4   0.0669   0.1073    .        .       0.0755
4    .       0.0561   0.0899    .       0.0530
5   0.0368   0.0036   0.0374   0.0116  -0.0074
5   0.0250  -0.0080   0.0302  -0.0015  -0.0464
6   0.0049  -0.0384  -0.0204  -0.0049  -0.0120
6   0.0071  -0.0308   0.0017  -0.0285  -0.0070
;
```

The following statements create the ANOM chart shown in Output 5.1.1:

```
ods graphics on;
title 'Analysis of Label Deviations';
proc anom data=LabelDev2;
   boxchart Deviation*Position / odstitle=title;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

Output 5.1.1 ANOM Chart with Unequal Group Sizes



Note that the decision limits are automatically adjusted for the varying group sizes. The legend reports the minimum and maximum group sizes.

PCHART Statement: ANOM Procedure

Overview: PCHART Statement

The PCHART statement creates ANOM charts for group (treatment level) proportions, also referred to as ANOM *p charts*.

You can use options in the PCHART statement to

- compute decision limits from the data based on specified parameters, such as the significance level (α)
- tabulate group sample sizes, group proportions, decision limits, and other information
- save decision limits in an output data set
- save group sample sizes and group proportions in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups on the same chart
- add block legends and symbol markers to identify special groups
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have two alternatives for producing ANOM *p* charts with the PCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced if SAS/GRAPH® is licensed.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: PCHART Statement

This section introduces the PCHART statement with simple examples that illustrate commonly used options. Complete syntax for the PCHART statement is presented in the section "Syntax: PCHART Statement" on page 85.

Creating ANOM Charts for Proportions from Group Counts

NOTE: See *Creating ANOM p Charts from Group Counts* in the SAS/QC Sample Library.

A health care system administrator uses ANOM to compare cesarean section rates for a set of medical groups. For more background concerning this application, refer to Rodriguez (1996).

The following statements create a SAS data set named Csection, which contains the number of c-sections and the total number of deliveries for each medical group over a one-year period.

```
data Csection;
   length ID $ 2;
   input ID Csections Total @@;
   label ID = 'Medical Group Identification Number';
   datalines;
1A  150  923 1K   45  298 1B   34  170 1D   18  132
3I   20  106 3M   12  105 1E   10   77 1N   19   74
1Q    7   69 3H   11   65 1R   11   49 1H    9   48
3J    7   20 1C    8   43 3B    6   43 1M    4   29
3C    5   28 1O    4   27 1J    6   22 1T    3   22
3E    4   18 1G    4   15 3D    4   13 3G    1   11
1L    2   10 1I    1    8 1P    0    3 1F    0    3
1S    1    3
;
```

A partial listing of Csection is shown in Figure 5.10.

Figure 5.10 The Data Set Csection

Cesarean Section Data

ID	Csections	Total
1A	150	923
1K	45	298
1B	34	170
1D	18	132
3I	20	106
3M	12	105
1E	10	77
1N	19	74
1Q	7	69
3H	11	65

The variable ID identifies the medical groups and is referred to as the *group-variable*. The variable Csections provides the number of c-sections, and is referred to as the *response variable* (or *response* for short). The variable Total provides the total number of deliveries.

The following statements create the *p* chart shown in Figure 5.11:

```
ods graphics off;
title 'Analysis of C-Sections';
proc anom data=Csection;
   pchart Csections*ID / groupn   = Total
                         hoffset  = 2
                         nolegend
                         turnhlabels;
   label Csections = 'Proportion of Cesarean Sections';
run;
```

This example illustrates the basic form of the PCHART statement. After the keyword PCHART, you specify the *response* to analyze (in this case, Csections, followed by an asterisk and the *group-variable* ID.

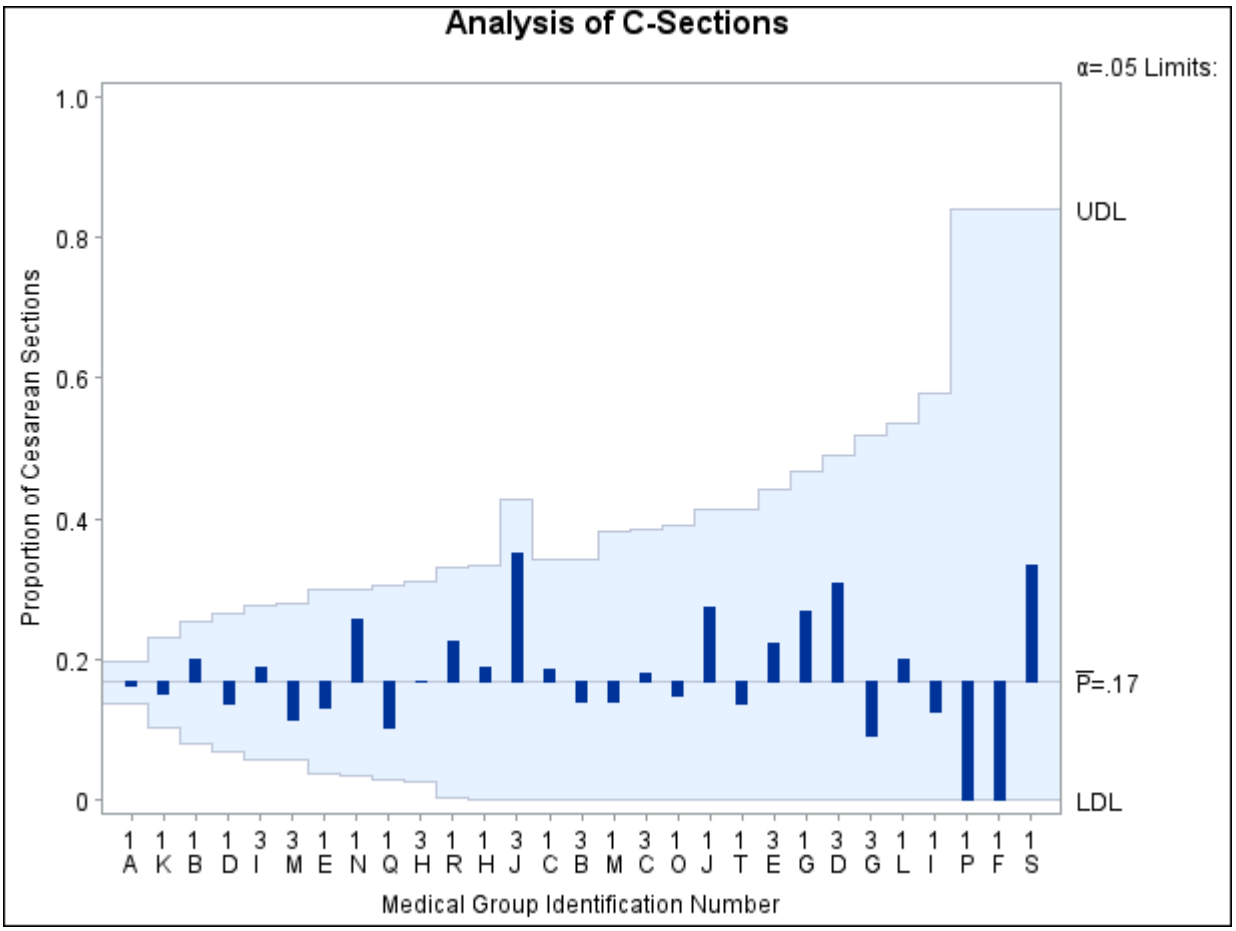
The input data set is specified with the DATA= option in the PROC ANOM statement. The GROUPN= option specifies the sample size in each group and is required with a DATA= input data set. The GROUPN= option specifies one of the following:

- a constant group sample size
- a variable in the input data set whose values provide the group sample sizes (in this case, Total)

The TURNHLABELS option turns the horizontal axis labels since the default labeling skips labels if the characters exceed the space allotted. See Axis and Axis Label Options. To angle the axis labels, see Example 5.2.

Options such as GROUPN= and TURNHLABELS are specified after the slash (/) in the PCHART statement. A complete list of options is presented in the section "Syntax: PCHART Statement" on page 85.

Figure 5.11 ANOM p Chart for Cesarean Sections



Each point on the p chart represents the proportion of c-sections for a particular group. For instance, the value plotted for group 1A is $150/923 = 0.163$.

Since all the points fall within the decision limits, it can be concluded that the variation in proportions of c-sections across medical groups is strictly due to chance.

By default, the decision limits shown correspond to a significance level of $\alpha = 0.05$. This means that, assuming all groups have the same proportion of c-sections, there is a 0.05 probability that one or more of the decision limits would be exceeded purely by chance. The formulas for the limits are given in "Decision Limits" on page 95. Note that the decision limits vary with the number of deliveries in each group, and the widest limits correspond to the group with the smallest number of deliveries.

For more details on reading group counts, see "DATA= Data Set" on page 99.

Creating ANOM Charts for Proportions from Group Summary Data

NOTE: See *Creating ANOM p Charts from Group Summary Data* in the SAS/QC Sample Library.

The previous example illustrates how you can create ANOM charts for proportions using count data. However, in many applications, the group data are provided in summarized form as proportions or percentages. This example illustrates how you can use the PCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data CsectProp;
   length ID $ 2;
   input ID CsectionsP CsectionsN @@;
   datalines;
1A  0.163  923   1K  0.151  298   1B  0.200  170   1D  0.136  132
3I  0.189  106   3M  0.114  105   1E  0.130   77   1N  0.257   74
1Q  0.101   69   3H  0.169   65   1R  0.224   49   1H  0.188   48
3J  0.350   20   1C  0.186   43   3B  0.140   43   1M  0.138   29
3C  0.179   28   1O  0.148   27   1J  0.273   22   1T  0.136   22
3E  0.222   18   1G  0.267   15   3D  0.308   13   3G  0.091   11
1L  0.200   10   1I  0.125    8   1P  0.000    3   1F  0.000    3
1S  0.333    3
;
```

A partial listing of CsectProp is shown in Figure 5.12. The groups are still indexed by ID. The variable CsectionsP contains the proportions of c-sections, and the variable CsectionsN contains the group sample sizes.

Figure 5.12 The Data Set CsectProp

Proportions of Cesarean Sections

ID	CsectionsP	CsectionsN
1A	0.163	923
1K	0.151	298
1B	0.200	170
1D	0.136	132
3I	0.189	106
3M	0.114	105
1E	0.130	77
1N	0.257	74
1Q	0.101	69
3H	0.169	65

You can analyze this data set by specifying it as a SUMMARY= data set in the PROC ANOM statement.

Note that Csections is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the two SAS variables CsectionsP and CsectionsN. The suffix characters *P* and *N* indicate *proportion* and *sample size*, respectively. Thus, you can specify two group variables in a SUMMARY= data set with a single name Csections, which is referred to as the *response*. The name ID specified after the asterisk is the name of the *group-variable*.

A SUMMARY= data set used with the PCHART statement must contain the following variables:

- group variable
- group proportion variable
- group sample size variable

Furthermore, the names of the group proportion and sample size variables must begin with the *response* name specified in the PCHART statement and end with the special suffix characters *P* and *N*, respectively.

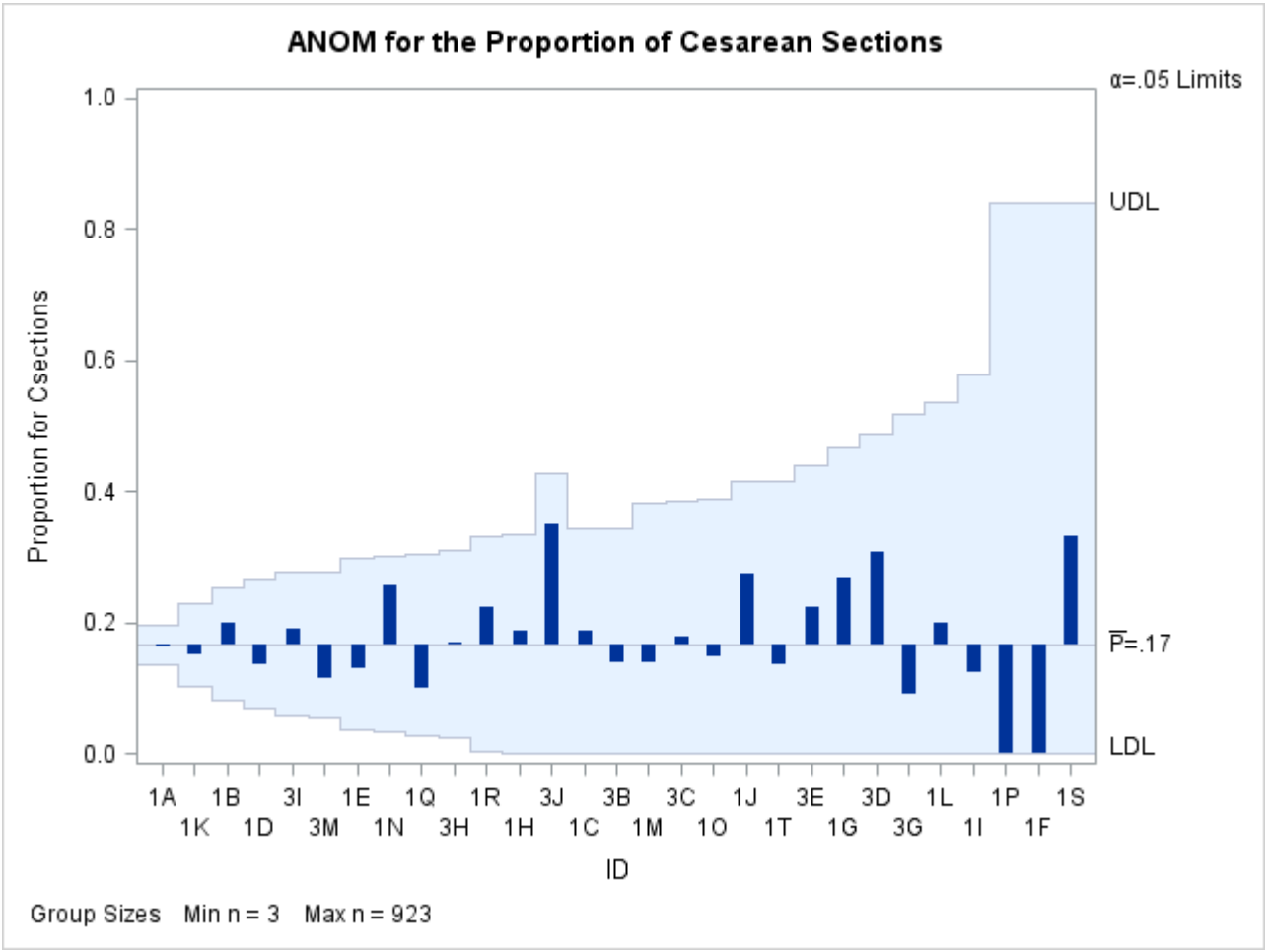
For more information, see "SUMMARY= Data Set" on page 101.

The following statements create a *p* Chart for C-Sections using the SUMMARY= data set CsectProp:

```
ods graphics on;
title 'ANOM for the Proportion of Cesarean Sections';
proc anom summary=CsectProp;
   pchart Csections*ID / odstitle = title1;
run;
```

The ODS GRAPHICS ON statement specified before the PROC ANOM statement enables ODS Graphics, so the *p* chart is created using ODS Graphics instead of traditional graphics. The resulting ANOM *p* chart is shown in Figure 5.13.

Figure 5.13 ANOM *p* Chart from Group Proportions



Saving Group Proportions

NOTE: See *Saving Group Proportions Using ANOM PCHART* in the SAS/QC Sample Library.

In this example, the PCHART statement is used to create a summary data set that can later be read by the ANOM procedure (as in the preceding example). The following statements read the data set CSection (see "Creating ANOM Charts for Proportions from Group Counts" on page 78) and create a summary data set named CSummary:

```
proc anom data=Csection;
   pchart Csections*ID / groupn     = Total
                         outsummary = CSummary
                         nochart;
run;
```

The OUTSUMMARY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 5.11. Figure 5.14 contains a partial listing of CSummary.

Figure 5.14 The Data Set CSummary

Group Proportions and Decision Limit Information

ID	CsectionsP	CsectionsN
1A	0.16251	923
1K	0.15101	298
1B	0.20000	170
1D	0.13636	132
3I	0.18868	106
3M	0.11429	105
1E	0.12987	77
1N	0.25676	74
1Q	0.10145	69
3H	0.16923	65

There are three variables in the data set CSummary:

- ID identifies the groups.
- CSectionsP contains the group proportions.
- CSectionsN contains the group sample sizes.

Note that the variables containing the group proportions and group sample sizes are named by adding the suffix characters *P* and *N* to the *response* CSections specified in the PCHART statement. In other words, the variable naming convention for OUTSUMMARY= data sets is the same as that for SUMMARY= data sets. For more information, see "OUTSUMMARY= Data Set" on page 97.

Saving Decision Limits

NOTE: See *Saving Decision Limits Using ANOM PCHART* in the SAS/QC Sample Library.

You can save the decision limits for an ANOM *p* chart in a SAS data set.

The following statements read the number of c-sections per group from the data set CSection (see "Creating ANOM Charts for Proportions from Group Counts" on page 78) and save the decision limits displayed in Figure 5.11 in a data set named CSectionLim:

```
proc anom data=Csection;
   pchart Csections*ID / groupn    = Total
                         outlimits = CsectionLim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set CSectionLim is listed in Figure 5.15.

Figure 5.15 The Data Set CSectionLim with Decision Limits

Decision Limits for the Proportion of Cesarean Sections

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLP_	_P_	_UDLP_	_LIMITK_
Csections	ID	ESTIMATE	V	0.05	V	0.16680	V	29

The data set CSectionLim contains one observation with the limits for the *response* CSections. The variables _LDLP_ and _UDLP_ contain the lower and upper decision limits, and the variable _P_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the decision limits, the value of _LIMITK_ is the number of groups, and the value of _ALPHA_ is the significance level associated with the decision limits. The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or a known (standard) value. Typically, the value of _TYPE_ is 'ESTIMATE.'

For more information, see the section "OUTLIMITS= Data Set" on page 96.

NOTE: See *Saving ANOM PCHART Summary Statistics and Decision Limits* in the SAS/QC Sample Library.

You can create an output data set containing both decision limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc anom data=Csection;
   pchart Csections*ID / groupn   = Total
                         outtable = CsectionTab
                         nochart;
run;
```

A partial listing of the data set CSectionTab is shown in Figure 5.16.

Figure 5.16 The Data Set CSectionTab

Proportions and Decision Limits for Cesarean Sections

VAR	ID	_ALPHA_	_LIMITN_	_SUBN_	_LDLP_	_SUBP_	_P_	_UDLP_	_EXLIM_
Csections	1A	0.05	923	923	0.13658	0.16251	0.16680	0.19703	
Csections	1K	0.05	298	298	0.10355	0.15101	0.16680	0.23006	
Csections	1B	0.05	170	170	0.08059	0.20000	0.16680	0.25302	
Csections	1D	0.05	132	132	0.06814	0.13636	0.16680	0.26547	
Csections	3I	0.05	106	106	0.05608	0.18868	0.16680	0.27752	
Csections	3M	0.05	105	105	0.05553	0.11429	0.16680	0.27807	
Csections	1E	0.05	77	77	0.03609	0.12987	0.16680	0.29752	
Csections	1N	0.05	74	74	0.03338	0.25676	0.16680	0.30023	
Csections	1Q	0.05	69	69	0.02849	0.10145	0.16680	0.30512	
Csections	3H	0.05	65	65	0.02417	0.16923	0.16680	0.30943	

This data set contains one observation for each group sample. The variables _SUBP_ and _SUBN_ contain the group proportions and group sample sizes. The variables _LDLP_ and _UDLP_ contain the lower and upper decision limits, and the variable _P_ contains the central line. The variables _VAR_ and ID contain the *response* name and values of the *group-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 98.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read the information in CSectionTab and display an ANOM *p* chart (not shown here) identical to the chart in Figure 5.11:

```
title 'Analysis of C-Sections';
proc anom table=CSectionTab;
   pchart CSections*id;
label _subp_ = 'Proportion of Cesarean Sections';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts. For more information, see "TABLE= Data Set" on page 101.

Syntax: PCHART Statement

The basic syntax for the PCHART statement is as follows:

PCHART *response* ∗ *group-variable* **;**

The general form of this syntax is as follows:

PCHART *responses* ∗ *group-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > < *options* > **;**

You can use any number of PCHART statements in the ANOM procedure. The components of the PCHART statement are described as follows.

response
responses

identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If response counts are read from a DATA= data set, *response* must be the name of the variable containing the counts. For an example, see "Creating ANOM Charts for Proportions from Group Summary Data" on page 80.
- If response proportions are read from a SUMMARY= data set, *response* must be the common prefix of the summary variables in the SUMMARY= data set. For an example, see "Creating ANOM Charts for Proportions from Group Summary Data" on page 80.
- If response proportions and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Decision Limits" on page 84.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM *p* charts for the responses Rejects and Reworks:

```
proc anom data=Measures;
   pchart (Rejects Reworks)*Sample / groupn=100;
run;
```

Note that when data are read from a DATA= data set, the GROUPN= option, which specifies group sample sizes, is required.

group-variable

is the variable that identifies groups in the data. The *group-variable* is required. In the preceding PCHART statement, Sample is the group variable.

block-variables

are optional variables that identify sets of consecutive groups on the chart. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker used to plot proportions. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements.

options

control the analysis, enhance the appearance of the chart, save results in data sets, and so on. The section "Summary of Options" lists all options by function.

Summary of Options

The following tables list the PCHART statement options by function. Options unique to the ANOM procedure are listed in Table 5.11, and are described in detail in "Dictionary of ANOM Chart Statement Options" on page 183. Options that are common to both the ANOM and SHEWHART procedures are listed in Table 5.12, and are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 5.11 PCHART Statement Special Options

Option	Description
Options for Specifying Decision Limits	
ALPHA=	Specifies significance level
LIMITK=	Specifies number of groups for decision limits
LIMITN=	Specifies either nominal sample size for fixed decision limits or varying limits
NOREADLIMITS	Computes decision limits for each *response* from the data rather than a LIMITS= data set
P=	Specifies the weighted average of group proportions
READINDEXES=	reads multiple sets of decision limits for each *response* from a LIMITS= data set
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Displaying Decision Limits	
CINFILL=	Specifies color for area inside decision limits
CLIMITS=	Specifies color of decision limits, central line, and related labels
LDLLABEL=	Specifies label for lower decision limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=	Specifies line type for decision limits
NDECIMAL=	Specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	Suppresses display of central line
NOLDL	Suppresses display of lower decision limit
NOLIMIT0	Suppresses display of lower decision limit if it is 0
NOLIMIT1	Suppresses display of upper decision limit if it is 1 (100%)
NOLIMITLABEL	Suppresses labels for decision limits and central line
NOLIMITS	Suppresses display of decision limits
NOLIMITSFRAME	Suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for decision limits
NOUDL	Suppresses display of upper decision limit
PSYMBOL=	Specifies label for central line
UDLLABEL=	Specifies label for upper decision limit
WLIMITS=	Specifies width for decision limits and central line
Input Data Set Option	
GROUPN=	Specifies group sample sizes as constant number *n* or as values of variable in a DATA= data set

Table 5.11 *continued*

Option	Description
Output Data Set Option	
OUTSUMMARY=	Creates output data set containing group summary statistics

Table 5.12 PCHART Statement General Options

Option	Description
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on ANOM *p* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside decision limits
COUTFILL=	Specifies color for shading areas between the connected points and decision limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels
LABELHEIGHT=	Specifies height of labels
NONEEDLES	Suppresses vertical needles connecting points to central line
OUTLABEL=	Labels points outside decision limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value for numeric horizontal axis with date, time, or datetime format

Table 5.12 *continued*

Option	Description
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of ANOM *p* chart
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for ANOM *p* chart
WAXIS=	Specifies width of axis lines
YSCALE=	Scales vertical axis in percent units (rather than proportions)
Plot Layout Options	
ALLN	Plots means for all groups
BILEVEL	Creates ANOM *p* chart using half-screens and half-pages
EXCHART	Creates ANOM *p* chart for a response only when a group mean exceeds the decision limits
INTERVAL=	Natural time interval between consecutive group positions when time, date, or datetime format is associated with a numeric group variable
MAXPANELS=	Maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for group sample sizes
NPANELPOS=	Specifies number of group positions per panel on each chart
REPEAT	Repeats last group position on panel as first group position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart

Table 5.12 *continued*

Option	Description
ZEROSTD	Displays ANOM *p* chart regardless of whether root mean square error is zero
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= option
CVREF=	Specifies color for lines requested by VREF= option
HREF=	Specifies position of reference lines perpendicular to horizontal axis on ANOM *p* chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on ANOM *p* chart
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on ANOM *p* chart
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	Specifies position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to ANOM *p* chart

Table 5.12 *continued*

Option	Description
DESCRIPTION=	Specifies description of ANOM *p* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on chart
NAME=	Specifies name of ANOM *p* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	Draws portions of line segments that connect points outside decision limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between decision limits and connected points lying outside the limits
STARFILL=	Specifies a variable identifying groups of stars filled with different colors
STARS=	Specifies a variable identifying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the decision limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots group points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote

Table 5.12 *continued*

Option	Description
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
OUTFILLTRANSPARENCY=	Specifies decision limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with groups
Input Data Set Option	
DATAUNIT=	Specifies that input values are proportions or percentages rather than counts
Output Data Set Options	
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing decision limits
OUTTABLE=	Creates output data set containing group summary statistics and decision limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of group means, group sample sizes, and decision limits
TABLEALL	Creates all the tables that are produced by the TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS options
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLEOUTLIM	Augments basic table with columns indicating decision limits exceeded
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend

Table 5.12 *continued*

Option	Description
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels decision limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on ANOM *p* chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments

Table 5.12 *continued*

Option	Description
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive ANOM Charts	
HTML=	Specifies a variable whose values create links to be associated with groups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Details: PCHART Statement

Constructing ANOM Charts for Proportions

The following notation is used in this section:

First Column	Second Column
X_i	Response number (count) in the ith group
k	Number of groups
n_i	Sample size of the ith group
N	Total sample size $= n_1 + \cdots + n_k$
p_i	Proportion in the ith group, where $p_i = X_i / n_i$
$\bar{p}$	Weighted average of proportions across groups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_k p_k}{N} = \frac{X_1 + \cdots + X_k}{N}$$
α	Significance level
$h(\alpha; k, n, \infty)$	Critical value for ANOM for normal data in the balanced case ($n_i \equiv n$)

Table 5.13 *continued*

First Column	Second Column
$h(\alpha; k, n_1, \ldots, n_k, \infty)$	Critical value for ANOM for normal data in the unbalanced case

Plotted Points

Each point on an ANOM p chart represents the response proportion ($p_i = X_i / n_i$) for a group.

Central Line

By default, the central line on an ANOM p chart is computed as $\bar{p}$, the weighted average of the group proportions. You can specify $\bar{p}$ with the P= option or with the variable _P_ in a LIMITS= data set.

Decision Limits

For the ith group, the response counts are assumed to have the binomial distribution $B(n_i, p_i)$. The ANOM method for proportions tests the null hypothesis that $p_1 = p_2 = \cdots = p_k$, that is, that the proportions are the same, against the alternative that at least one of the p_i's is different from the average of the k proportions.

The decision limits are computed using the normal approximation to the binomial distribution, which is appropriate when the sample sizes for the groups are large; refer to Ramig (1983). A commonly recommended check for this assumption is that $n_i p_i > 5$ and $n_i(1 - p_i) > 5$ for all the groups. The critical values in the ANOM method for normally distributed data are adapted to the binomial case by using infinite degrees of freedom for the variance.

When the sample sizes are constant across groups ($n_i \equiv n$), the decision limits are computed as follows:

$$\text{lower decision limit (LDL)} = \max\left(\bar{p} - h(\alpha; k, n, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{k-1}{N}}, 0\right)$$

$$\text{upper decision limit (UDL)} = \min\left(\bar{p} + h(\alpha; k, n, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{k-1}{N}}, 1\right)$$

For the theoretical derivation of the decision limits, refer to Nelson (1982a).

When the sample sizes (n_i) are different across groups (the unbalanced case), the decision limits are computed as follows:

$$\text{lower decision limit (LDL)} = \max\left(\bar{p} - h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{N-n_i}{Nn_i}}, 0\right)$$

$$\text{upper decision limit (UDL)} = \min\left(\bar{p} + h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{p}(1-\bar{p})}\sqrt{\frac{N-n_i}{Nn_i}}, 1\right)$$

Note that the decision limits for the ith group depend on n_i. If the sample sizes are constant across groups ($n_i \equiv n$), the decision limits in the unbalanced case reduce to the formulas given for the balanced case since $n_i = n$ and $N = kn$ so

$$\sqrt{\frac{N - n_i}{N n_i}} = \sqrt{\frac{kn - n}{Nn}} = \sqrt{\frac{k-1}{N}}$$

For the derivation of the decision limits for unequal sample sizes, refer to Nelson (1991).

Exact critical values $h(\alpha; k, n, \infty)$ were first tabulated by L. S. Nelson (1983). Refer to Nelson (1993) for derivation of critical values.

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, $\alpha = 0.05$.
- Specify a constant nominal sample size $n_i \equiv n$ for the decision limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. By default, n is the observed sample size in the balanced case.
- Specify $\bar{p}$ with the P= option or with the variable _P_ in a LIMITS= data set. By default, $\bar{p}$ is the weighted average of the group proportions.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 5.14 OUTLIMITS= Data Set

Variable	Description
ALPHA	Significance level (α)
GROUP	*Group-variable* specified in the PCHART statement
INDEX	Optional identifier for the decision limits specified with the OUTINDEX= option
LDLP	Lower decision limit for proportions
LIMITK	Number of groups
LIMITN	Nominal sample size associated with the decision limits
_P_	Average proportion of nonconforming items ($\bar{p}$)
TYPE	Type (standard or estimate) of _P_
UDLP	Upper decision limit for proportions
VAR	*Response* specified in the PCHART statement

Notes:

1. If the decision limits vary with group sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LDLP_, and _UDLP_.
2. A group must have at least one nonmissing value ($n_i \geq 1$), and there must be at least one group with $n_i \geq 2$.
3. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the PCHART statement. For an example, see "Saving Decision Limits" on page 84.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables are saved:

- the *group-variable*
- a group proportion variable named by *response* suffixed with *P*
- a group sample size variable named by *response* suffixed with *N*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the PCHART statement. For example, consider the following statements:

```
proc anom data=Input;
   pchart (Rework Rejected)*Batch / outsummary=Summary
                                    groupn =30;
run;
```

The data set Summary contains variables named Batch, ReworkP, ReworkN, RejetedP, and RejetedN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTSUMMARY= data set, see "Saving Group Proportions" on page 83.

Note that an OUTSUMMARY= data set created with the PCHART statement can be reused as a SUMMARY= data set.

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. Table 5.15 lists the variables that are saved:

Table 5.15 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Significance level (α)
EXLIM	Decision limit exceeded on p chart
Group	Values of the group variable
LDLP	Lower decision limit for proportions
LIMITN	Nominal sample size associated with the decision limits
SUBP	Group proportion
SUBN	Group sample size
UDLP	Upper decision limit for proportions
VAR	*Response* specified in the PCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

NOTE: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Decision Limits" on page 84.

ODS Tables

The following table summarizes the ODS tables that you can request with the PCHART statement.

Table 5.16 ODS Tables Produced with the PCHART Statement

Table Name	Description	Options
PChartSummary	p chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. PCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the PCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 5.17.

Table 5.17 ODS Graphics Produced by the PCHART Statement

ODS Graph Name	Plot Description
PChart	ANOM *p* chart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read count data from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the PCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for group samples indexed by the values of the *group-variable*. The *group-variable*, which is specified in the PCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *response* and a value for the *group-variable*. The data set must contain one observation for each group. Note that you can specify the DATAUNIT= option in the PCHART statement to read proportions or percentages instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the PCHART statement, the GROUPN= option (which specifies the group sample size) is required.

For an example of a DATA= data set, see "Creating ANOM Charts for Proportions from Group Counts" on page 78.

LIMITS= Data Set

You can read preestablished decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set Conlims:

```
proc anom data=Info limits=Conlims;
   pchart Rejects*Batch / groupn= 100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LDLP_, _P_, and _UDLP_, which specify the decision limits directly
- the variable _P_, without providing _LDLP_ and _UDLP_. The value of _P_ is used to calculate the decision limits according to the equations in the section "Decision Limits" on page 95.

In addition, note the following:

- The variables _VAR_ and _GROUP_ are always required. These must be character variables whose lengths are no greater than 32.
- _LDLP_ and _UDLP_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are 'ESTIMATE,' 'STANDARD,' 'STDMEAN,' and 'STDRMS.' The default is 'ESTIMATE.'
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to create your own SUMMARY= data set.

A SUMMARY= data set used with the PCHART statement must contain the following:

- the *group-variable*
- a group proportion variable for each *response*
- a group sample size variable for each *response*

The names of the proportion sample size variables must be the *response* name concatenated with the special suffix characters *P* and *N*, respectively.

For example, consider the following statements:

```
proc anom summary=Summary;
   pchart (Rework Rejected)*Batch / groupn=50;
run;
```

The data set Summary must include the variables Batch, ReworkP, ReworkN, RejetedP, and RejetedN.

Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

For an example of a SUMMARY= data set, see "Creating ANOM Charts for Proportions from Group Summary Data" on page 80.

TABLE= Data Set

You can read summary statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure. Because the ANOM procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

Table 5.18 lists the variables required in a TABLE= data set used with the PCHART statement.

Table 5.18 Variables Required in a TABLE= Data Set

Variable	Description
Group-variable	Values of the *group-variable*
LDLP	Lower decision limit for proportions
LIMITN	Nominal sample size associated with the decision limits
_P_	Average proportion of nonconforming items
SUBN	Group sample size
SUBP	Group proportion of nonconforming items
UDLP	Upper decision limit for proportions

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Decision Limits" on page 84.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	All	*Group-variable*
Vertical	DATA=	*Response*
Vertical	SUMMARY=	Group proportion variable
Vertical	TABLE=	_SUBP_

For example, the following sets of statements specify the label *Proportion Nonconforming* for the vertical axis of the *p* chart:

```
proc anom data=Circuits;
   pchart Fail*Batch / groupn=50;
   label Fail = 'Proportion Nonconforming';
run;
```

```
proc anom summary=Cirhist;
   pchart Fail*Batch ;
   label Failp = 'Proportion Nonconforming';
run;

proc anom table=Cirtable;
   pchart Fail*batch ;
   label _SUBP_ = 'Proportion Nonconforming';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. Missing values of response variables generally lead to unequal group sample sizes. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: PCHART Statement

This section provides advanced examples of the PCHART statement.

Example 5.2: ANOM p Charts with Angled Axis Labels

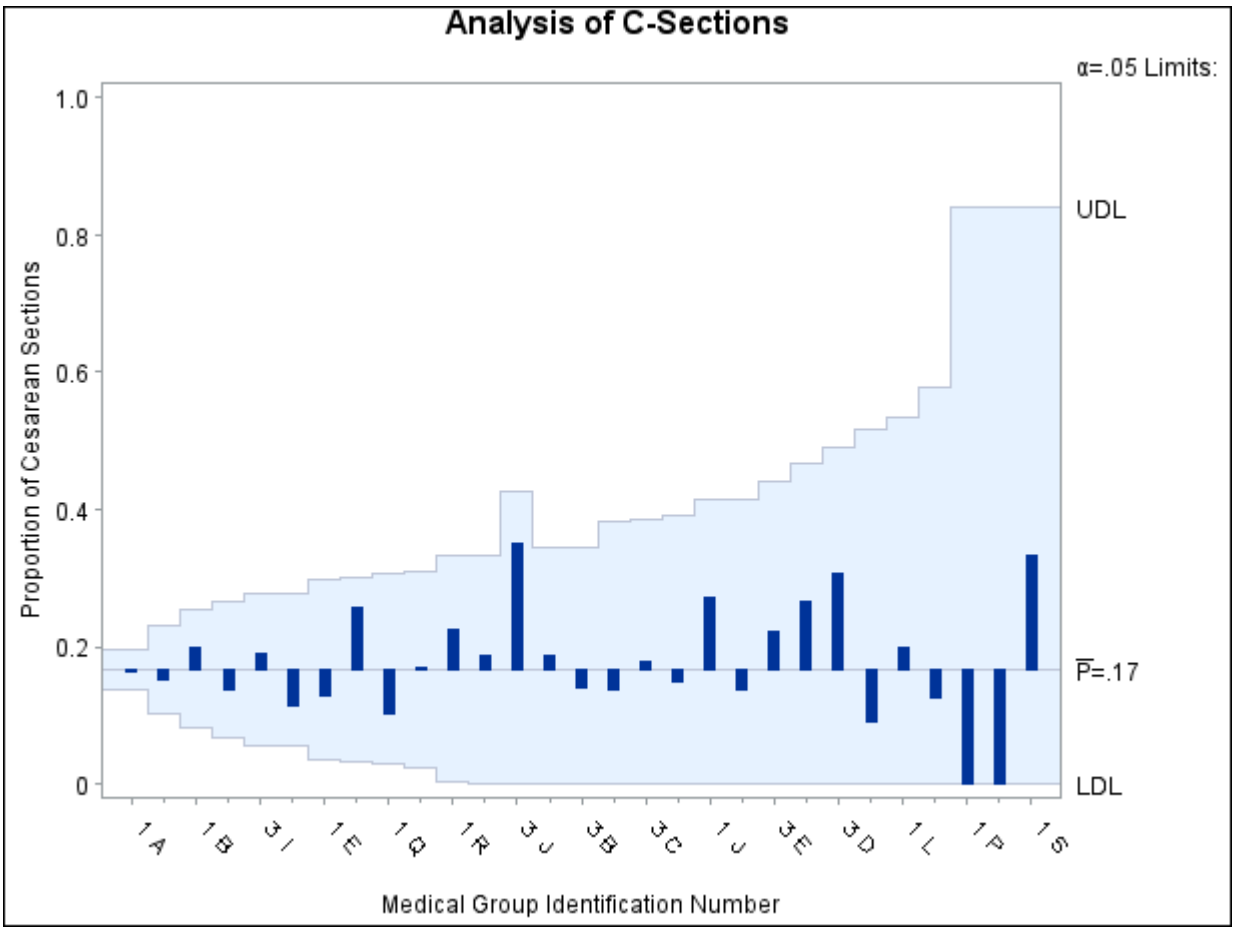
NOTE: See *ANOM p Charts with Angled Axis Labels* in the SAS/QC Sample Library.

Consider the example described in the section "Creating ANOM Charts for Proportions from Group Counts" on page 78. In the example, the option TURNHLABELS was used to vertically display the horizontal axis labels. You can also use an AXIS statement to create an angled display of the horizontal or vertical axis labels. The following statements create the p CHART shown in Output 5.2.1:

```
ods graphics off;
title 'Analysis of C-Sections';
proc anom data=Csection;
   pchart Csections*ID / groupn   = Total
                         nolegend
                         haxis    = axis1;
   axis1 value     = (a=-45 h=2.0pct);
   label Csections = 'Proportion of Cesarean Sections';
run;
```

The angle is specified with the a= option in the AXIS1 statement. Valid angle values are between -90 and 90. The height of the labels is specified with the h= option in the AXIS1 statement. See Axis and Axis Label Options.

Output 5.2.1 ANOM p Chart for C-Sections with Angled Axis Labels



UCHART Statement: ANOM Procedure

Overview: UCHART Statement

The UCHART statement creates ANOM charts for group (treatment level) rates, also referred to as ANOM *u charts*. The rate plotted on a *u* chart is the number or *count* of events occurring in a group divided by a measure of the opportunity for an event to occur.

You can use options in the UCHART statement to

- compute decision limits from the data based on specified parameters, such as the significance level (α)
- tabulate group summary statistics and decision limits
- save decision limits in an output data set
- save group summary statistics in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups on the same chart
- add block legends and symbol markers to identify special groups
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have two alternatives for producing ANOM *u* charts with the UCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced if SAS/GRAPH® is licensed.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: UCHART Statement

This section introduces the UCHART statement with simple examples that illustrate commonly used options. Complete syntax for the UCHART statement is presented in the section "Syntax: UCHART Statement" on page 110.

Creating ANOM Charts for Rates from Group Counts

NOTE: See *Creating ANOM Charts for Rates from Group Counts* in the SAS/QC Sample Library.

A health care system administrator uses ANOM to compare medical/surgical admissions rates for set of clinics. For more background concerning this application, refer to Rodriguez (1996).

The following statements create a SAS data set named MSAdmits, which contains the number of admissions and the number of member-months for each clinic during a one-year period.

```
data MSAdmits;
   length ID $ 2;
   input ID Count MemberMonths @@;
   KMemberYrs = MemberMonths/12000;
   label ID = 'Medical Group Id Number';
   datalines;
1A   1882    697204 1K    600    224715 1B    438    154720
1D    318     82254 3M    183     76450 3I    220     73529
1N    121     60169 3H    105     52886 1Q    124     52595
1E    171     51229 3B     88     34775 1C    100     31959
1H    112     28782 3C     84     27478 1R     69     26494
1T     21     25096 1M    130     24723 1O     61     24526
3D     66     22359 1J     54     19101 3J     30     16089
3G     36     13851 3E     26     10587 1G     28     10351
1I     25      6041 1L     20      5138 1S      7      2723
1F      7      2424 1P      2      2030
;
proc sort data=MSAdmits;
   by ID;
run;
```

A partial listing of MSAdmits is shown in Figure 5.17.

Figure 5.17 The Data Set MSAdmits

Medical/Surgical Admissions

ID	Count	MemberMonths	KMemberYrs
1A	1882	697204	58.1003
1B	438	154720	12.8933
1C	100	31959	2.6633
1D	318	82254	6.8545
1E	171	51229	4.2691
1F	7	2424	0.2020
1G	28	10351	0.8626
1H	112	28782	2.3985
1I	25	6041	0.5034
1J	54	19101	1.5918

There is a single observation per clinic. The variable ID identifies the clinics and is referred to as the *group-variable*. The variable Count provides the number of admissions for each clinic, which is referred to as the *response variable* (or *response* for short). The variable MemberMonths, which provides the number of member-months for each clinic, is divided by 1200 to compute the variable KMemberYrs, the number of 1000-member-years, which serves as the measure of opportunity for an admission to occur.

The following example illustrates the basic form of the UCHART statement. After the keyword UCHART, you specify the *response* to analyze (in this case, Count), followed by an asterisk and the *group-variable* (ID).

The following statements create the *u* chart shown in Figure 5.18:

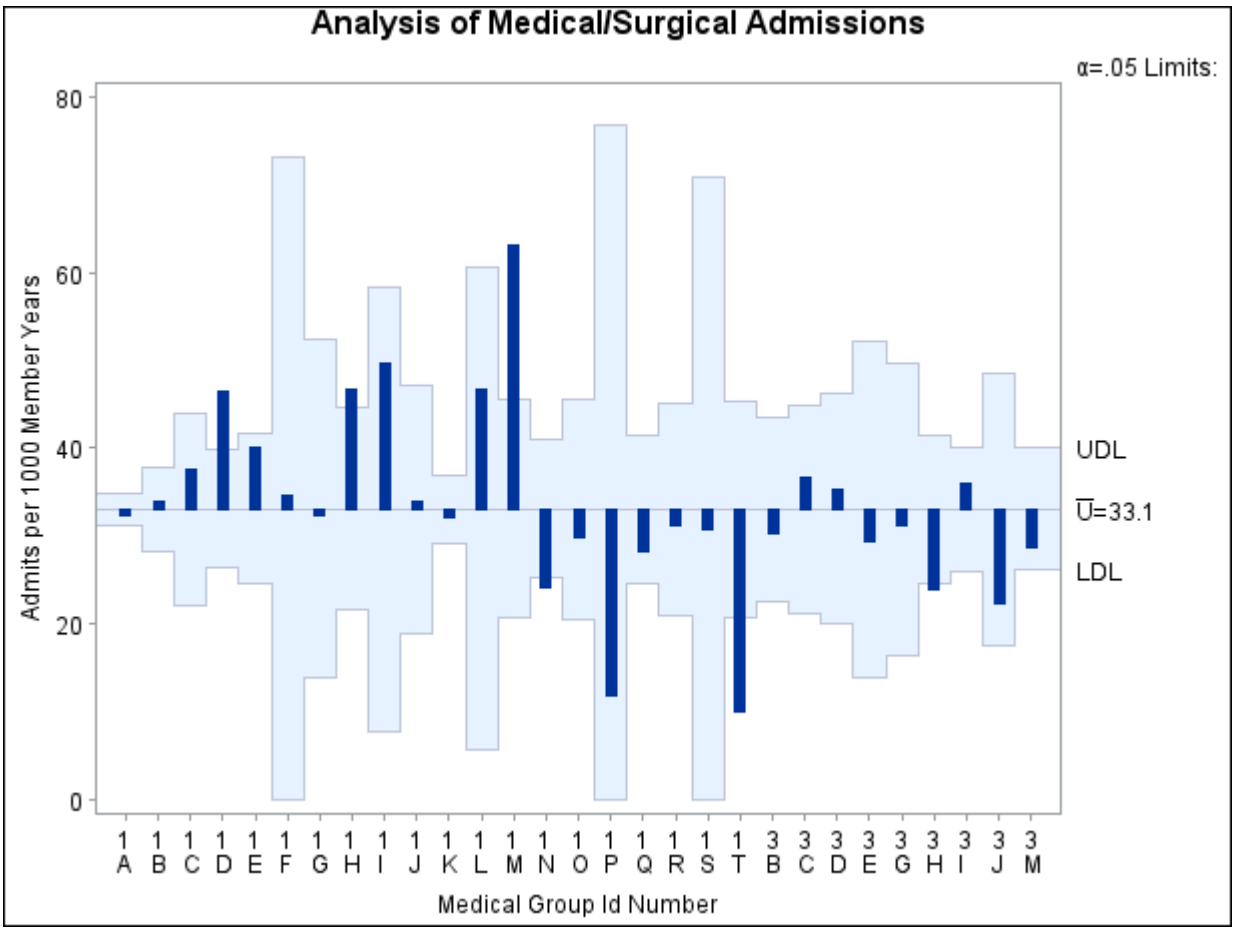
```
ods graphics off;
title 'Analysis of Medical/Surgical Admissions';
proc anom data=MSAdmits;
   uchart Count*ID / groupn = KMemberYrs
                     turnhlabels
                     nolegend;
   label Count = 'Admits per 1000 Member Years';
run;
```

The TURNHLABELS option is used to vertically display the horizontal axis labels. The GROUPN= option specifies the number of "occurrence opportunity" units in each group and is required if the input data set is a DATA= data set. In this example, 1000 member years represent one unit of opportunity. The number of units per group can be thought of as the group "sample size." You can use the GROUPN= option to specify one of the following:

- a constant number of units, which applies to all the groups
- an input variable name, which provides the number of units for each group (KMemberYrs in this example)

Options such as GROUPN= are specified after the slash (/) in the UCHART statement. A complete list of options is presented in the section "Syntax: UCHART Statement" on page 110.

Figure 5.18 u Chart Example



The input data set is specified with the DATA= option in the PROC ANOM statement.

Each point on the u chart represents the rate of occurrence, computed as the count divided by the number of opportunity units. The points are displayed in the sort order for the *group-variable* ID. The chart shows that Clinics 1D, 1H, and 1M have significantly higher admissions rates, and Clinics 1N, 1T, and 3H have significantly lower admissions rates.

By default, the decision limits correspond to a significance level of $\alpha = 0.05$. This means that, assuming all clinics have the same rate of admissions, there is a 0.05 probability that one or more of the decision limits would be exceeded purely by chance. The formulas for the limits are given in the section "Decision Limits" on page 119. Note that the decision limits vary with the number of 1000-member-years for each clinic.

For more details on reading count data, see "DATA= Data Set" on page 123.

Saving Decision Limits

NOTE: See *Saving Decision Limits Using ANOM UCHART* in the SAS/QC Sample Library.

You can save the decision limits for an ANOM u chart in a SAS data set.

The following statements read the data set MSAdmits (see "Creating ANOM Charts for Rates from Group Counts" on page 106) and save the decision limits displayed in Figure 5.18 in a data set named MSLimits:

```
proc anom data=MSAdmits;
   uchart Count*ID / groupn    = KMemberYrs
                     outlimits = MSLimits
                     nochart;
run;
```

The GROUPN= option specifies the number of opportunity units for each group. The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set MSLimits is listed in Figure 5.19.

Figure 5.19 Data Set MSLimits Containing Decision Limits

Decision Limits for Medical/Surgical Admissions Rates

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLU_	_U_	_UDLU_	_LIMITK_
Count	ID	ESTIMATE	V	0.05	V	33.0789	V	29

The data set MSLimits contains one observation with the limits for *response* Count. The variables _LDLU_ and _UDLU_ contain the lower and upper decision limits, and the variable _U_ contains the central line. The value of _LIMITN_ is the nominal number of units associated with the decision limits (which are varying in this case), the value of _LIMITK_ is the number of groups, and the value of _ALPHA_ is the significance level of the decision limits. The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard (known) value. Typically, the value of _TYPE_ is 'ESTIMATE.' For more information, see "OUTLIMITS= Data Set" on page 121.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the decision limits and the group statistics, as illustrated by the following statements:

```
proc anom data=MSAdmits;
   uchart Count*ID / groupn   = KMemberYrs
                     outtable = MSTable
                     nochart;
run;
```

The a partial listing of the data set MSTable is shown in Figure 5.20.

Figure 5.20 Data Set MSTable

Rates and Decision Limits for Medical/Surgical Admissions

VAR	ID	_ALPHA_	_LIMITN_	_SUBN_	_LDLU_	_SUBU_	_U_	_UDLU_	_EXLIM_
Count	1A	0.05	58.1003	58.1003	31.2135	32.3922	33.0789	34.9443	
Count	1B	0.05	12.8933	12.8933	28.2837	33.9710	33.0789	37.8741	
Count	1C	0.05	2.6633	2.6633	22.1550	37.5481	33.0789	44.0028	
Count	1D	0.05	6.8545	6.8545	26.3640	46.3929	33.0789	39.7938	UPPER
Count	1E	0.05	4.2691	4.2691	24.4964	40.0554	33.0789	41.6615	
Count	1F	0.05	0.2020	0.2020	0.0000	34.6535	33.0789	73.0631	
Count	1G	0.05	0.8626	0.8626	13.7710	32.4606	33.0789	52.3868	
Count	1H	0.05	2.3985	2.3985	21.5579	46.6959	33.0789	44.5999	UPPER
Count	1I	0.05	0.5034	0.5034	7.7757	49.6607	33.0789	58.3822	
Count	1J	0.05	1.5918	1.5918	18.8992	33.9249	33.0789	47.2587	

This data set contains one observation for each group. The variables _SUBU_ and _SUBN_ contain the rate of occurrence and the number of opportunity units for each group. The variables _LDLU_ and _UDLU_ contain the lower and upper decision limits, and the variable _U_ contains the central line. The variables _VAR_ and ID contain the *response* name and values of the *group-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 122.

NOTE: See *Saving ANOM UCHART Summary Statistics and Decision Limits* in the SAS/QC Sample Library.

An OUTTABLE= data set can be read later as a TABLE= data set by the ANOM procedure. For example, the following statements read MSTable and display a *u* chart (not shown here) identical to the chart in Figure 5.18:

```
ods graphics off;
title 'Analysis of Medical/Surgical Admissions';
proc anom table=MSTable;
   uchart Count*id ;
   label _subu_ = 'Admits per 1000 Member Years';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts. For more information, see the section "TABLE= Data Set" on page 126.

Syntax: UCHART Statement

The basic syntax for the UCHART statement is as follows:

UCHART *response* * *group-variable* **;**

The general form of this syntax is as follows:

UCHART *responses* * *group-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > < *options* > **;**

You can use any number of UCHART statements in the ANOM procedure. The components of the UCHART statement are described as follows.

response

responses

identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If counts are read from a DATA= data set, *response* must be the name of the variable containing the counts. For an example, see "Creating ANOM Charts for Rates from Group Counts" on page 106.
- If rates and numbers of opportunity units are read from a SUMMARY= data set, *response* must be the common prefix of the appropriate variables in the SUMMARY= data set.
- If rates, numbers of opportunity units, and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM *u* charts for Defects and Flaws:

```
proc anom data=Measures;
   uchart (Defects Flaws)*Sample / groupn=30;
run;
```

Note that when data are read from a DATA= data set with the UCHART statement, the GROUPN= option (which specifies the number of opportunity units per group) is required.

group-variable
is the variable that identifies groups in the data. The *group-variable* is required. In the preceding UCHART statement, sample is the group variable.

block-variables
are optional variables that identify sets of consecutive groups on the chart. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker used to plot the rates. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements.

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function.

Summary of Options

The following tables list the UCHART statement options by function. Options unique to the ANOM procedure are listed in Table 5.19, and are described in detail in "Dictionary of ANOM Chart Statement Options" on page 183. Options that are common to both the ANOM and SHEWHART procedures are listed in Table 5.20, and are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 5.19 UCHART Statement Special Options

Option	Description
Options for Specifying Decision Limits	
ALPHA=	Specifies significance level
LIMITK=	Specifies number of groups for decision limits
LIMITN=	Specifies either nominal sample size for fixed decision limits or varying limits
NOREADLIMITS	Computes decision limits for each *response* from the data rather than a LIMITS= data set
READINDEXES=	Reads multiple sets of decision limits for each *response* from a LIMITS= data set
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U=	Specifies the weighted average of group rates

Table 5.19 *continued*

Option	Description
Options for Displaying Decision Limits	
CINFILL=	Specifies color for area inside decision limits
CLIMITS=	Specifies color of decision limits, central line, and related labels
LDLLABEL=	Specifies label for lower decision limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=	Specifies line type for decision limits
NDECIMAL=	Specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	Suppresses display of central line
NOLDL	Suppresses display of lower decision limit
NOLIMIT0	Suppresses display of lower decision limit if it is 0
NOLIMITLABEL	Suppresses labels for decision limits and central line
NOLIMITS	Suppresses display of decision limits
NOLIMITSFRAME	Suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for decision limits
NOUDL	Suppresses display of upper decision limit
UDLLABEL=	Specifies label for upper decision limit
USYMBOL=	Specifies label for central line
WLIMITS=	Specifies width for decision limits and central line
Input Data Set Option	
GROUPN=	Specifies group sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Option	
OUTSUMMARY=	Creates output data set containing group summary statistics

Table 5.20 UCHART Statement General Options

Option	Description
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on ANOM *u* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line

Table 5.20 *continued*

Option	Description
COUT=	Specifies color for portions of line segments that connect points outside decision limits
COUTFILL=	Specifies color for shading areas between the connected points and decision limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels
LABELHEIGHT=	Specifies height of labels
NONEEDLES	Suppresses vertical needles connecting points to central line
OUTLABEL=	Labels points outside decision limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value for numeric horizontal axis with date, time, or datetime format
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of ANOM *u* chart
VFORMAT=	Specifies format for vertical axis tick mark labels

Table 5.20 *continued*

Option	Description
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for ANOM *u* chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all groups
BILEVEL	Creates ANOM *u* chart using half-screens and half-pages
EXCHART	Creates ANOM *u* chart for a response only when a group mean exceeds the decision limits
INTERVAL=	Specifies the natural time interval between consecutive group positions when time, date, or datetime format is associated with a numeric group variable
MAXPANELS=	Specifies the maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for group sample sizes
NPANELPOS=	Specifies number of group positions per panel on each chart
REPEAT	Repeats last group position on panel as first group position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays ANOM *u* chart regardless of whether root mean square error is zero
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= option
CVREF=	Specifies color for lines requested by VREF= option
HREF=	Specifies position of reference lines perpendicular to horizontal axis on ANOM *u* chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on ANOM *u* chart
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups

Table 5.20 *continued*

Option	Description
VREF=	Specifies position of reference lines perpendicular to vertical axis on ANOM *u* chart
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	Specifies position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to ANOM *u* chart
DESCRIPTION=	Specifies description of ANOM *u* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on chart
NAME=	Specifies name of ANOM *u* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	Draws portions of line segments that connect points outside decision limits in a contrasting color

Table 5.20 *continued*

Option	Description
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between decision limits and connected points lying outside the limits
STARFILL=	Specifies a variable identifying groups of stars filled with different colors
STARS=	Specifies a variable identifying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the decision limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots group points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
OUTFILLTRANSPARENCY=	Specifies decision limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with groups

Table 5.20 *continued*

Option	Description
Output Data Set Options	
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing decision limits
OUTTABLE=	Creates output data set containing group summary statistics and decision limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of group means, group sample sizes, and decision limits
TABLEALL	Creates all the tables that are produced by the TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS options
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLEOUTLIM	Augments basic table with columns indicating decision limits exceeded
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels decision limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set

Table 5.20 *continued*

Option	Description
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on ANOM u chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive ANOM Charts	
HTML=	Specifies a variable whose values create links to be associated with groups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend

Table 5.20 *continued*

Option	Description
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Details: UCHART Statement

Constructing ANOM Charts for Rates

The following notation is used in this section:

c_i	count (number of occurrences) in the ith group
k	number of groups
n_i	number of occurrence opportunity units in the ith group
N	total sample size $= n_1 + \cdots + n_k$
u_i	occurrence rate in the ith group ($u_i = c_i / n_i$)
$\bar{u}$	average of occurrence rates taken across groups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_k u_k}{N} = \frac{c_1 + \cdots + c_k}{N}$$
α	significance level
$h(\alpha; k, n, \infty)$	critical value for ANOM for normal data in the balanced case ($n_i \equiv n$)
$h(\alpha; k, n_1, \ldots, n_k, \infty)$	critical value for ANOM for normal data in the unbalanced case

Plotted Points

Each point on a u chart indicates the rate of occurrence (u_i) in a group.

Central Line

In an ANOM chart for rates, the central line represents the weighted average of the group rates, which is denoted by $\bar{u}$.

Decision Limits

For the ith group, the occurrence counts are assumed to have a Poisson distribution with parameter λ_i. The ANOM method tests the null hypothesis that $\lambda_1 = \cdots = \lambda_k$, that is, that the rates are the same, against the alternative that at least one of the λ_i's is different from the average of the k rates.

The decision limits are computed using the normal approximation to the Poisson distribution, which is appropriate when the sample sizes for the groups are large; see Ramig (1983). A commonly recommended check for this assumption is that $c_i > 5$ for all the groups. The critical values in the ANOM method for normally distributed data are adapted to the Poisson case by using infinite degrees of freedom for the variance.

When the number of opportunity units is constant ($n_i \equiv n$) across groups, the decision limits are computed as follow:

$$\begin{aligned}
\text{lower decision limit (LDLU)} &= \max\left(\bar{u} - h(\alpha; k, n, \infty)\sqrt{\bar{u}}\sqrt{\frac{k-1}{N}}\ , 0\right) \\
\text{upper decision limit (UDLU)} &= \bar{u} + h(\alpha; k, n, \infty)\sqrt{\bar{u}}\sqrt{\frac{k-1}{N}}
\end{aligned}$$

For the theoretical derivation of the decision limits, refer to Nelson (1982a).

When the number of opportunity units (n_i) is different across groups (the unbalanced case), the decision limits are computed as follows:

$$\begin{aligned}
\text{lower decision limit (LDLU)} &= \max\left(\bar{u} - h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{u}}\sqrt{\frac{N-n_i}{Nn_i}}\ , 0\right) \\
\text{upper decision limit (UDLU)} &= \bar{u} + h(\alpha; k, n_1, \ldots, n_k, \infty)\sqrt{\bar{u}}\sqrt{\frac{N-n_i}{Nn_i}}
\end{aligned}$$

Note that the decision limits for the *i*th group depend on n_i. If the sample sizes are constant across groups ($n_i \equiv n$), the decision limits in the unbalanced case reduce to the formulas given for the balanced case, since $n_i \equiv n$ and $N = kn$, so

$$\sqrt{\frac{N-n_i}{Nn_i}} = \sqrt{\frac{kn-n}{Nn}} = \sqrt{\frac{k-1}{N}}$$

For the derivation of the decision limits for unequal sample sizes, refer to Nelson (1991).

Exact critical values were first tabulated by Nelson (1982a). Refer to Nelson (1993) for a derivation of the critical values $h(\alpha; k, n, \infty)$ and Nelson (1991) for a derivation of the critical values $h(\alpha; k, n_1, \ldots, n_k, \infty)$. Note that the critical values in the unequal sample size case have not been tabulated.

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a nominal constant number of opportunity units $n_i \equiv n$ with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify $\bar{u}$ with the U= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. The following variables can be saved:

Table 5.22 OUTLIMITS= Data Set

Variable	Description
ALPHA	Significance level (α)
GROUP	*Group-variable* specified in the UCHART statement
INDEX	Optional identifier for the decision limits specified with the OUTINDEX= option
LDLU	Lower decision limit for occurrence rates
LIMITK	Number of groups
LIMITN	Number of opportunity units associated with the decision limits
TYPE	Type (estimate or standard value) of _U_
_U_	Value of central line of u chart ($\bar{u}$)
UDLU	Upper decision limit for occurrence rates
VAR	*Response* specified in the UCHART statement

Notes:

1. If the decision limits vary with the number of opportunity units, the special missing value V is assigned to the variables _LDLU_, _UDLU_, and _LIMITN_.
2. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the UCHART statement. For an example, see "Saving Decision Limits" on page 108.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables are saved:

- the *group-variable*
- a response rate variable, whose name is *response* suffixed with *U*
- a number of opportunity units variable, whose name is *response* suffixed with *N*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the UCHART statement. For example, consider the following statements:

```
proc anom data=Fabric;
   uchart (Flaws Ndefects)*Treatment / outsummary=Summary
                                       groupn = 30;
run;
```

The data set summary contains the variables Treatment, FlawsU, FlawsN, NdefectsU, and NdefectsN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. Table 5.23 lists the variables that are saved.

Table 5.23 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Significance level (α)
EXLIM	Decision limit exceeded on u chart
Group	Values of the group variable
LDLU	Lower decision limit for group rate
LIMITN	Nominal number of opportunity units associated with the decision limits
SUBU	Group rate
SUBN	Number of opportunity units in group
UDLU	Upper decision limit for group rate
VAR	*Response* specified in the UCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

NOTE: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

ODS Tables

The following table summarizes the ODS tables that you can request with the UCHART statement.

Table 5.24 ODS Tables Produced with the UCHART Statement

Table Name	Description	Options
UChartSummary	ANOM *u* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. UCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the UCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 5.25.

Table 5.25 ODS Graphics Produced by the UCHART Statement

ODS Graph Name	Plot Description
UChart	ANOM *u* chart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read response counts for groups from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the UCHART statement must be a SAS variable in the data set. This variable provides the count (number of occurrences) for groups indexed by the *group-variable*. The *group-variable*, specified in the UCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *response* and a value for the *group-variable*. The data set should contain one observation per group. When you use a DATA= data set with the UCHART statement, the

GROUPN= option (which specifies the number of inspection units per group) is required. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

For an example of a DATA= data set, see "Creating ANOM Charts for Rates from Group Counts" on page 106.

LIMITS= Data Set

You can read decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set Conlims:

```
proc anom data=Info limits=Conlims;
   uchart Defects*Treatment / groupn = 30;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LDLU_, _U_, and _UDLU_, which specify the decision limits
- the variable _U_, without providing the variables _LDLU_ and _UDLU_, which is used to calculate the decision limits (see "Decision Limits" on page 119)

In addition, note the following:

- The variables _VAR_ and _GROUP_ are always required. These must be character variables whose lengths are no greater than 32.
- _LDLU_ and _UDLU_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.

- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are 'ESTIMATE,' 'STANDARD,' 'STDMEAN,' and 'STDRMS.' The default is 'ESTIMATE.'
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to read output data sets created with SAS summarization procedures.

A SUMMARY= data set used with the UCHART statement must contain the following variables:

- *group-variable*
- response rates for each *response*
- number of occurrence units for each *response*

The names of the variables containing the rates and number of occurrence units must be the *response* name concatenated with the special suffix characters *U* and *N*, respectively. For example, consider the following statements:

```
proc anom summary=Summary;
   uchart (Flaws Ndefects)*Treatment;
run;
```

The data set Summary must include the variables Treatment, FlawsU, FlawsN, NdefectsU, and NdefectsN.

Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

TABLE= Data Set

You can read group statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure or to create your own TABLE= data set. Because the ANOM procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

Table 5.26 lists the variables required in a TABLE= data set used with the UCHART statement.

Table 5.26 Variables Required in a TABLE= Data Set

Variable	Description
Group-variable	Values of the *group-variable*
LDLU	Lower decision limit for rate
LIMITN	Nominal number of opportunity units associated with the decision limits
SUBN	Number of opportunity units in group
SUBU	Response rate for group
_U_	Weighted average of group rates
UDLU	Upper decision limit for rate

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Decision Limits" on page 108.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*Group-variable*
Vertical	DATA=	*Response*
Vertical	SUMMARY=	Group defects per unit variable
Vertical	TABLE=	_SUBU_

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: UCHART Statement

This section provides an advanced example of the UCHART statement.

Example 5.3: ANOM u Charts with Angled Axis Labels

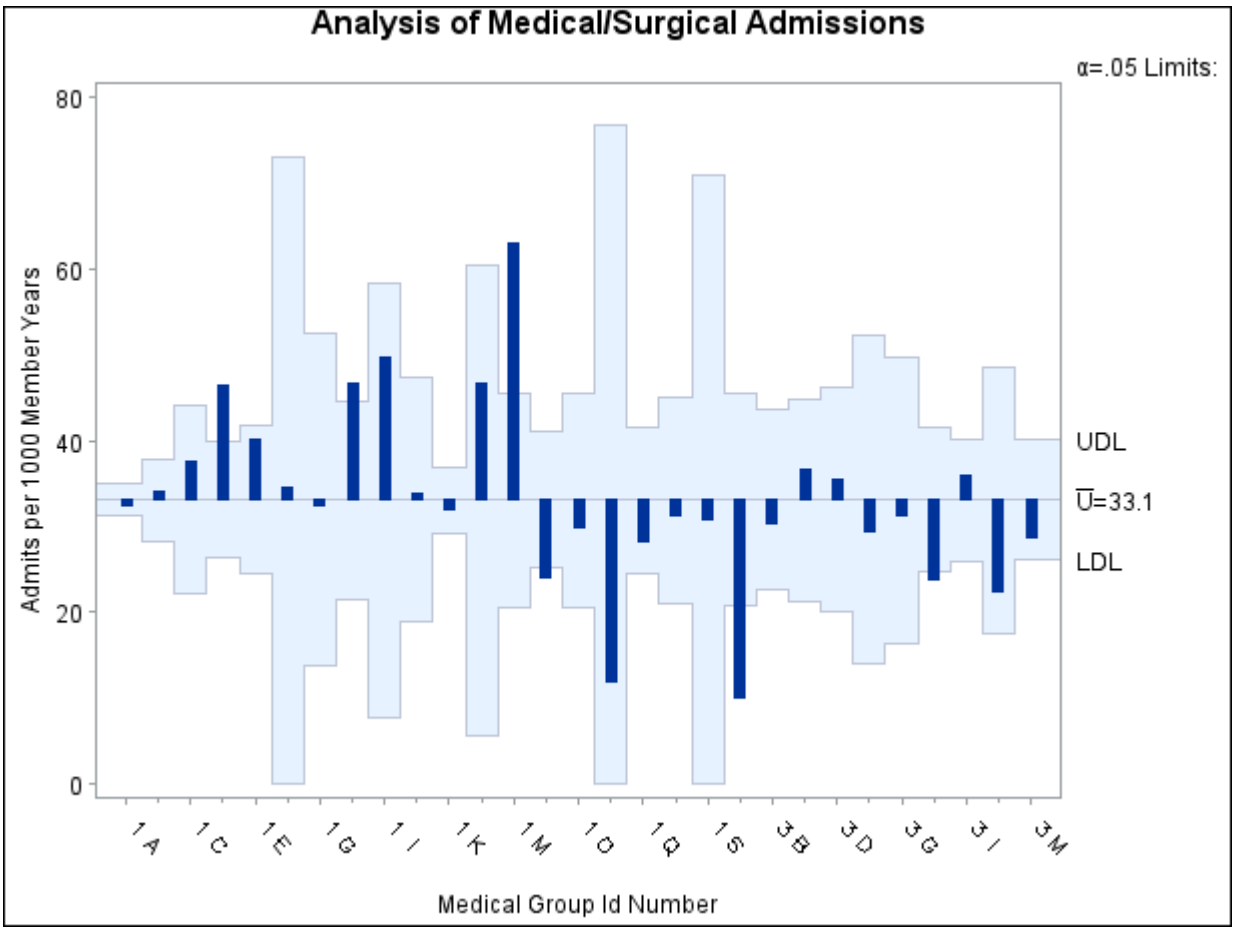
NOTE: See *Creating ANOM Charts with Angled Axis Labels* in the SAS/QC Sample Library.

Consider the example described in "Creating ANOM Charts for Rates from Group Counts" on page 106. In the example, the option TURNHLABELS was used to vertically display the horizontal axis labels. You can also use an AXIS statement to create an angled display of the horizontal or vertical axes labels. The following statements create the u CHART shown in Output 5.3.1:

```
ods graphics off;
title 'Analysis of Medical/Surgical Admissions';
proc anom data=MSAdmits;
   uchart Count*ID / groupn   = KMemberYrs
                     nolegend
                     haxis    = axis1;
axis1 value = (a=-45 h=2.0pct);
label Count = 'Admits per 1000 Member Years';
run;
```

The angle is specified with the A= option in the AXIS1 statement. Valid angle values are between `-90` and `90`. The height of the labels is specified with the H= option in the AXIS1 statement. See Axis and Axis Label Options in Table 5.20 for a list of UCHART statement axis options and *SAS/GRAPH: Reference* for a complete description of the AXIS statement.

Output 5.3.1 ANOM u Chart for C-Sections with Angled Axis Labels



XCHART Statement: ANOM Procedure

Overview: XCHART Statement

The XCHART statement creates an ANOM chart for group (treatment level) means of response values. You can use options in the XCHART statement to

- compute decision limits from the data based on specified parameters, such as the significance level (α)
- tabulate group sample sizes, group means, decision limits, and other information
- save decision limits in an output data set
- save group sample sizes and group means in an output data set
- read decision limits and decision limit parameters from a data set
- display distinct sets of decision limits for different sets of groups
- add block legends and symbol markers to identify special groups
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have two alternatives for producing ANOM charts with the XCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced if SAS/GRAPH® is licensed.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: XCHART Statement

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the section "Syntax: XCHART Statement" on page 137, and advanced examples are given in the section "Examples: XCHART Statement" on page 157.

Creating ANOM Charts for Means from Response Values

NOTE: See *Creating ANOM Charts for Means from Response Variables* in the SAS/QC Sample Library.

A manufacturing engineer carries out a study to determine the source of excessive variation in the positioning of labels on shampoo bottles. [2] A labeling machine removes bottles from the line, attaches the labels, and returns the bottles to the line. There are six positions on the machine, and the engineer suspects that one or more of the position heads might be faulty.

A sample of 60 bottles, 10 per position, is run through the machine. For each bottle, the deviation of the label is measured in millimeters, and the machine position is recorded. The following statements create a SAS data set named LabelDeviations, which contains the deviation measurements for the 60 bottles:

```
data LabelDeviations;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.02386  -0.02853  -0.03001  -0.00428  -0.03623
1  -0.04222  -0.00144  -0.06466   0.00944  -0.00163
2  -0.02014  -0.02725   0.02268  -0.03323   0.03661
2   0.04378   0.05562   0.00977   0.05641   0.01816
3  -0.00728   0.02849  -0.04404  -0.02214  -0.01394
3   0.04855   0.03566   0.02345   0.01339  -0.00203
4   0.06694   0.10729   0.05974   0.06089   0.07551
4   0.03620   0.05614   0.08985   0.04175   0.05298
5   0.03677   0.00361   0.03736   0.01164  -0.00741
5   0.02495  -0.00803   0.03021  -0.00149  -0.04640
6   0.00493  -0.03839  -0.02037  -0.00487  -0.01202
6   0.00710  -0.03075   0.00167  -0.02845  -0.00697
;
```

A partial listing of LabelDeviations is shown in Figure 5.21.

[2] This example is based on a case study described by Hansen (1990).

Figure 5.21 Partial Listing of the Data Set LabelDeviations

The Data Set LabelDeviations

Position	Deviation
1	-0.02386
1	-0.02853
1	-0.03001
1	-0.00428
1	-0.03623
1	-0.04222
1	-0.00144
1	-0.06466
1	0.00944
1	-0.00163
2	-0.02014
2	-0.02725
2	0.02268
2	-0.03323
2	0.03661
2	0.04378
2	0.05562
2	0.00977
2	0.05641
2	0.01816

The data set LabelDeviations is said to be in "strung-out" form, since each observation contains the position and the deviation measurement for a single bottle. The first 10 observations contain the measurements for the first position, the second 10 observations contain the measurements for the second position, and so on. Because the variable Position classifies the observations into groups (treatment levels), it is referred to as the *group-variable*. The input data set must be sorted by the group variable. The variable Deviation contains the deviation measurements and is referred to as the *response variable* (or *response* for short).

The following statements create an ANOM chart for Position:

```
ods graphics on;
title 'Analysis of Label Deviations';
proc anom data=LabelDeviations;
   xchart Deviation*Position / alpha    = 0.05
                               odstitle = title;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

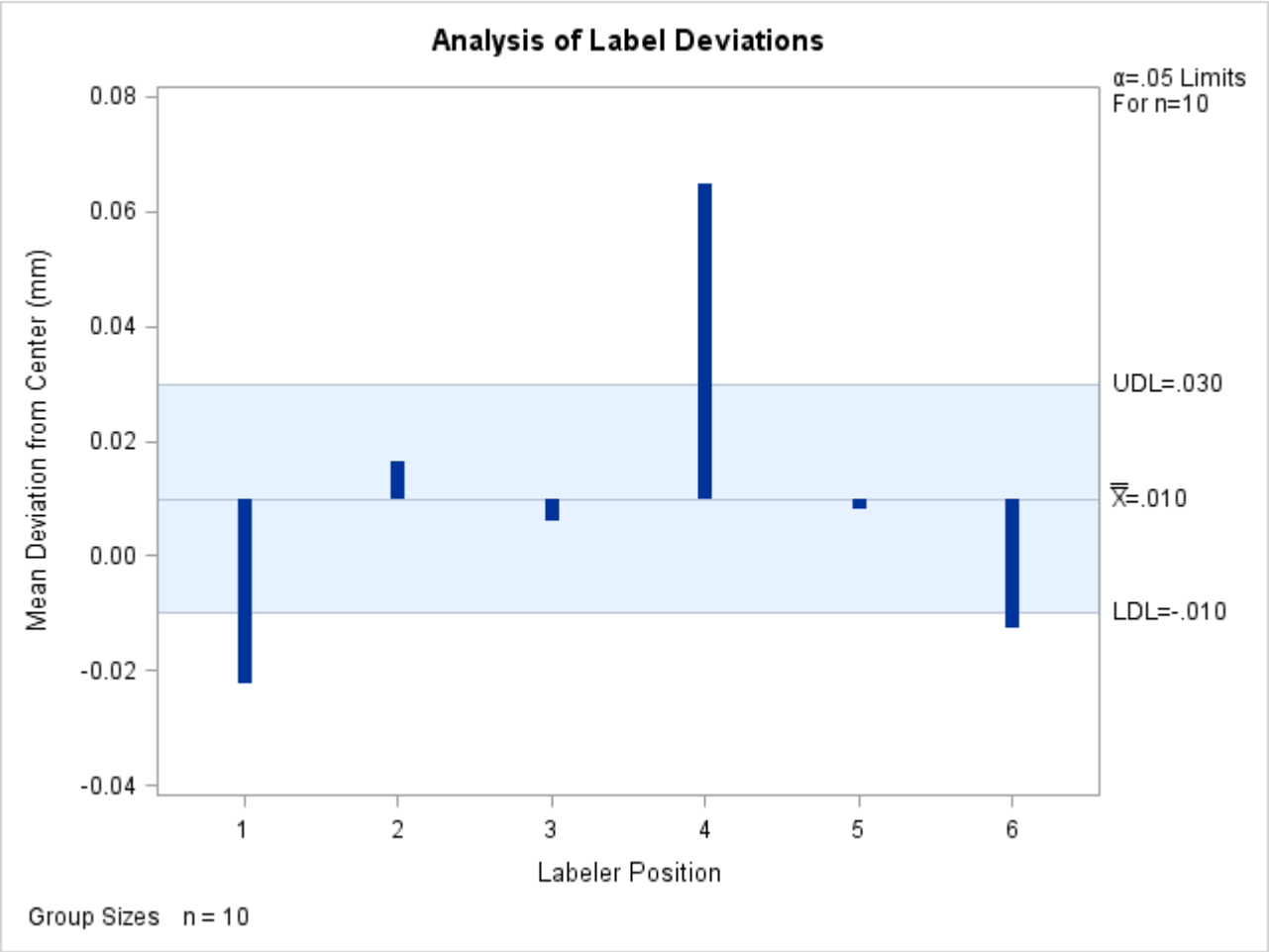
The ODS GRAPHICS ON statement specified before the PROC ANOM statement enables ODS Graphics, so the ANOM chart is created using ODS Graphics instead of traditional graphics. The resulting chart is shown in Figure 5.22.

This example illustrates the basic form of the XCHART statement. After the keyword XCHART, you specify the *response* to analyze (in this case, Deviation) followed by an asterisk and the *group-variable* (Position). Options are specified after the slash (/) in the XCHART statement. A complete list of options is presented in the section "Syntax: XCHART Statement" on page 137.

The input data set is specified with the DATA= option in the PROC ANOM statement when it contains raw measurements for the *response*.

Each point on the ANOM chart represents the average (mean) of the response measurements for a particular sample.

Figure 5.22 ANOM Chart for Means of Labeler Position Data



The average for Position 1 is below the lower decision limit (LDL), and the average for Position 6 is slightly below the lower decision limit. The average for Position 4 exceeds the upper decision limit (UDL). The conclusion is that Positions 1, 4, and 6 are operating differently.

By default, the decision limits shown correspond to a significance level of $\alpha = 0.05$; the formulas for the limits are given in the section "Decision Limits" on page 147. You can also read decision limits from an input data set.

For computational details, see "Constructing ANOM Charts for Means" on page 146. For details on reading raw measurements, see "DATA= Data Set" on page 153.

Creating ANOM Charts for Means from Group Summary Data

NOTE: See *Creating ANOM Charts for Means from Group Summary Data* in the SAS/QC Sample Library.

The previous example illustrates how you can create ANOM charts for means using measurement data. However, in many applications, the data are provided as group summary statistics. This example illustrates how you can use the XCHART statement with data of this type.

The following data set (Labels) provides the data from the preceding example in summarized form:

```
data Labels;
   input Position DeviationX DeviationS;
   DeviationN = 10;
   datalines;
1  -0.02234  0.02281
2   0.01624  0.03348
3   0.00601  0.02885
4   0.06473  0.02149
5   0.00812  0.02592
6  -0.01281  0.01597
;
```

A listing of Labels is shown in Figure 5.23. There is exactly one observation for each group (note that the groups are still indexed by Position). The variable DeviationX contains the group means, the variable DeviationS contains the group standard deviations, and the variable DeviationN contains the group sample sizes (these are all 10).

Figure 5.23 The Summary Data Set Labels

The Data Set Labels

Position	DeviationX	DeviationS	DeviationN
1	-0.02234	0.02281	10
2	0.01624	0.03348	10
3	0.00601	0.02885	10
4	0.06473	0.02149	10
5	0.00812	0.02592	10
6	-0.01281	0.01597	10

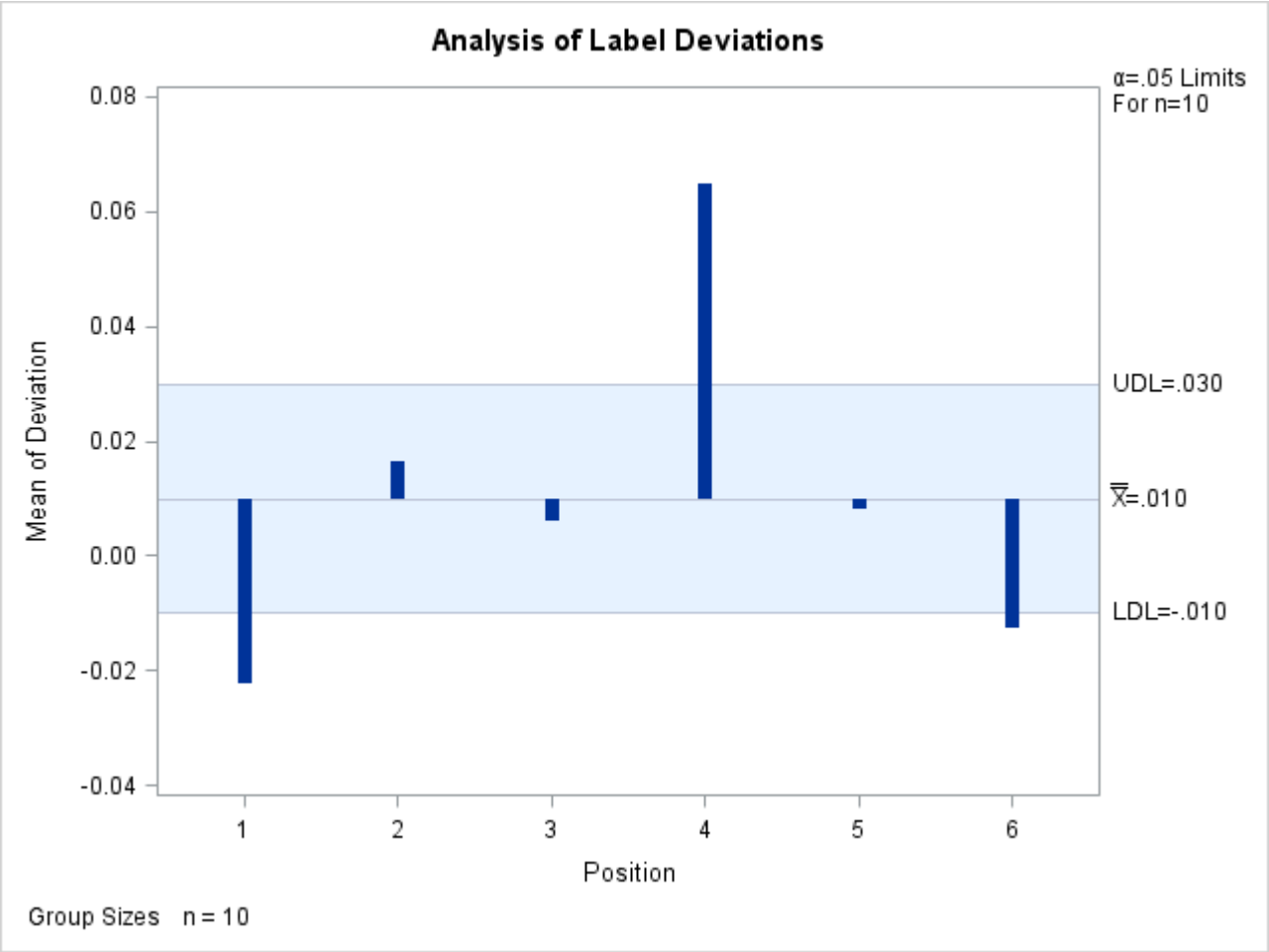
You can read this data set by specifying it as a SUMMARY= data set in the PROC ANOM statement, as follows:

```
title 'Analysis of Label Deviations';
proc anom summary=Labels;
   xchart Deviation*Position / odstitle=title1;
run;
```

The resulting chart is shown in Figure 5.24.

Note that Deviation is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables DeviationX, DeviationS, and DeviationN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three group summary variables in a SUMMARY= data set with a single name (Deviation), which is referred to as the *response*. The name Position specified after the asterisk is the name of the *group-variable*.

Figure 5.24 ANOM Chart for Means in Data Set Labels



In general, a SUMMARY= input data set used with the XCHART statement must contain the following variables:

- group variable
- group mean variable
- group standard deviation variable
- group sample size variable

Furthermore, the names of the group mean, standard deviation, and sample size variables must begin with the *response* name specified in the XCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC ANOM statement to rename the variables for the duration of the ANOM procedure step. If a label is associated with the group mean variable, it is used to label the vertical axis.

In summary, the interpretation of *response* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *response* is the name of the SAS variable containing the response measurements.
- If summary data are read using the SUMMARY= option (as in this example), *response* is the common prefix for the names of the variables containing the summary statistics.

For more information, see the section "SUMMARY= Data Set" on page 155.

Saving Summary Statistics for Groups

NOTE: See *Saving Summary Statistics for Groups Using ANOM Charts* in the SAS/QC Sample Library.

In this example, the XCHART statement is used to create a data set containing group summary statistics that can be read later by the ANOM procedure (as in the preceding example). The following statements read measurements from the data set LabelDeviations and create a summary data set named LabelSummary:

```
proc anom data=LabelDeviations;
   xchart Deviation*Position / outsummary=LabelSummary
                               nochart;
run;
```

The OUTSUMMARY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 5.22.

Figure 5.25 contains a listing of LabelSummary.

Figure 5.25 The Summary Data Set LabelSummary

The Data Set LabelSummary

Position	DeviationX	DeviationS	DeviationN
1	-0.022342	0.022805	10
2	0.016241	0.033478	10
3	0.006011	0.028847	10
4	0.064729	0.021492	10
5	0.008121	0.025920	10
6	-0.012812	0.015974	10

There are four variables in the data set LabelSummary.

- Position identifies the group.
- DeviationX contains the group means.
- DeviationS contains the group standard deviations.
- DeviationN contains the group sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *response* Deviation specified in the XCHART statement. In other words, the variable naming convention for OUTSUMMARY= data sets is the same as that for SUMMARY= data sets.

For more information, see the section "OUTSUMMARY= Data Set" on page 151.

Saving Decision Limits

NOTE: See *Saving Decision Limits Using ANOM Charts for Means* in the SAS/QC Sample Library.

You can save the decision limits for an ANOM chart, together with the parameters used to compute the limits, in a SAS data set.

The following statements read measurements from the data set LabelDeviations (see the section "Creating ANOM Charts for Means from Response Values" on page 130) and save the decision limits displayed in Figure 5.22 in a data set named LabelLimits:

```
proc anom data=LabelDeviations;
   xchart Deviation*Position / outlimits=LabelLimits
                               nochart;
run;
```

The OUTLIMITS= option names the data set containing the decision limits, and the NOCHART option suppresses the display of the chart. The data set LabelLimits is listed in Figure 5.26.

Figure 5.26 The Data Set LabelLimits Containing Decision Limit Information

Decision Limits for Labler Position Deviations

VAR	_GROUP_	_TYPE_	_LIMITN_	_ALPHA_	_LDLX_	_MEAN_	_UDLX_	_MSE_	_DFE_	_LIMITK_
Deviation	Position	ESTIMATE	10	0.05	-.009878975	.009991333	0.029862	.000643646	54	6

The data set LabelLimits contains one observation with the limits for *response* Deviation. The values of _LDLX_ and _UDLX_ are the lower and upper decision limits for the means, and the value of _MEAN_ is the weighted average of the group means, which is represented by the central line.

The values of _MEAN_, _MSE_, _DFE_, _LIMITN_, _LIMITK_, and _ALPHA_ are the parameters used to compute the decision limits as described in the section "Constructing ANOM Charts for Means" on page 146. The value of _MSE_ is the mean square error, and the value of _DFE_ is the associated degrees of freedom. The value of _LIMITN_ is the nominal sample size (n) associated with the decision limits, the value of _LIMITK_ is the number of groups (k), and the value of _ALPHA_ is the value of the significance level (α). The variables _VAR_ and _GROUP_ are bookkeeping variables that save the *response* and *group-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _MSE_ are estimates computed from the data or standard (known) values specified with procedure options. In most applications, the value of _TYPE_ will be 'ESTIMATE.'

NOTE: See *Saving Summary Statistics & Decision Limits Using ANOM Charts* in the SAS/QC Sample Library.

You can create an output data set containing both decision limits and group summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc anom data=LabelDeviations;
   xchart Deviation*Position / outtable=LabelTab
                               nochart;
run;
```

The data set LabelTab is listed in Figure 5.27.

Figure 5.27 The Data Set LabelTab

Summary Statistics and Decision Limits

VAR	Position	_ALPHA_	_LIMITN_	_SUBN_	_LDLX_	_SUBX_	_MEAN_	_UDLX_	_EXLIM_
Deviation	1	0.05	10	10	-.009878975	-0.022342	.009991333	0.029862	LOWER
Deviation	2	0.05	10	10	-.009878975	0.016241	.009991333	0.029862	
Deviation	3	0.05	10	10	-.009878975	0.006011	.009991333	0.029862	
Deviation	4	0.05	10	10	-.009878975	0.064729	.009991333	0.029862	UPPER
Deviation	5	0.05	10	10	-.009878975	0.008121	.009991333	0.029862	
Deviation	6	0.05	10	10	-.009878975	-0.012812	.009991333	0.029862	LOWER

This data set contains one observation for each group sample. The variables _SUBX_ and _SUBN_ contain the group means and sample sizes. The variables _LDLX_ and _UDLX_ contain the lower and upper decision limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Position contain the *response* name and values of the *group-variable*, respectively. For more information, see the section "OUTTABLE= Data Set" on page 152.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read LabelTab and display an ANOM chart (not shown here) identical to the chart in Figure 5.22:

```
title 'Analysis of Label Deviations';
proc anom table=LabelTab;
   xchart Deviation*Position;
label _SUBX_ = 'Mean Deviation from Center (mm)';
run;
```

Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

For more information, see the section "TABLE= Data Set" on page 156.

Syntax: XCHART Statement

The basic syntax for the XCHART statement is as follows:

XCHART *response* ∗ *group-variable* **;**

The general form of this syntax is as follows:

XCHART *responses* ∗ *group-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > < *options* > **;**

You can use any number of XCHART statements in the ANOM procedure. The components of the XCHART statement are described as follows.

response

responses

identify one or more responses to be analyzed. The specification of *response* depends on the input data set specified in the PROC ANOM statement.

- If response values (raw data) are read from a DATA= data set, *response* must be the name of the variable containing the values. For an example, see the section "Creating ANOM Charts for Means from Response Values" on page 130.
- If summary data are read from a SUMMARY= data set, *response* must be the common prefix of the summary variables in the SUMMARY= data set. For an example, see the section "Creating ANOM Charts for Means from Group Summary Data" on page 133.
- If summary data and decision limits are read from a TABLE= data set, *response* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see the section "Saving Decision Limits" on page 136.

A *response* is required. If you specify more than one response, enclose the list in parentheses. For example, the following statements request distinct ANOM charts for the means of Weight, Length, and Width:

```
proc anom data=Measures;
   xchart (Weight Length Width)*Day;
run;
```

group-variable

is the variable that identifies groups in the data. The *group-variable* is required. In the preceding XCHART statement, Day is the group variable.

block-variables

are optional variables that group the data into blocks of consecutive groups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker used to plot the means. Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements.

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section The section "Summary of Options" lists all options by function.

Summary of Options

The following tables list the XCHART statement options by function. Options unique to the ANOM procedure are listed in Table 5.27, and are described in detail in "Dictionary of ANOM Chart Statement Options" on page 183. Options that are common to both the ANOM and SHEWHART procedures are listed in Table 5.28, and are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 5.27 XCHART Statement Special Options

Option	Description
Options for Specifying Parameters for Decision Limits	
ALPHA=	Specifies the probability of a Type I error
DFE=	Specifies the degrees of freedom associated with the root mean square error
LIMITK=	Specifies number of groups for decision limits
LIMITN=	Specifies either nominal sample size for fixed decision limits or varying limits
MEAN=	Specifies the mean
MSE=	Specifies the mean square error
NOREADLIMITS	Computes decision limits for each *response* from the data rather than a LIMITS= data set
READINDEXES=	Reads multiple sets of decision limits for each *response* from a LIMITS= data set
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Displaying Decision Limits	
CINFILL=	Specifies color for area inside decision limits
CLIMITS=	Specifies color of decision limits, central line, and related labels
LDLLABEL=	Specifies label for lower decision limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the decision limit
LLIMITS=	Specifies line type for decision limits
NDECIMAL=	Specifies number of digits to right of decimal place in default labels for decision limits and central line
NOCTL	Suppresses display of central line
NOLDL	Suppresses display of lower decision limit
NOLIMITLABEL	Suppresses labels for decision limits and central line
NOLIMITS	Suppresses display of decision limits
NOLIMITSFRAME	Suppresses default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for decision limits
NOUDL	Suppresses display of upper decision limit
UDLLABEL=	Specifies label for upper decision limit
WLIMITS=	Specifies width for decision limits and central line
XSYMBOL=	Specifies label for central line
Output Data Set Option	
OUTSUMMARY=	Creates output data set containing group summary statistics

Table 5.28 XCHART Statement General Options

Option	Description
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on ANOM chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside decision limits
COUTFILL=	Specifies color for shading areas between the connected points and decision limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels
LABELHEIGHT=	Specifies height of labels
NONEEDLES	Suppresses vertical needles connecting points to central line
OUTLABEL=	Labels points outside decision limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent group values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis

Table 5.28 *continued*

Option	Description
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of ANOM chart
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for ANOM chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all groups
BILEVEL	Creates ANOM charts using half-screens and half-pages
EXCHART	Creates ANOM chart for a response only when a group mean exceeds the decision limits
INTERVAL=	Specifies the natural time interval between consecutive group positions when time, date, or datetime format is associated with a numeric group variable
MAXPANELS=	Specifies the maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed decision limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for group sample sizes
NPANELPOS=	Specifies number of group positions per panel on each chart
REPEAT	Repeats last group position on panel as first group position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays ANOM chart regardless of whether root mean square error is zero
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= option
CVREF=	Specifies color for lines requested by VREF= option
HREF=	Specifies position of reference lines perpendicular to horizontal axis on ANOM chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on ANOM chart

Table 5.28 *continued*

Option	Description
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on ANOM chart
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	Specifies position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to ANOM chart
DESCRIPTION=	Specifies description of ANOM chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on chart
NAME=	Specifies name of ANOM chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option

Table 5.28 *continued*

Option	Description
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	Draws portions of line segments that connect points outside decision limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between decision limits and connected points lying outside the limits
STARFILL=	Specifies a variable identifying groups of stars filled with different colors
STARS=	Specifies a variable identifying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the decision limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots group points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
OUTFILLTRANSPARENCY=	Specifies decision limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases

Table 5.28 *continued*

Option	Description
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with groups
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing decision limits
OUTTABLE=	Creates output data set containing group summary statistics and decision limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of group means, group sample sizes, and decision limits
TABLEALL	Creates all the tables that are produced by the TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS options
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLEOUTLIM	Augments basic table with columns indicating decision limits exceeded
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend

Table 5.28 *continued*

Option	Description
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels decision limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	Superimposes star at each point on ANOM chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points

Table 5.28 *continued*

Option	Description
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive ANOM Charts	
HTML=	Specifies a variable whose values create links to be associated with groups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Details: XCHART Statement

Constructing ANOM Charts for Means

The following notation is used in this section:

X_{ij}	jth response in the ith group
k	Number of groups
n_i	Sample size of ith group
N	Total sample size $= n_1 + \cdots + n_k$
μ_i	Expected value of a response in the ith group
σ	Standard deviation of a response
$\bar{X}_i$	Average response in the ith group
$\overline{\overline{X}}$	Weighted average of k group means
s_i^2	Sample variance of the responses in the ith group
$\widehat{\sigma^2}$	Mean square error (MSE)
ν	Degrees of freedom associated with the mean square error
α	Significance level
$h(\alpha; k, n, \nu)$	Critical value for analysis of means when the sample sizes n_i are equal ($n_i \equiv n$)
$h(\alpha; k, n_1, \ldots, n_k, \nu)$	Critical value for analysis of means when the sample sizes n_i are not equal

Plotted Points

Each point on an ANOM chart indicates the value of a group mean ($\bar{X}_i$).

Central Line

By default, the central line on an ANOM chart for means represents the weighted average of the group means, which is computed as

$$\overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_k\bar{X}_k}{n_1 + \cdots + n_k}$$

You can specify a value for $\overline{\overline{X}}$ with the MEAN= option in the XCHART statement or with the variable _MEAN_ in a LIMITS= data set.

Decision Limits

In the analysis of means for continuous data, it is assumed that the responses in the ith group are at least approximately normally distributed with a constant variance:

$$X_{ij} \sim N(\mu_i, \sigma^2), \qquad j = 1, \ldots, n_i$$

When the group sizes are constant ($n_i \equiv n$), then $\nu = N - k = k(n-1)$ and the decision limits are computed as follows:

$$\begin{aligned}
\text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}} \\
\text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{k-1}{N}}
\end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{1}{k}\sum_{j=1}^{k} s_j^2$$

For details concerning the function $h(\alpha; k, n, \nu)$, see Nelson (1982a, 1993).

When the group sizes are not constant (the unbalanced case), $\nu = N - k$ and the decision limits for the ith group are computed as follows:

$$\begin{aligned}
\text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}} \\
\text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; k, n_1, \ldots, n_k, \nu)\sqrt{\text{MSE}}\sqrt{\frac{N-n_i}{Nn_i}}
\end{aligned}$$

Here the mean square error (MSE) is computed as follows:

$$\text{MSE} = \widehat{\sigma^2} = \frac{(n_1-1)s_1^2 + \cdots + (n_k-1)s_k^2}{n_1 + \cdots + n_k - k}$$

This requires that ν be positive. A chart is not produced if $\nu > 0$ but MSE is equal to zero (unless you specify the ZEROSTD option). For details concerning the function $h(\alpha; k, n_1, \ldots, n_k, \nu)$, see Nelson (1991).

You can specify parameters for the limits as follows:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, $\alpha = 0.05$.
- Specify a constant nominal sample size $n_i \equiv n$ for the decision limits in the balanced case with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. By default, n is the observed sample size in the balanced case.
- Specify k with the LIMITK= option or with the variable _LIMITK_ in a LIMITS= data set. By default, k is the number of groups.
- Specify $\overline{\overline{X}}$ with the MEAN= option or with the variable _MEAN_ in a LIMITS= data set. By default, $\overline{\overline{X}}$ is the weighted average of the responses.
- Specify $\widehat{\sigma^2}$ with the MSE= option or with the variable _MSE_ in a LIMITS= data set. By default, $\widehat{\sigma^2}$ is computed as indicated above.
- Specify ν with the DFE= option or with the variable _DFE_ in a LIMITS= data set. By default, ν is determined as indicated above.

Constructing ANOM Charts for Two-Way Layouts

This section provides the computational details for constructing an ANOM chart for the lth factor in an experiment involving two factors ($l = 1$ or 2). It is assumed that there is no interaction effect. See Example 5.5 for an illustration.

The following notation is used in this section:

X_{ijk}	kth response at the ith level of factor 1 and the jth level of factor 2, where $k = 1, 2, \ldots, n_{ij}$
f_l	Number of groups (levels) for the lth factor, $l = 1, 2$
n_{ij}	Number of replicates in cell (i, j)
N	Total sample size $= \sum_{i=1}^{f_1} \sum_{j=1}^{f_2} n_{ij}$
σ^2	Variance of a response
$\overline{X}_{ij\cdot}$	Average response in cell (i, j)
$\overline{X}_{i\cdot\cdot}$	Average response for ith level of factor 1 $= \left(\sum_{j=1}^{f_2} n_{ij} \overline{X}_{ij\cdot}\right) / \left(\sum_{j=1}^{f_2} n_{ij}\right)$
$\overline{X}_{\cdot j\cdot}$	Average response for jth level of factor 2 $= \left(\sum_{i=1}^{f_1} n_{ij} \overline{X}_{ij\cdot}\right) / \left(\sum_{i=1}^{f_1} n_{ij}\right)$
$\overline{\overline{X}}$	$\sum_{i=1}^{f_1} \sum_{j=1}^{f_2} n_{ij} \overline{X}_{ij} / N$
s_{ij}^2	Sample variance of the responses for the ith level of factor 1 and the jth level of factor 2
$\widehat{\sigma^2}$	Mean square error (MSE) in the two-way analysis of variance
ν	Degrees of freedom associated with the mean square error in the two-way analysis of variance
α	Significance level

$h(\alpha; f_l, n, \nu)$	Critical value for analysis of means in a one-way layout for f_l groups (treatment levels) when the sample sizes for each level are constant ($\equiv n$) and ν is the degrees of freedom associated with the mean square error; see the section "Constructing ANOM Charts for Means" on page 146.

Plotted Points

The points on the ANOM chart for factor 1 represent $\overline{X}_{i\cdot\cdot}$, $i = 1, \ldots, f_1$ and the points on the ANOM chart for factor 2 represent $\overline{X}_{\cdot j\cdot}$, $j = 1, \ldots, f_2$.

Central Line

The central line on the ANOM chart for the lth factor is the overall weighted average $\overline{\overline{X}}$. Some authors use the notation $\overline{X}_{\cdots}$ for this average.

Decision Limits

It is assumed that

$$X_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$$

where the quantities ϵ_{ijk} are independent and at least approximately normally distributed with

$$\epsilon_{ijk} \sim N(0, \sigma^2)$$

The correct decision limits for a given factor in a two-way layout are not computed by default when the lth factor is specified as the *group-variable* in the XCHART statement, since the mean square error and degrees of freedom are not adjusted for the two-way structure of the data. Consequently, $\widehat{\sigma^2}$ and ν must be precomputed and provided to the ANOM procedure, as illustrated in Example 5.5.

In the case of a two-way layout with equal group sizes ($n_{ij} \equiv n$), the appropriate decision limits are:

$$\begin{aligned} \text{lower decision limit (LDL)} &= \overline{\overline{X}} - h(\alpha; f_l, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{f_l - 1}{N}} \\ \text{upper decision limit (UDL)} &= \overline{\overline{X}} + h(\alpha; f_l, n, \nu)\sqrt{\text{MSE}}\sqrt{\frac{f_l - 1}{N}} \end{aligned}$$

where the mean square error (MSE) is computed as in the ANOVA or GLM procedure:

$$\text{MSE} = \widehat{\sigma^2} = \frac{1}{f_1 f_2}\sum_{i=1}^{f_1}\sum_{j=1}^{f_2} s_{ij}^2$$

and the degrees of freedom for error is $\nu = f_1 f_2(n-1)$. For details concerning the function $h(\alpha; f_l, n, \nu)$, see Nelson (1982a, 1993).

You can provide the appropriate values of MSE and ν by

- specifying $\widehat{\sigma^2}$ with the MSE= option or with the variable _MSE_ in a LIMITS= data set
- specifying ν with the DFE= option or with the variable _DFE_ in a LIMITS= data set

In addition you can:

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set. By default, $\alpha = 0.05$.
- Specify a constant nominal sample size $n_{ij} \equiv n$ for the decision limits in the balanced case with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify f_l with the LIMITK= option or with the variable _LIMITK_ in a LIMITS= data set.
- Specify $\overline{\overline{X}}$ with the MEAN= option or with the variable _MEAN_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves decision limits and decision limit parameters. Table 5.31 lists the variables can be saved.

Table 5.31 OUTLIMITS= Data Set

Variable	Description
ALPHA	Significance level (α).
DFE	Degrees of freedom for mean square error (ν)
GROUP	*Group-variable* specified in the XCHART statement
INDEX	Optional identifier for the decision limits specified with the OUTINDEX= option
LDLX	Lower decision limit for group means
LIMITK	Number of groups
LIMITN	Group sample size associated with the decision limits
MEAN	Weighted average of group means ($\overline{\overline{X}}$)
MSE	Mean square error ($\widehat{\sigma^2}$).
TYPE	Type (estimate or standard value) of _MEAN_ and _MSE_
UDLX	Upper decision limit for group means
VAR	*Response* specified in the XCHART statement

Notes:

1. In the unbalanced case, the special missing value *V* is assigned to the variables _LIMITN_, _LDLX_, and _UDLX_ to indicate that the decision limits vary with the group sample size.
2. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *response* specified in the XCHART statement. For an example, see the section "Saving Decision Limits" on page 136.

OUTSUMMARY= Data Set

The OUTSUMMARY= data set saves group summary statistics. The following variables can be saved:

- the *group-variable*
- a group mean variable named by *response* suffixed with *X*
- a group sample size variable named by *response* suffixed with *N*
- a group standard deviation variable named by *response* suffixed with *S*

Given a *response* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Group summary variables are created for each *response* specified in the XCHART statement. For example, consider the following statements:

```
proc anom data=Steel;
   xchart (Width Diameter)*Lot / outsummary=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthS, WidthN, DiameterX, DiameterS, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTSUMMARY= data set, see the section "Saving Summary Statistics for Groups" on page 135.

OUTTABLE= Data Set

The OUTTABLE= data set saves group summary statistics, decision limits, and related information. Table 5.32 lists the variables that can be saved.

Table 5.32 OUTTABLE= Data Set

Variable	Description
ALPHA	Significance level (α)
EXLIM	Decision limit exceeded (if any)
Group	Values of the group variable
LDLX	Lower decision limit for group mean
LIMITN	Nominal sample size associated with the decision limits
MEAN	Central line
SUBN	Group sample size
SUBX	Group mean
UDLX	Upper decision limit for group mean
VAR	*Response* specified in the XCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

NOTE: The variable _EXLIM_ is a character variable of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see the section "Saving Decision Limits" on page 136.

ODS Tables

The following table summarizes the ODS tables that you can request with the XCHART statement.

Table 5.33 ODS Tables Produced with the XCHART Statement

Table Name	Description	Options
XChartSummary	ANOM chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT,

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. XCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the XCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 5.34.

Table 5.34 ODS Graphics Produced by the XCHART Statement

ODS Graph Name	Plot Description
XChart	ANOM chart for means

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (response values) from a DATA= data set specified in the PROC ANOM statement. Each *response* specified in the XCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into group samples indexed by the *group-variable*. The *group-variable*, which is specified in the XCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *response* and a value for the *group-variable*. If the ith group contains n_i items, there should be n_i consecutive observations for which the value of the *group-variable* is the index of the ith group. For example, if each group contains five items and there are 10 groups, the DATA= data set should contain 50 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option.

For an example of a DATA= data set, see the section "Creating ANOM Charts for Means from Response Values" on page 130.

LIMITS= Data Set

You can read preestablished decision limits (or parameters from which the decision limits can be calculated) from a LIMITS= data set specified in the PROC ANOM statement. For example, the following statements read decision limit information from the data set Conlims:

```
proc anom data=Info limits=Conlims;
   xchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the ANOM procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 5.31. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following minimal combinations of variables:

- the variables _LDLX_, _MEAN_, and _UDLX_, which specify the decision limits directly
- the variables _MEAN_ and _MSE_, with _DFE_ recommended, which are used to calculate the decision limits according to the equations in the section "Decision Limits" on page 147

In addition, note the following:

- The variables _VAR_ and _GROUP_ are always required. These must be character variables whose lengths are no greater than 32.
- _DFE_ is optional. The default is $\nu = N - k$, and in the case of equal group sizes, $\nu = k(n-1)$.
- _MSE_ is optional if _LDLX_ and _UDLX_ are specified; otherwise it is required.
- _LDLX_ and _UDLX_ must be specified together; otherwise their values are computed.
- _ALPHA_ is optional but is recommended in order to maintain a complete set of decision limit information. The default value is 0.05.
- _LIMITK_ is optional. The default value is k, the number of groups. A group must have at least one nonmissing value ($n_i \geq 1$) and there must be at least one group with $n_i \geq 2$. If specified, _LIMITK_ overrides the value of k.
- _LIMITN_ is optional. The default value is the common group size (n), in the balanced case $n_i \equiv n$. If specified, _LIMITN_ overrides the value of n.
- The variable _TYPE_ is optional, but is recommended to maintain a complete set of decision limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are 'ESTIMATE,' 'STANDARD,' 'STDMEAN,' and 'STDRMS.' The default is 'ESTIMATE.'

- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- BY variables are required if specified with a BY statement.

SUMMARY= Data Set

You can read group summary statistics from a SUMMARY= data set specified in the PROC ANOM statement. This enables you to reuse OUTSUMMARY= data sets that have been created in previous runs of the ANOM procedure or to read output data sets created with SAS summarization procedures, such as PROC MEANS.

A SUMMARY= data set used with the XCHART statement must contain the following:

- the *group-variable*
- a group mean variable for each *response*
- a group sample size variable for each *response*
- a group standard deviation variable for each *response*

The names of the group mean, group range, and group sample size variables must be the *response* name concatenated with the suffix characters *X*, *S*, and *N*, respectively.

For example, consider the following statements:

```
proc anom summary=Summary;
   xchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightS, WeightN, YieldstrengthX, YieldstrengthS, and YieldstrengthN. Note that if you specify a *response* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *response* name, suffixed with the appropriate character.

Other variables that can be read from a SUMMARY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the ANOM procedure reads all of the observations in a SUMMARY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option.

For an example of a SUMMARY= data set, see the section "Creating ANOM Charts for Means from Group Summary Data" on page 133.

TABLE= Data Set

You can read summary statistics and decision limits from a TABLE= data set specified in the PROC ANOM statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the ANOM procedure. Because the ANOM procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized ANOM charts.

Table 5.35 lists the variables required in a TABLE= data set used with the XCHART statement:

Table 5.35 Variables Required in a TABLE= Data Set

Variable	Description
Group-variable	Values of the *group-variable*
LDLX	Lower decision limit for mean
LIMITN	Nominal sample size associated with the decision limits
MEAN	Central line
SUBN	Group sample size
SUBX	Group mean
UDLX	Upper decision limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *response* is specified or if the data set contains information for more than one *response*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see the section "Saving Decision Limits" on page 136.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	All	*Group-variable*
Vertical	DATA=	*Response*
Vertical	SUMMARY=	Group mean variable
Vertical	TABLE=	_SUBX_

Missing Values

An observation read from a DATA=, SUMMARY=, or TABLE= data set is not analyzed if the value of the group variable is missing. For a particular response variable, an observation read from a DATA= data set is not analyzed if the value of the response variable is missing. Missing values of response variables generally lead to unequal group sample sizes. For a particular response variable, an observation read from a SUMMARY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: XCHART Statement

This section provides advanced examples of the XCHART statement.

Example 5.4: ANOM Charts with Unequal Group Sizes

NOTE: See *ANOM Charts with Unequal Group Sizes* in the SAS/QC Sample Library.

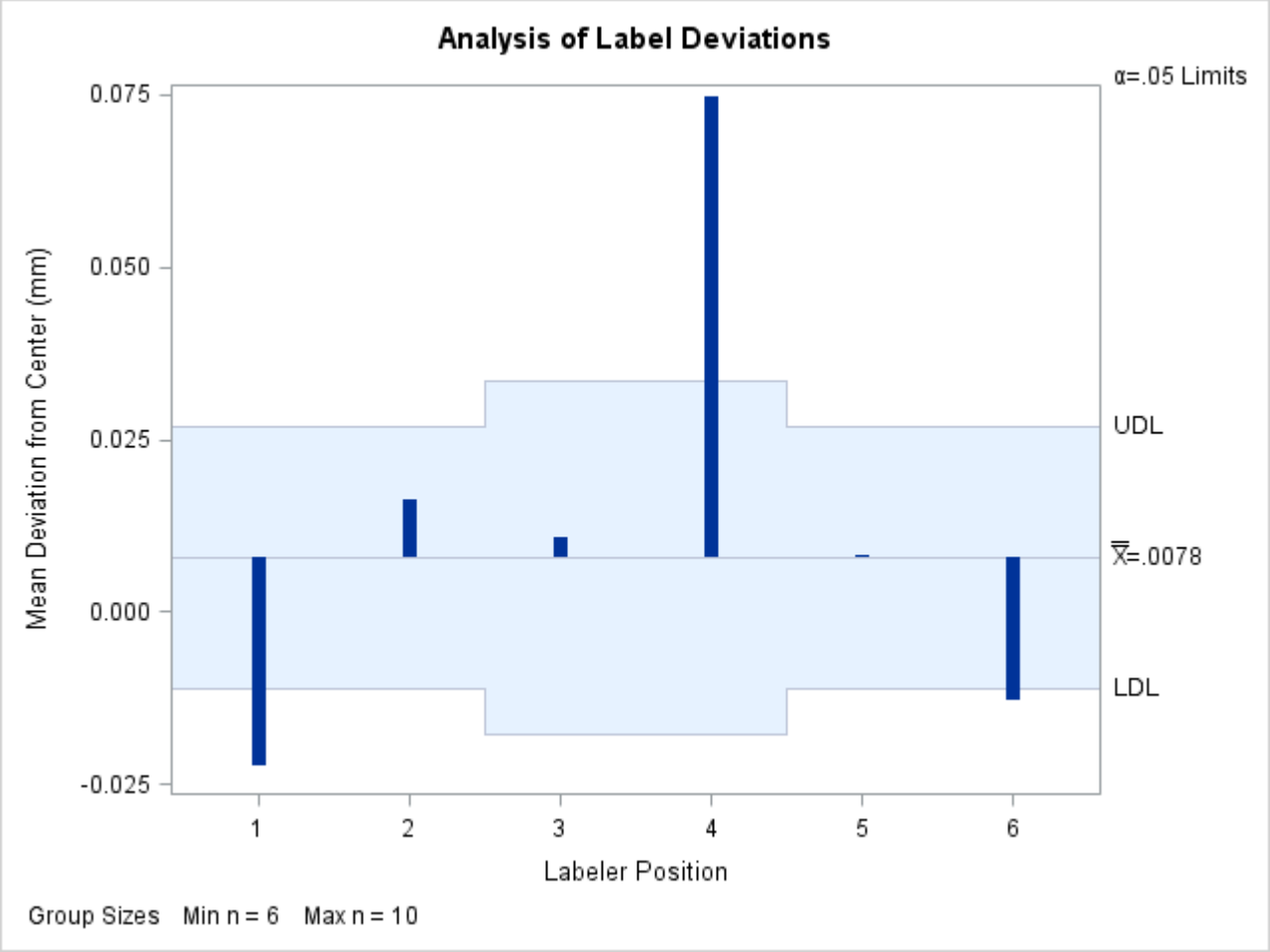
Consider the example described in "Creating ANOM Charts for Means from Response Values" on page 130. Suppose that four of the 10 measurements were missing for the third and fourth labeler positions. The following statements create a SAS data set named LabelDev2, which contains the resulting deviation measurements:

```
data LabelDev2;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.0239  -0.0285  -0.0300  -0.0043  -0.0362
1  -0.0422  -0.0014  -0.0647   0.0094  -0.0016
2  -0.0201  -0.0273   0.0227  -0.0332   0.0366
2   0.0438   0.0556   0.0098   0.0564   0.0182
3  -0.0073   0.0285     .        .      -0.0139
3     .      0.0357   0.0235     .      -0.0020
4   0.0669   0.1073     .        .       0.0755
4     .      0.0561   0.0899     .       0.0530
5   0.0368   0.0036   0.0374   0.0116  -0.0074
5   0.0250  -0.0080   0.0302  -0.0015  -0.0464
6   0.0049  -0.0384  -0.0204  -0.0049  -0.0120
6   0.0071  -0.0308   0.0017  -0.0285  -0.0070
;
```

The following statements create the ANOM chart shown in Output 5.4.1:

```
ods graphics on;
title 'Analysis of Label Deviations';
proc anom data=LabelDev2;
   xchart Deviation*Position / odstitle=title;
   label Deviation = 'Mean Deviation from Center (mm)';
   label Position  = 'Labeler Position';
run;
```

Output 5.4.1 ANOM Chart with Unequal Group Sizes



Note that the decision limits are automatically adjusted for the varying group sizes. The legend reports the minimum and maximum group sizes.

Example 5.5: ANOM for a Two-Way Classification

NOTE: See *ANOM for a Two-Way Classification* in the SAS/QC Sample Library.

A chemical engineer is interested in the effects of two factors, position and depth, on the concentration of a cleaning solution; refer to Ramig (1983) for details concerning the use of ANOM in a two-way classification such as this. The engineer is interested in the following questions:

1. Are there significant group or interaction effects due to position or depth?
2. Assuming a main effect is significant, which levels are significantly different from the overall mean and in which direction?

There are five positions and three depths. The engineer runs a two-way factorial experiment with two replications per cell. The following statements create a data set named Cleaning, which provides the concentration measurements for the $5 \times 3 \times 2 = 30$ observations.

```
data Cleaning;
   do position = 1 to 5;
      do depth = 1 to 3;
         do rep = 1 to 2;
            input concentration @@;
            output;
         end;
      end;
   end;
datalines;
15 16 15 14 19 5
15 16 14 14 0  8
19 15 16 16 11 8
18 16 19 15 8  14
15 12 19 15 8  11
;
```

In order to test for main effects and an interaction effect, the following statements use the GLM procedure:

```
ods graphics off;
proc glm data=Cleaning;
   class position depth;
   model concentration = position depth position*depth;
run;
```

The results are shown in Output 5.5.1:

Output 5.5.1 GLM Results

Analysis of Label Deviations

The GLM Procedure

Dependent Variable: concentration

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	390.4666667	27.8904762	2.21	0.0694
Error	15	189.0000000	12.6000000		
Corrected Total	29	579.4666667			

R-Square	Coeff Var	Root MSE	concentration Mean
0.673838	26.22893	3.549648	13.53333

Source	DF	Type I SS	Mean Square	F Value	Pr > F
position	4	50.4666667	12.6166667	1.00	0.4374
depth	2	281.6666667	140.8333333	11.18	0.0011
position*depth	8	58.3333333	7.2916667	0.58	0.7802

The results in Output 5.5.1 show no significant interaction effect[3] and a significant main effect due to depth. Since no interaction effect is present, you can use analysis of means to evaluate the effect of each factor as if two separate experiments had been run to determine the effect of each factor. In other words, the analysis of means is done twice, once for each factor. However, each analysis must be based on the mean square error ($\widehat{\sigma^2} = 12.6$) and the degrees of freedom for error ($\nu = 15$) from the two-way analysis of variance. These values must be specified since the ANOM procedure assumes a one-way layout by default for computing the decision limits.

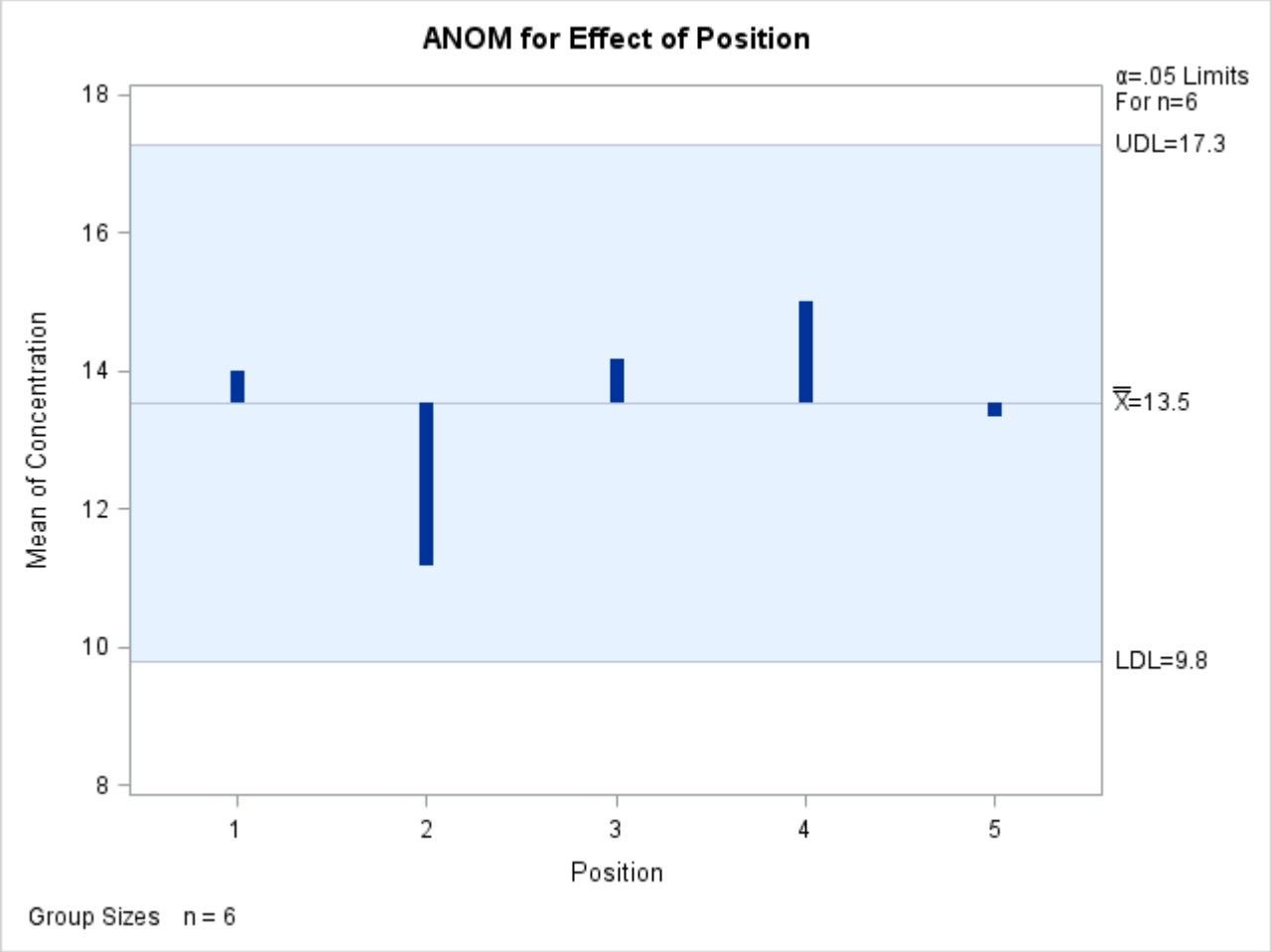
The following statements create the ANOM chart for the effect of position shown in Output 5.5.2:

```
ods graphics on;
title "ANOM for Effect of Position";
proc anom data=Cleaning;
   xchart concentration * position /
      mse      = 12.6
      dfe      = 15
      outtable = posmain
      odstitle = title;
   label position      = 'Position'
         concentration = 'Mean of Concentration';
run;
```

The MSE= and DFE= options are used to specify $\widehat{\sigma^2}$ and ν respectively. See the section "Constructing ANOM Charts for Two-Way Layouts" on page 148 for how the specified values are used to compute the decision limits. The OUTTABLE= option stores the output data set PosMain, which can be used to create a combined chart for the two factors.

[3] See Example 5.7 for an example that discusses the use of ANOM for the cell means when an interaction effect is present.

Output 5.5.2 ANOM for Effect of Position



Each point on the ANOM chart represents the average response for a particular level of position. In this case, all of the points are between the upper decision limit (UDL) and the lower decision limit (LDL). This is not surprising considering the fact that the main effect of Position was not significant in the ANOVA.

In order to create the ANOM chart for the effect of depth, the response must be sorted by depth.

```
proc sort data=Cleaning out=Cleaning2;
   by depth;
run;
```

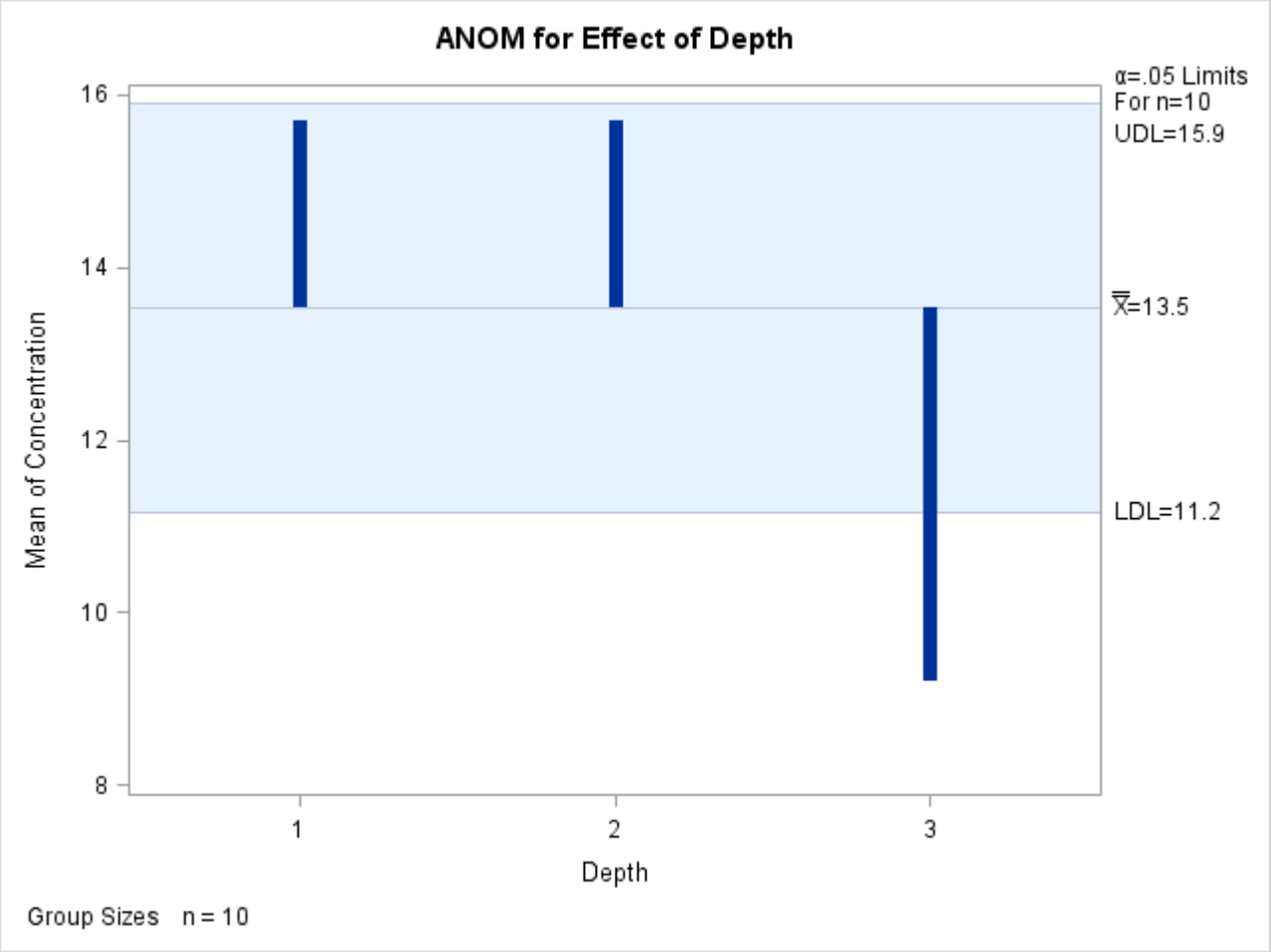
Note that for the previous chart, the measurements were sorted by Position in the original data set.

The following statements create the chart for depth:

```
title "ANOM for Effect of Depth";
proc anom data=Cleaning2;
   xchart concentration * depth /
      mse      = 12.6
      dfe      = 15
      outtable = depmain
      odstitle = title;
   label depth           = 'Depth'
         concentration   = 'Mean of Concentration';
run;
```

This produces the chart shown in Output 5.5.3: The OUTTABLE= option stores the output data set depmain, which can be used to create a combined chart for the two factors.

Output 5.5.3 ANOM for Effect of Depth



Since the average concentration for Depth 3 is less than the lower decision limit, you can conclude that the average response for Depth 3 is significantly less than the overall mean.

Example 5.6: ANOM Charts Using LIMITS= Data Set

NOTE: See *ANOM Charts Using LIMITS= Data Set* in the SAS/QC Sample Library.

In Example 5.5, statistics from a two-way ANOVA were passed to the ANOM procedure using options in order to compute the decision limits for the factor effects. This example shows how you can pass the statistics in a LIMITS= data set using the variables _MSE_ and _DFE_.

The GLM output in Output 5.5.1 provides the statistics. The following statements save the results from PROC GLM in the data sets MyFit, MyMeans, and MyOverAll:

```
ods select FitStatistics ModelANOVA OverAllANOVA;
ods output FitStatistics = MyFit
           ModelANOVA    = MyLimits
           OverAllANOVA  = MyOverAll;

proc glm data=Cleaning;
   class position depth;
   model concentration = position depth position*depth;
run;
```

The results of PROC GLM are identical to the results in Output 5.5.1.

The following statements create a LIMITS= data set to be used to create an ANOM chart for the effect of Position:

```
data ANOMParms;
   keep _var_ _group_ _alpha_ _mean_;
      length _var_ _group_ $ 14;
   set MyFit (rename=(Dependent=_var_ DepMean  =_mean_));
      _group_ = 'position';
      _alpha_ = 0.05;
run;

data ANOMParms;
   merge ANOMParms
         MyLimits (where=(source='position')
                   keep = source DF);
   _limitk_ = DF+1;
   drop source DF;
   merge MyOverAll (where=(source='Error')
            keep = source df ms
            rename=( df = _dfe_ ms = _mse_));
   drop source;
   merge MyOverAll (where=(source='Corrected Total')
                    keep = source DF);
   _limitn_ = (DF+1)/_limitk_;
   drop source DF;
run;
```

The data set ANOMParms contains a complete set of parameters, as shown in Output 5.6.1. Note these are the same values specified in the options for Example 5.5.

Output 5.6.1 Data Set ANOMParms

Parameters for ANOM for Effect of Position

var	_group_	_mean_	_alpha_	_limitk_	_dfe_	_mse_	_limitn_
concentration	position	13.53333	0.05	5	15	12.6000000	6

The following statements read the parameters in ANOMParms to create an ANOM chart for the effect of position:

```
ods graphics on;
title "ANOM for Effect of Position";
proc anom data=Cleaning limits=ANOMParms;
   xchart concentration * position /
      outtable = postable
      odstitle = title;
   label position      = 'Position'
         concentration = 'Mean of Concentration';
run;
```

The chart produced is identical to the one in Output 5.5.2. Note that the procedure creates a TABLE= input data set postable. You can use postable to create a combined chart for the two factors position and depth.

You can create a LIMITS= data set ANOMParmsB for the factor depth by using the above code and substituting 'depth' for the _group_ variable. You can use the OUTTABLE= statement to store the TABLE= input data set for depth as deptable. The resulting data set ANOMParmsB is shown in Output 5.6.2:

Output 5.6.2 Data Set ANOMParmsB

ANOM for Effect of Position

Obs	_var_	_group_	_mean_	_alpha_	_limitk_	_dfe_	_mse_	_limitn_
1	concentration	depth	13.53333	0.05	3	15	12.6000000	10

Example 5.7: ANOM for Cell Means in Presence of Interaction

NOTE: See *ANOM for Cell Means in the Presence of Interaction* in the SAS/QC Sample Library.

This example illustrates the use of analysis of means in an experiment with two factors where an interaction effect is present. The following data set CleaningInteract is a modified version of the data set Cleaning, which includes an interaction effect for position and depth.

Consider the following data set CleaningInteract:

```
data CleaningInteract;
   do position = 1 to 5;
      do depth = 1 to 3;
         do rep = 1 to 2;
            input concentration @@;
            output;
         end;
      end;
   end;
```

```
datalines;
15 16 15 14 19  5
15 16 14 14  0  1
19 15 16 16 11  8
18 16 24 23  8 14
15 12 23 24  8 11
;
```

The following statements use PROC GLM to test for an interaction:

```
ods graphics off;
proc glm data=CleaningInteract;
   class position depth;
   model concentration = position depth position*depth;
run;
```

The analysis of variance results in Output 5.7.1 indicate a significant interaction between position and depth.

Output 5.7.1 GLM Results

ANOM for Effect of Position

The GLM Procedure

Dependent Variable: concentration

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	885.666667	63.261905	6.66	0.0004
Error	15	142.500000	9.500000		
Corrected Total	29	1028.166667			

R-Square	Coeff Var	Root MSE	concentration Mean
0.861404	21.75676	3.082207	14.16667

Source	DF	Type I SS	Mean Square	F Value	Pr > F
position	4	169.0000000	42.2500000	4.45	0.0144
depth	2	515.4666667	257.7333333	27.13	<.0001
position*depth	8	201.2000000	25.1500000	2.65	0.0496

Since an interaction effect is present, an appropriate way to analyze the data is to create an ANOM chart for the cell means.

In order to create the chart you first need to compute the cell means and a new *group* variable which designates the cells. The following statements use PROC MEANS for this purpose.

```
proc means data=CleaningInteract n mean std;
   class position depth;
   var concentration;
   types position*depth;
   output out=cellmeans mean=concentrationX std=concentrationS;
run;
data cellmeans; set cellmeans;
   rename _FREQ_ = concentrationN;
```

```
    pos = put(position, z1.);
    dep = put(depth, z1.);
    cell  = cat('P',pos, 'D', dep);
    drop _TYPE_ pos dep;
run;
```

The cell means are stored in the data set cellmeans shown in Output 5.7.2:

Output 5.7.2 Data Set cellmeans

ANOM for Effect of Position

The MEANS Procedure

Analysis Variable : concentration					
position	depth	N Obs	N	Mean	Std Dev
1	1	2	2	15.5000000	0.7071068
	2	2	2	14.5000000	0.7071068
	3	2	2	12.0000000	9.8994949
2	1	2	2	15.5000000	0.7071068
	2	2	2	14.0000000	0
	3	2	2	0.5000000	0.7071068
3	1	2	2	17.0000000	2.8284271
	2	2	2	16.0000000	0
	3	2	2	9.5000000	2.1213203
4	1	2	2	17.0000000	1.4142136
	2	2	2	23.5000000	0.7071068
	3	2	2	11.0000000	4.2426407
5	1	2	2	13.5000000	2.1213203
	2	2	2	23.5000000	0.7071068
	3	2	2	9.5000000	2.1213203

ANOM for Effect of Position

position	depth	concentrationN	concentrationX	concentrationS	cell
1	1	2	15.5	0.70711	P1D1
1	2	2	14.5	0.70711	P1D2
1	3	2	12.0	9.89949	P1D3
2	1	2	15.5	0.70711	P2D1
2	2	2	14.0	0.00000	P2D2
2	3	2	0.5	0.70711	P2D3
3	1	2	17.0	2.82843	P3D1
3	2	2	16.0	0.00000	P3D2
3	3	2	9.5	2.12132	P3D3
4	1	2	17.0	1.41421	P4D1
4	2	2	23.5	0.70711	P4D2
4	3	2	11.0	4.24264	P4D3
5	1	2	13.5	2.12132	P5D1
5	2	2	23.5	0.70711	P5D2
5	3	2	9.5	2.12132	P5D3

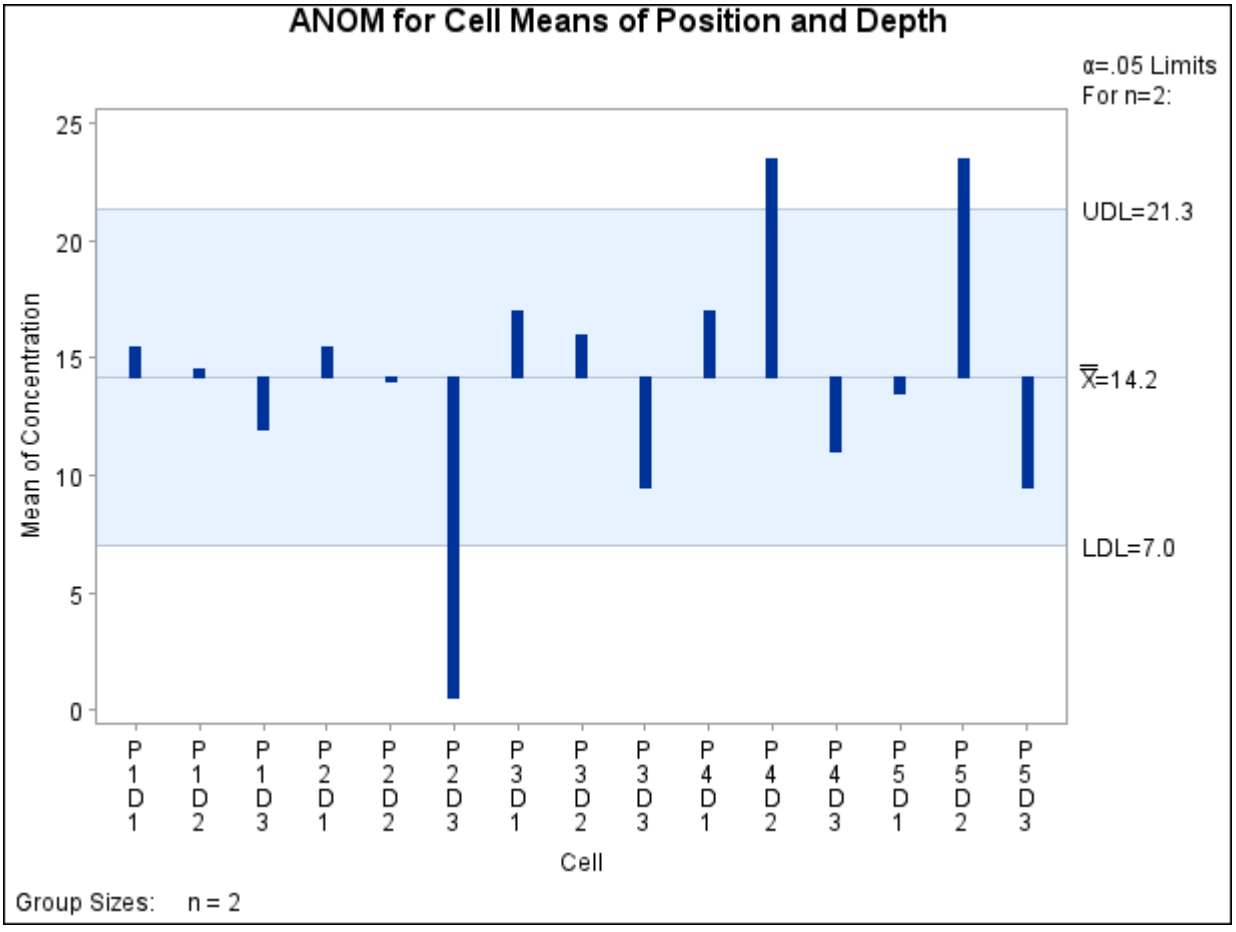
The data set cellmeans has the structure of a SUMMARY= input data set for the ANOM procedure. For details concerning a SUMMARY= data set, see the section "Creating ANOM Charts for Means from Group Summary Data" on page 133.

The following statements use cellmeans to create the ANOM chart for the cell means using SUMMARY= option:

```
ods graphics off;
title "ANOM for Cell Means of Position and Depth";
proc ANOM summary = cellmeans;
   xchart concentration * cell / turnhlabels;
   label concentrationX = 'Mean of Concentration';
   label cell           = 'Cell';
run;
```

The chart is shown in Output 5.7.3:

Output 5.7.3 ANOM for Cell Means of Position and Depth



The chart shows that the cell means for P2D3, P4D2, and P5D2 are significantly different from the average concentration level.

INSET Statement: ANOM Procedure

Overview: INSET Statement

The INSET statement enables you to enhance an ANOM chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. An inset can display statistics calculated by the ANOM procedure or arbitrary values provided in a SAS data set.

Note that an INSET statement by itself does not produce a display but must be used in conjunction with a chart statement.

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started: INSET Statement

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the section "Syntax: INSET Statement" on page 173.

Displaying Summary Statistics on an ANOM Chart

NOTE: See *Displaying Summary Statistics on an ANOM Chart* in the SAS/QC Sample Library.

A manufacturing engineer carries out a study to determine the source of excessive variation in the positioning of labels on shampoo bottles. [4] A labeling machine removes bottles from the line, attaches the labels, and returns the bottles to the line. There are six positions on the machine, and the engineer suspects that one or more of the position heads might be faulty.

A sample of 60 bottles, 10 per position, is run through the machine. For each bottle, the deviation of each label is measured in millimeters, and the machine position is recorded. The following statements create a SAS data set named LabelDeviations, which contains the deviation measurements for the 60 bottles:

[4] This example is based on a case study described by Hansen (1990).

```
data LabelDeviations;
   input Position @;
   do i = 1 to 5;
      input Deviation @;
      output;
   end;
   drop i;
   datalines;
1  -0.02386  -0.02853  -0.03001  -0.00428  -0.03623
1  -0.04222  -0.00144  -0.06466   0.00944  -0.00163
2  -0.02014  -0.02725   0.02268  -0.03323   0.03661
2   0.04378   0.05562   0.00977   0.05641   0.01816
3  -0.00728   0.02849  -0.04404  -0.02214  -0.01394
3   0.04855   0.03566   0.02345   0.01339  -0.00203
4   0.06694   0.10729   0.05974   0.06089   0.07551
4   0.03620   0.05614   0.08985   0.04175   0.05298
5   0.03677   0.00361   0.03736   0.01164  -0.00741
5   0.02495  -0.00803   0.03021  -0.00149  -0.04640
6   0.00493  -0.03839  -0.02037  -0.00487  -0.01202
6   0.00710  -0.03075   0.00167  -0.02845  -0.00697
;
```

The following statements generate an ANOM chart from the LabelDeviations data. An INSET statement is used to display the mean square error (MSE) and the number of groups outside of the decision limits (NOUT) on the chart:

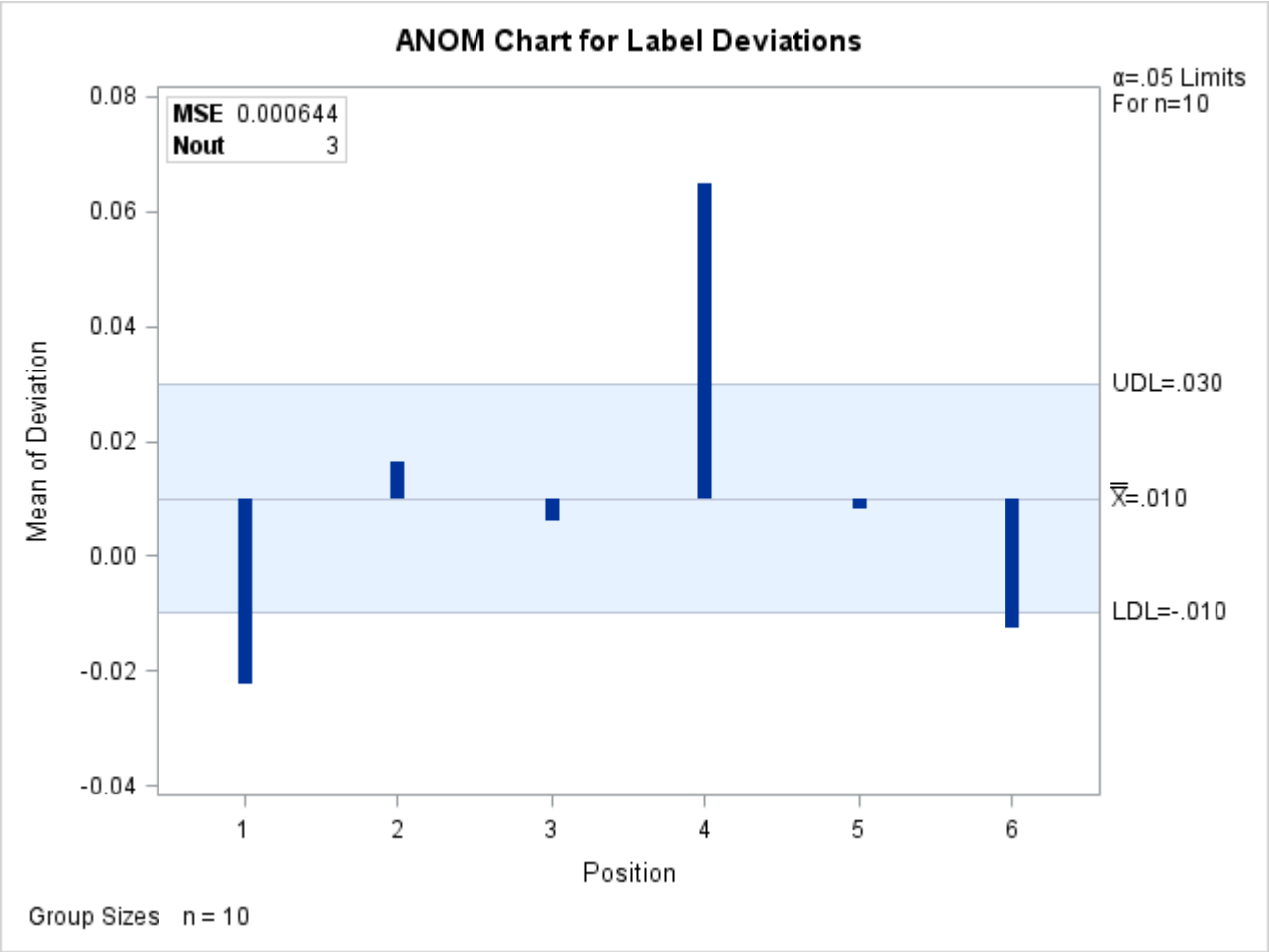
```
ods graphics on;
title 'ANOM Chart for Label Deviations';
proc anom data=LabelDeviations;
   xchart Deviation*Position / odstitle = title;
   inset mse nout / height = 3;
run;
```

The ODS GRAPHICS ON statement specified before the PROC ANOM statement enables ODS Graphics, so the chart is created using ODS Graphics instead of traditional graphics. The resulting ANOM chart is displayed in Figure 5.28.

The INSET statement immediately follows the chart statement that creates the graphical display (in this case, the XCHART statement). Specify the keywords for inset statistics (such as ALPHA) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords. The HEIGHT= option on the INSET statement specifies the text height used to display the statistics in the inset.

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 174. Note that the set of keywords available for a particular display may depend on both the chart statement that precedes the INSET statement and the options that you specify in the chart statement.

Figure 5.28 An ANOM Chart with an Inset



The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

NOTE: See *Formatting and Positioning the Inset on an ANOM Chart* in the SAS/QC Sample Library.

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 5.28 the default format used for the MSE prints an excessive number of decimal places. In the inset produced by the following statements, the unwanted decimal places are eliminated and the default MSE label is replaced by one specified by the user:

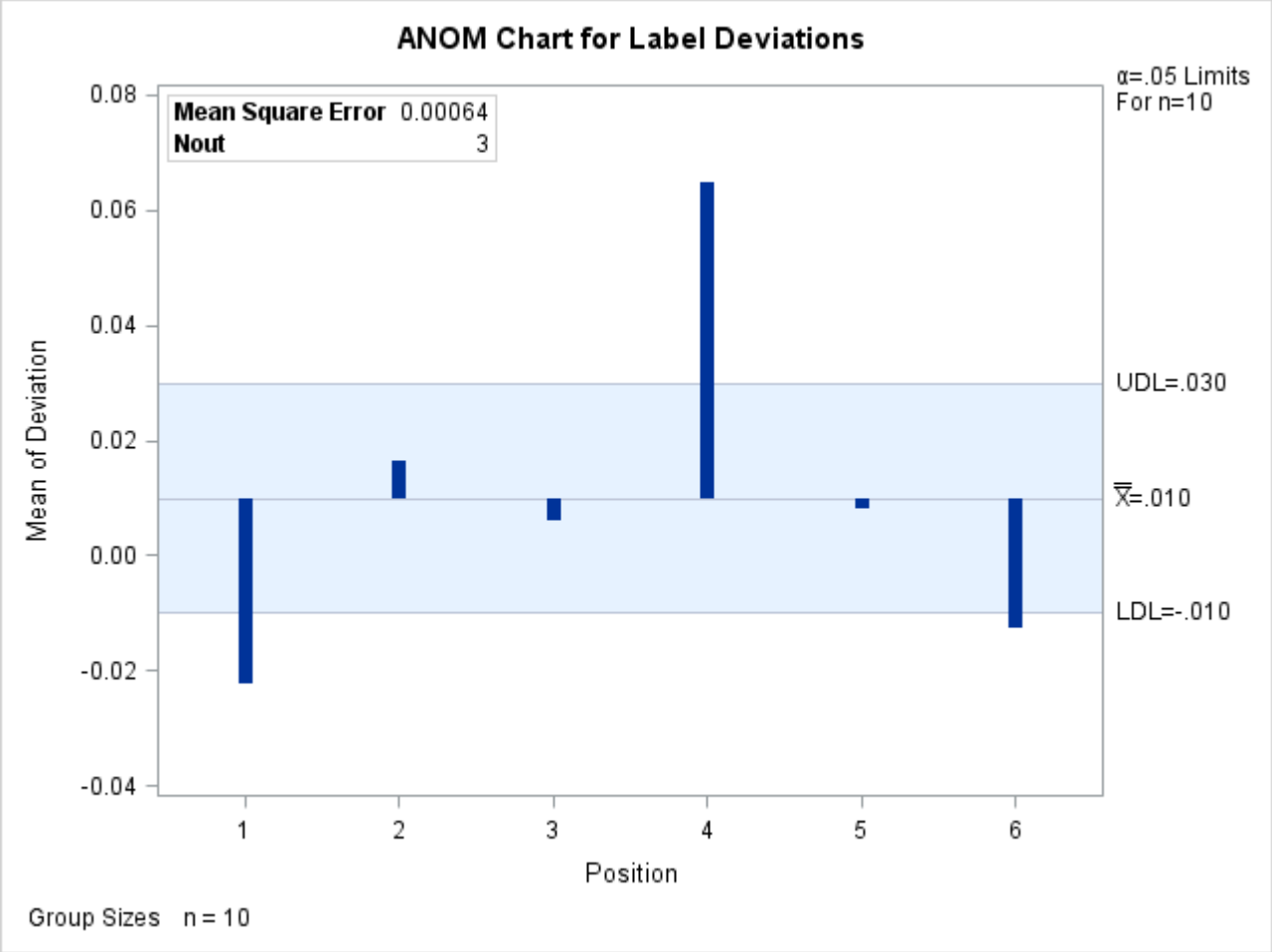
```
title 'ANOM Chart for Label Deviations';
proc anom data=LabelDeviations;
   xchart Deviation*Position / odstitle=title;
   inset mse='Mean Square Error' (7.5) nout;
run;
```

The resulting ANOM chart is displayed in Figure 5.29. You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 7.5 specified in parentheses after the MSE keyword displays the statistic with a field width of seven and five decimal places. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option (see Figure 5.30). For more information about SAS formats, refer to *SAS Formats and Informats: Reference*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format.

Figure 5.29 Formatting Values and Customizing Labels in an Inset



Adding a Header and Positioning the Inset

NOTE: See *Adding a Header and Positioning the Inset on an ANOM Chart* in the SAS/QC Sample Library.

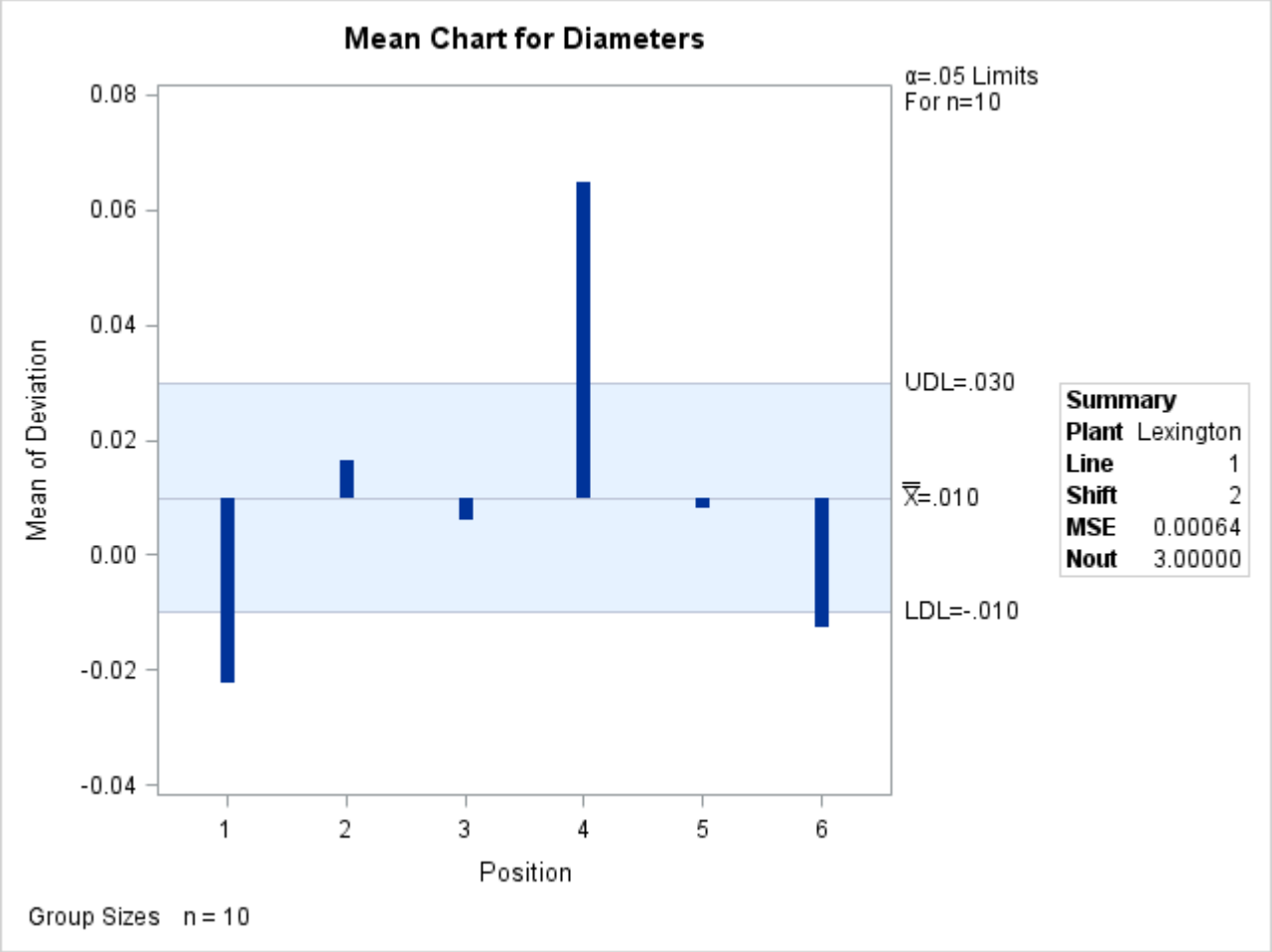
In the previous examples, the insets are displayed in the upper left corners of the plots, the default position for insets added to ANOM charts. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and the chart shown in Figure 5.30:

```
data Location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
   datalines;
Plant     Lexington
Line      1
Shift     2
;

title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position / odstitle=title;
   inset data=Location mse nout /
      format   = 8.5
      position = rm
      cshadow  = black
      height   = 3
      header   = 'Summary';
run;
```

The header (in this case, *Summary*) can be up to 40 characters. POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details: INSET Statement" on page 178. The CSHADOW= option is used to display a drop shadow on this inset. The *options*, such as HEADER=, POSITION=, and CSHADOW=, are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 176.

Note that the contents of the data set Location appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset. The FORMAT= option applies format 8.5 to the statistics listed in the INSET statement. Note that the format does not apply to the values from the Location data set. You can associate a format with the _VALUE_ variable in the data set to format those values.

Figure 5.30 Adding a Header and Repositioning the Inset



Syntax: INSET Statement

The syntax for the INSET statement is as follows:

INSET *keyword-list* < / *options* > **;**

You can use any number of INSET statements in the ANOM procedure. However, when ODS Graphics is enabled, at most two insets are displayed inside the plot area and at most two are displayed in the chart margins. Each INSET statement produces a separate inset and must follow one of the chart statements. The inset appears on every panel (page) produced by the last chart statement preceding it. The statistics are displayed in the order in which they are specified. The following statements produce an ANOM boxchart with two insets and an ANOM chart for means with one inset.

```
proc anom data=LabelDeviations;
   boxchart Deviation*Position;
      inset alpha mse dfe;
      inset ldl mean udl;
   xchart Deviation*Position;
      inset ngroups nmin nmax;
run;
```

The statistics displayed in an inset are computed for a specific response variable using observations for the current BY group. For example, in the following statements, there are two response variables (Weight and Diameter) and a BY variable (Location). If there are three different locations (levels of Location), then a total of six ANOM charts are produced. The statistics in each inset are computed for a particular variable and location. The labels in the inset are the same for each ANOM chart.

```
proc anom data=Axles;
   by Location;
   xchart (Weight Diameter)*Batch;
   inset alpha mse dfe;
run;
```

The components of the INSET statement are described as follows.

keyword-list

can include any of the *keywords* listed in "Summary of INSET Keywords" on page 174. By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 170.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
inset n nmin nmax / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 175 for a list of all available *options*, and "Dictionary of Options" on page 176 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

All keywords available with the ANOM procedure's INSET statement request a single statistic in an inset, except for the DATA= keyword. The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset. Figure 5.30 shows an inset containing entries from a DATA= data set.

Table 5.36 Summary Statistics

Keyword	Description
ALPHA	Significance level
DATA=	(Label, Value) pairs from *SAS-data-set*
DFE	Degrees of freedom
LDL	Lower decision limit
MEAN	Weighted average of group means
MSE	Mean square error
N	Nominal group size
NGROUPS	Number of groups
NHIGH	Number of groups above upper decision limit
NLOW	Number of groups below lower decision limit
NMAX	Maximum group size
NMIN	Minimum group size
NOBS	Total number of observations
NOUT	Total number of groups outside decision limits
RMSE	Root mean square error
UDL	Upper decision limit

You can use the keywords in Table 5.37 only when producing ODS Graphics output. The labels for the statistics use Greek letters.

Table 5.37 Keywords Specific to ODS Graphics Output

Keyword	Description
UALPHA	Probability of Type 1 error
UMU	Weighted average of group means

Summary of Options

The following table lists the INSET statement options. For complete descriptions, see "Dictionary of Options" on page 176.

Table 5.38 INSET Options

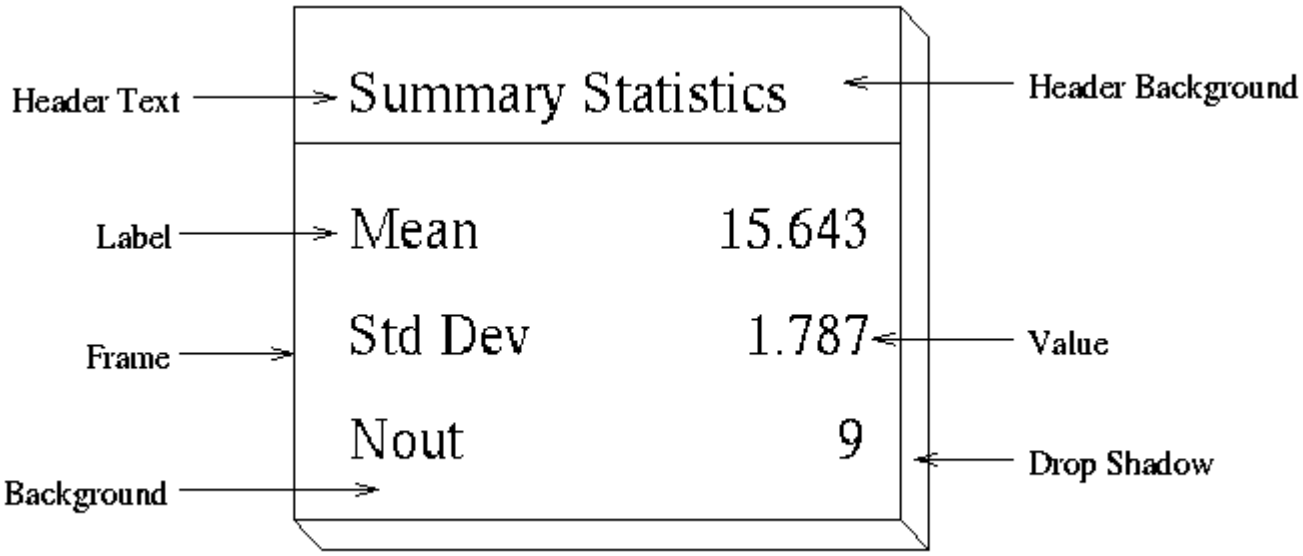
Option	Description
CFILL=	Specifies color of inset background
CFILLH=	Specifies color of header background
CFRAME=	Specifies color of frame
CHEADER=	Specifies color of header text
CSHADOW=	Specifies color of drop shadow
CTEXT=	Specifies color of inset text
DATA	Specifies data units for POSITION=(x, y) coordinates

Table 5.38 *continued*

Option	Description
FONT=	Specifies font of text
FORMAT=	Specifies format of values in inset
HEADER=	Specifies header text
HEIGHT=	Specifies height of inset text
NOFRAME	Suppresses frame around inset
POSITION=	Specifies position of inset
REFPOINT=	Specifies reference point of inset positioned with POSITION=(x, y) coordinates

The following sections provide detailed descriptions of options for the INSET statement. Terms used in this section are illustrated in Figure 5.31.

Figure 5.31 The Inset



Dictionary of Options

General Options

You can specify the following options whether you use ODS Graphics or traditional graphics:

DATA

specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 180. See Figure 5.33 for an example.

FONT=*font*

specifies the font used for text in the inset. By default, the font associated with the GraphLabelText style element is used for the inset header and that associated with the GraphValueText style element is used for text in the body of the inset.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER='*string*'

specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*height*

HEIGHT=SMALL

specifies the height of the text in the inset. By default, the GraphLabelText style element determines the size of inset header text and the GraphValueText style element determines the size of text in the body of the inset.

When you produce traditional graphics, you can specify the *height* in screen percent units to be used for text in both the header and the body of the inset.

When you produce ODS Graphics output, you can specify HEIGHT=SMALL to reduce the height of text in the inset. The GraphValueText size is used for the inset header and the GraphDataText size is used in the inset body.

NOFRAME

suppresses the frame drawn around the text.

POSITION=*position*

POS=*position*

determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details: INSET Statement" on page 178. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

NOTE: You cannot specify coordinates with the POSITION= option when producing ODS Graphics output.

REFPOINT=BR | BL | TR | TL

RP=BR | BL | TR | TL

specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 5.34 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 180.

Options for Traditional Graphics

You can specify the following options only when traditional graphics are produced. The ANOM procedure produces traditional graphics when ODS Graphics is disabled and SAS/GRAPH is licensed.

CFILL=*color* | BLANK
specifies the color of the background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as needles representing group data or decision limits) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Figure 5.30 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

Details: INSET Statement

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

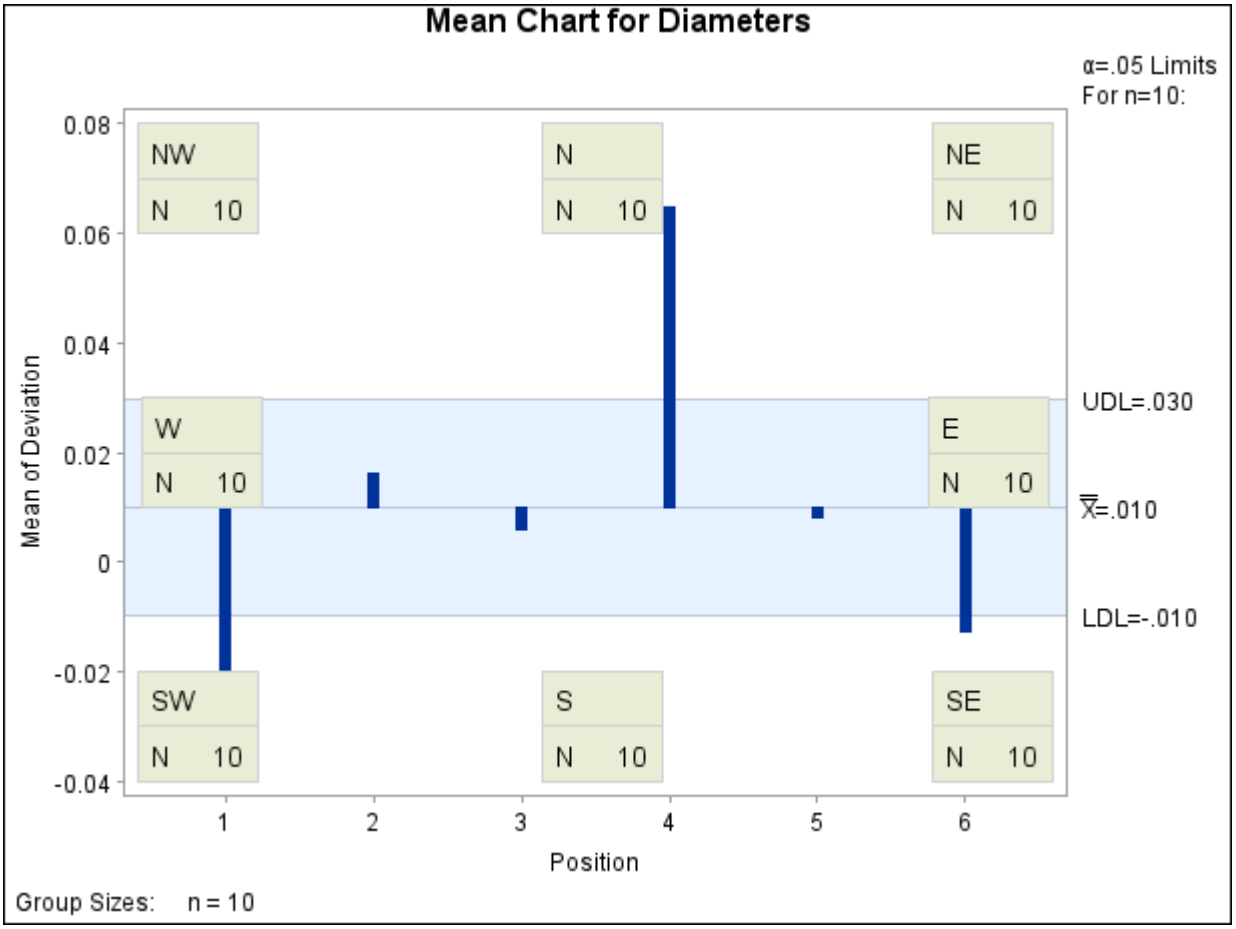
Positioning the Inset Using Compass Points

NOTE: See *Positioning the Inset on an ANOM Chart Using Compass Points* in the SAS/QC Sample Library.

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 5.32, which demonstrates all eight compass positions. The default is NW.

```
ods graphics off;
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset n / height=3 cfill=ywh header='NW' pos=nw;
   inset n / height=3 cfill=ywh header='N ' pos=n ;
   inset n / height=3 cfill=ywh header='NE' pos=ne;
   inset n / height=3 cfill=ywh header='E ' pos=e ;
   inset n / height=3 cfill=ywh header='SE' pos=se;
   inset n / height=3 cfill=ywh header='S ' pos=s ;
   inset n / height=3 cfill=ywh header='SW' pos=sw;
   inset n / height=3 cfill=ywh header='W ' pos=w ;
run;
```

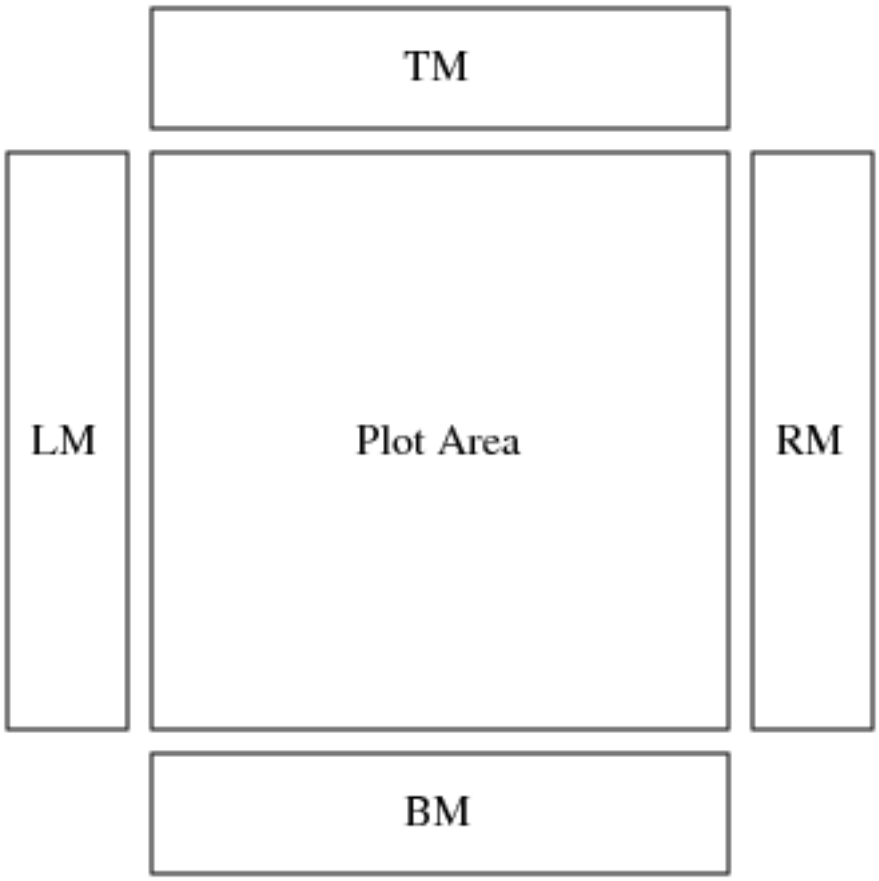
Figure 5.32 Insets Positioned Using Compass Points



Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 5.7.4.

Output 5.7.4 Positioning Insets in the Margins



For an example of an inset placed in the right margin, see Figure 5.30. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

NOTE: See *Positioning the Inset Using Coordinates on an ANOM Chart* in the SAS/QC Sample Library.

When you produce traditional graphics, you can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

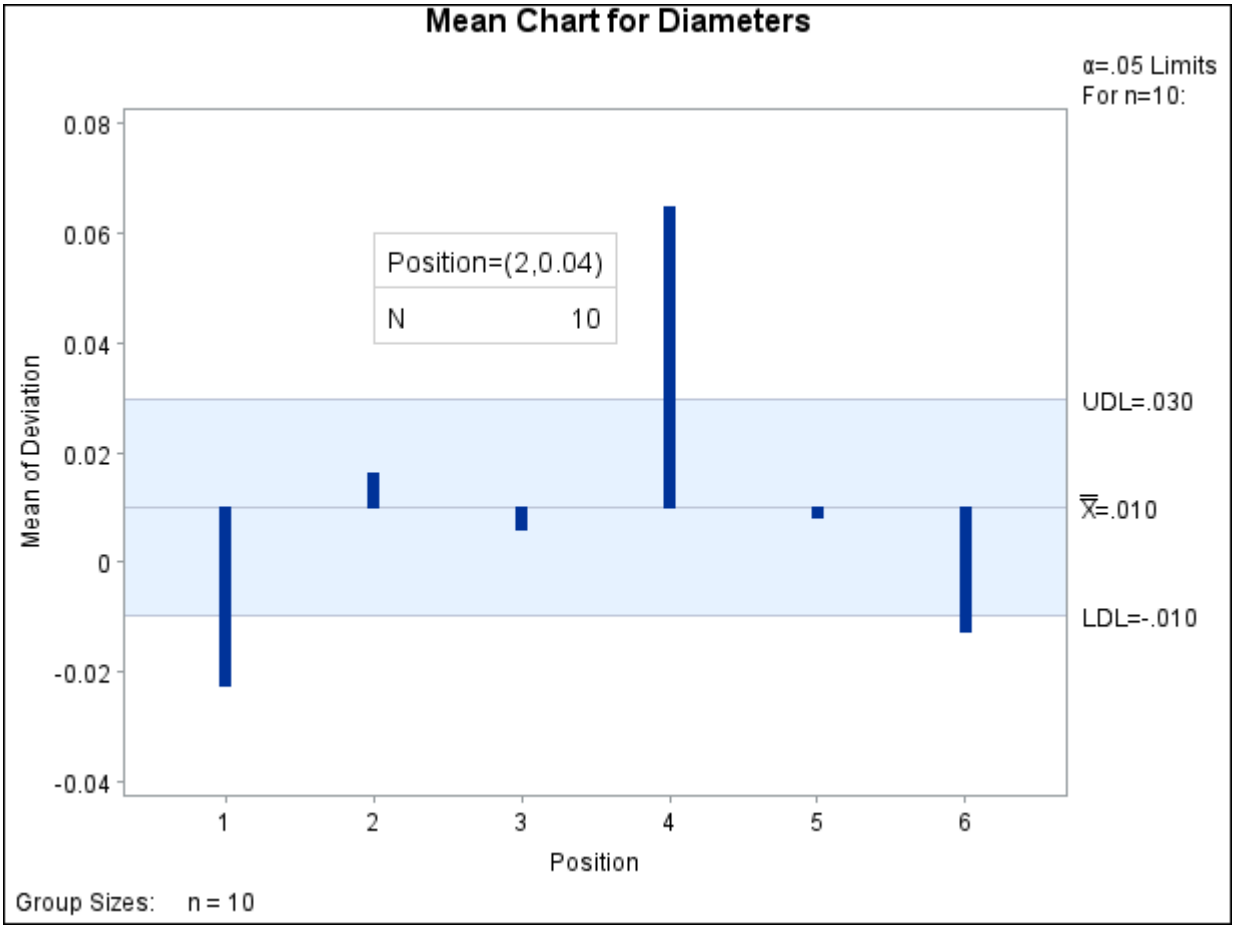
Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 2 on the horizontal axis and 0.04 on the vertical axis:

```
ods graphics off;
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset n /
      header   = 'Position=(2,0.04)'
      height   = 3
      position = (2,0.04) data;
run;
```

The ANOM chart is displayed in Figure 5.33. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

Figure 5.33 Inset Positioned Using Data Unit Coordinates



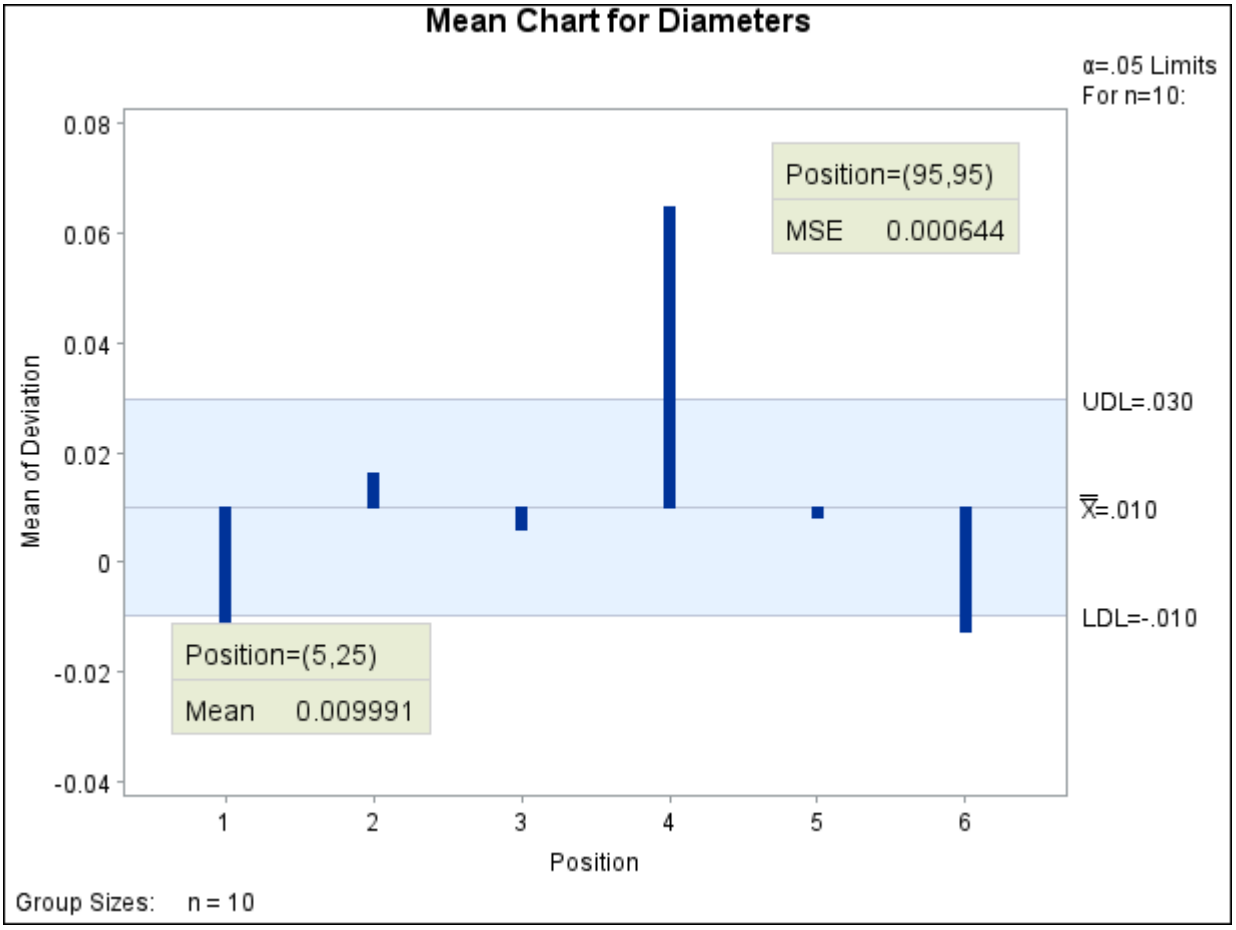
Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create an ANOM chart with two insets, both positioned using coordinates in axis percent units:

```
title 'Mean Chart for Diameters';
proc anom data=LabelDeviations;
   xchart Deviation*Position;
   inset mean / position = (5,25)
                header   = 'Position=(5,25)'
                height   = 3
                cfill    = ywh
                refpoint = tl;
   inset mse / position = (95,95)
               header   = 'Position=(95,95)'
               height   = 3
               cfill    = ywh
               refpoint = tr;
run;
```

The display is shown in Figure 5.34. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

Figure 5.34 Inset Positioned Using Axis Percent Unit Coordinates



Dictionary of ANOM Chart Statement Options

This section provides detailed descriptions of options that you can specify in the following chart statements:

- BOXCHART
- PCHART
- UCHART
- XCHART

Options that are common to the ANOM and SHEWHART procedures are listed in the "Summary of Options" subsection in the sections for each chart statement. They are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993

Options are specified after the slash (/) in a chart statement. For example, to place the label "Mean" on the center line of an ANOM chart, you can use the XSYMBOL= option as follows:

```
proc anom data=Measures;
   xchart Length*Sample / xsymbol='Mean';
run;
```

The options described in this section are listed alphabetically. For tables of options organized by function, see the "Summary of Options" sections in the sections for the various chart statements. Unless indicated otherwise, the options listed here are available with every chart statement.

ALPHA=*value*
specifies the probability of a Type I error.

CINFILL=*color* | **EMPTY** | **NONE**
specifies the color for the area inside the decision limits. By default, this area filled with an appropriate color from the ODS style. You can specify the keyword EMPTY or NONE to leave the area between the decision limits unfilled. If you specify a color, it is ignored when ODS Graphics is enabled.

CLIMITS=*color*
specifies the color for the decision limits, the central line, and related labels in traditional graphics. This option is ignored when ODS Graphics is enabled.

DFE=*n*
specifies the degrees of freedom *n* associated with the root mean square error.

GROUPN=*value*
GROUPN=*variable*
specifies the group sizes as a constant *value* or as the values of a variable in the DATA= data set. The GROUPN= option is available only in the PCHART and UCHART statements. You must specify GROUPN= in a PCHART or UCHART statement when your input data set is a DATA= data set.

If you specify multiple *responses* in a chart statement, the GROUPN= option is used with all of the *responses* listed.

LDLLABEL='*label*'
specifies a label for the lower decision limit in the ANOM chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form LDL=value if the decision limit has a fixed value; otherwise, the default label is LDL. A related option is UDLLABEL=.

LIMITK=*k*
specifies the number of groups for computing decision limits.

LIMITN=*n*
specifies either a nominal sample size for fixed decision limits or varying limits.

LIMLABSUBCHAR='*c*'
specifies a substitution character *c* for labels provided as quoted strings with the LDLLABEL=, UDLLABEL=, PSYMBOL=, USYMBOL=, and XSYMBOL= options. The substitution character must appear in the label. When the label is displayed on the chart, the character is replaced with the value of the corresponding decision limit or center line, provided that this value is constant across groups. Otherwise, the default label for a varying decision limit or center line is displayed.

LLIMITS=*linetype*
specifies the line type for decision limits in traditional graphics. This option is ignored when ODS Graphics is enabled.

MEAN=*value*
specifies the (known) mean of the response. This value is used for each response specified in the chart statement.

MSE=*value*
specifies the mean square error.

NDECIMAL=
specifies the number of digits to the right of decimal place in default labels for decision limits and central line

NOCTL
suppresses display of the central line.

NOLDL
suppresses display of the lower decision limit.

NOLIMITLABEL
suppresses labels for the decision limits and central line.

NOLIMIT0
suppresses display of the lower decision limit if it is 0.

NOLIMIT1
suppresses display of the upper decision limit if it is 1 (100%).

NOLIMITS
suppresses display of the decision limits.

NOLIMITSFRAME
suppresses the default frame around decision limit information when multiple sets of decision limits are read from a LIMITS= data set.

NOLIMITSLEGEND
suppresses the decision limits legend.

NONEEDLES
suppresses the needles connecting points to the center line.

NOREADLIMITS
specifies that the decision limits for each response listed in the chart statement are *not* to be read from the LIMITS= data set specified in the PROC ANOM statement. There are two basic methods of displaying decision limits: calculating decision limits from the data and reading decision limits from a LIMITS= data set. If you specify a LIMITS= data set but want the decision limits to be calculated from the data, specify the NOREADLIMITS option.

NOUDL
suppresses display of the upper decision limit.

OUTSUMMARY=*SAS-data-set*

OUT=*SAS-data-set*

OUTHISTORY=*SAS-data-set*
creates an output data set that contains group summary statistics. You can use an OUTSUMMARY= data set as a SUMMARY= input data set in a subsequent run of the procedure. You cannot request an OUTSUMMARY= data set if the input data set is a TABLE= data set. See "Output Data Sets" in the section for the chart statement in which you are interested.

P=
specifies the weighted average of group proportions.

PSYMBOL='*label***'**
specifies the label for the central line on an ANOM *p* chart.

READINDEXES=
reads multiple sets of decision limits for each *response* from a LIMITS= data set.

TYPE=
identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set.

U=
specifies the weighted average of group rates.

UDLLABEL=
specifies the label for the upper decision limit.

USYMBOL='*label***'**
specifies the label for the central line on an ANOM *u* chart.

WLIMITS=
specifies the width for the decision limits and central line in traditional graphics. This option is ignored when ODS Graphics is enabled.

XSYMBOL='*label***'**
specifies the label for the central line on an ANOM chart or ANOM boxchart.

References

Fritzsch, K., and Hsu, J. C. (1997). "On Analysis of Means." In *Advances in Statistical Decision Theory and Methodology*, edited by S. Panchapakesan and N. Balakrishnan, 114–119. Boston: Birkhäuser.

Halperin, M., Greenhouse, S. W., Cornfield, J., and Zalokar, J. (1955). "Tables of Percentage Points for the Studentized Maximum Absolute Deviate in Normal Samples." *Journal of the American Statistical Association* 50:185–195.

Hansen, E. (1990). "Making the 'Complex' Simple." In *Problem-Driven Case Studies in Quality Improvement: Second Annual Symposium*, 7.1.1–7.1.21. Madison: Center for Quality and Productivity Improvement, University of Wisconsin.

Laplace, P. S. (1827). "Mémoire sur le flux et reflux lunaire atmosphérique." *Connaissance des Temps pour l'An 1830,* 3–18.

Nelson, L. S. (1983). "Exact Critical Values for Use with the Analysis of Means." *Journal of Quality Technology* 15:40–44.

Nelson, P. R. (1981). "Numerical Evaluation of an Equicorrelated Multivariate Non-central *t* Distribution." *Communications in Statistics—Simulation and Computation* 10:41–50.

Nelson, P. R. (1982a). "Exact Critical Points for the Analysis of Means." *Communications in Statistics—Theory and Methods* 11:699–709.

Nelson, P. R. (1982b). "Multivariate Normal and *t* Distributions with $\rho_{jk} = \alpha_j \alpha_k$." *Communications in Statistics—Simulation and Computation* 11:239–248.

Nelson, P. R. (1991). "Numerical Evaluation of Multivariate Normal Integrals with Correlations $\rho_{lj} = -\alpha_l \alpha_j$." In *Frontiers of Statistical Scientific Theory and Industrial Applications: Proceedings of the ICOSCO I Conference*, edited by A. Öztürk and E. C. van der Meulen, 97–114. Columbus, OH: American Sciences Press.

Nelson, P. R. (1993). "Additional Uses for the Analysis of Means and Extended Tables of Critical Values." *Technometrics* 35:61–71.

Nelson, P. R., Coffin, M., and Copeland, K. A. F. (2003). *Introductory Statistics for Engineering Experimentation*. San Diego: Academic Press/Elsevier.

Ott, E. R. (1967). "Analysis of Means: A Graphical Procedure." *Industrial Quality Control* 24:101–109. Reprinted in *Journal of Quality Technology* 15 (1983): 10–18.

Ott, E. R. (1975). *Process Quality Control: Troubleshooting and Interpretation of Data*. New York: McGraw-Hill.

Ramig, P. F. (1983). "Application of the Analysis of Means." *Journal of Quality Technology* 15:19–25.

Rodriguez, R. N. (1996). "Health Care Applications of Statistical Process Control: Examples Using the SAS System." In *Proceedings of the Twenty-First Annual SAS Users Group International Conference*, 1381–1396. Cary, NC: SAS Institute Inc. http://www.sascommunity.org/sugi/SUGI96/Sugi-96-229%20Rodriguez.pdf.

Soong, W. C., and Hsu, J. C. (1997). "Using Complex Integration to Compute Multivariate Normal Probabilities." *Journal of Computational and Graphical Statistics* 6:397–415.

Chapter 6
The CAPABILITY Procedure

Contents

Introduction: CAPABILITY Procedure **193**
Learning about the CAPABILITY Procedure 194
PROC CAPABILITY and General Statements **195**
Overview: CAPABILITY Procedure 195
Getting Started: CAPABILITY Procedure 197
Computing Descriptive Statistics 197
Computing Capability Indices 199
Syntax: CAPABILITY Procedure 201
PROC CAPABILITY Statement 201
BY Statement 212
CLASS Statement 212
FREQ Statement 214
ID Statement 214
SPEC Statement 214
VAR Statement 218
WEIGHT Statement 218
Graphical Enhancement Statements 219
Details: CAPABILITY Procedure 219
Input Data Sets 219
Output Data Set 222
Descriptive Statistics 224
Signed Rank Statistic 227
Tests for Normality 227
Percentile Computations 230
Robust Estimators 232
Computing the Mode 234
Assumptions and Terminology for Capability Indices 235
Standard Capability Indices 235
Specialized Capability Indices 239
Missing Values 246
ODS Tables 247
Examples: CAPABILITY Procedure 248
Example 6.1: Reading Specification Limits 248
Example 6.2: Enhancing Reference Lines 251
Example 6.3: Displaying a Confidence Interval for Cpk 253
CDFPLOT Statement: CAPABILITY Procedure **255**

Overview: CDFPLOT Statement . . . 255
Getting Started: CDFPLOT Statement . . . 256
Creating a Cumulative Distribution Plot . . . 256
Syntax: CDFPLOT Statement . . . 257
Summary of Options . . . 258
Dictionary of Options . . . 263
Details: CDFPLOT Statement . . . 270
ODS Graphics . . . 270
Examples: CDFPLOT Statement . . . 271
Example 6.4: Fitting a Normal Distribution . . . 271
Example 6.5: Using Reference Lines with CDF Plots . . . 273
COMPHISTOGRAM Statement: CAPABILITY Procedure . . . 274
Overview: COMPHISTOGRAM Statement . . . 274
Getting Started: COMPHISTOGRAM Statement . . . 275
Creating a One-Way Comparative Histogram . . . 276
Adding Fitted Normal Curves to a Comparative Histogram . . . 277
Syntax: COMPHISTOGRAM Statement . . . 278
Summary of Options . . . 280
Dictionary of Options . . . 284
Details: COMPHISTOGRAM Statement . . . 293
ODS Graphics . . . 293
Examples: COMPHISTOGRAM Statement . . . 294
Example 6.6: Adding Insets with Descriptive Statistics . . . 294
Example 6.7: Creating a Two-Way Comparative Histogram . . . 296
HISTOGRAM Statement: CAPABILITY Procedure . . . 299
Overview: HISTOGRAM Statement . . . 299
Getting Started: HISTOGRAM Statement . . . 300
Creating a Histogram with Specification Limits . . . 300
Adding a Normal Curve to the Histogram . . . 301
Customizing a Histogram . . . 304
Syntax: HISTOGRAM Statement . . . 305
Summary of Options . . . 306
Dictionary of Options . . . 314
Details: HISTOGRAM Statement . . . 336
Formulas for Fitted Curves . . . 336
Kernel Density Estimates . . . 347
Printed Output . . . 348
Output Data Sets . . . 356
ODS Tables . . . 359
ODS Graphics . . . 360
SYMBOL and PATTERN Statement Options . . . 360
Examples: HISTOGRAM Statement . . . 363
Example 6.8: Fitting a Beta Curve . . . 363
Example 6.9: Fitting Lognormal, Weibull, and Gamma Curves . . . 367

Example 6.10: Comparing Goodness-of-Fit Tests 372
Example 6.11: Computing Capability Indices for Nonnormal Distributions 374
Example 6.12: Computing Kernel Density Estimates 375
Example 6.13: Fitting a Three-Parameter Lognormal Curve 377
Example 6.14: Annotating a Folded Normal Curve 379
INSET Statement: CAPABILITY Procedure **385**
Overview: INSET Statement 385
Getting Started: INSET Statement 386
Displaying Summary Statistics on a Histogram 386
Formatting Values and Customizing Labels 387
Adding a Header and Positioning the Inset 389
Syntax: INSET Statement 390
Summary of INSET Keywords 392
Summary of Options 402
Dictionary of Options 402
Details: INSET Statement 405
Positioning the Inset Using Compass Points 405
Positioning the Inset in the Margins 406
Positioning the Inset Using Coordinates 407
Examples: INSET Statement 410
Example 6.15: Inset for Goodness-of-Fit Statistics 410
Example 6.16: Inset for Areas Under a Fitted Curve 411
INTERVALS Statement: CAPABILITY Procedure **413**
Overview: INTERVALS Statement 413
Getting Started: INTERVALS Statement 413
Computing Statistical Intervals 413
Computing One-Sided Lower Prediction Limits 416
Syntax: INTERVALS Statement 417
Summary of Options 417
Dictionary of Options 418
Details: INTERVALS Statement 420
Methods for Computing Statistical Intervals 420
OUTINTERVALS= Data Set 423
ODS Tables 423
OUTPUT Statement: CAPABILITY Procedure **424**
Overview: OUTPUT Statement 424
Getting Started: OUTPUT Statement 424
Saving Summary Statistics in an Output Data Set 424
Saving Percentiles in an Output Data Set 426
Syntax: OUTPUT Statement 427
Details: OUTPUT Statement 433
OUT= Data Set 433
Examples: OUTPUT Statement 434
Example 6.17: Computing Nonstandard Capability Indices 434

Example 6.18: Approximate Confidence Limits for Cpk 436
PPPLOT Statement: CAPABILITY Procedure **439**
Overview: PPPLOT Statement 439
Getting Started: PPPLOT Statement 440
Creating a Normal Probability-Probability Plot 440
Syntax: PPPLOT Statement 442
Summary of Options 443
Dictionary of Options 447
Details: PPPLOT Statement 455
Construction and Interpretation of P-P Plots 455
Comparison of P-P Plots and Q-Q Plots 458
Summary of Theoretical Distributions 459
Specification of Symbol Markers 460
Specification of the Distribution Reference Line 460
ODS Graphics 461
PROBPLOT Statement: CAPABILITY Procedure **461**
Overview: PROBPLOT Statement 461
Getting Started: PROBPLOT Statement 462
Creating a Normal Probability Plot 463
Creating Lognormal Probability Plots 464
Syntax: PROBPLOT Statement 468
Summary of Options 469
Dictionary of Options 473
Details: PROBPLOT Statement 486
Summary of Theoretical Distributions 486
SYMBOL Statement Options 488
ODS Graphics 489
Examples: PROBPLOT Statement 490
Example 6.19: Displaying a Normal Reference Line 490
Example 6.20: Displaying a Lognormal Reference Line 491
QQPLOT Statement: CAPABILITY Procedure **493**
Overview: QQPLOT Statement 493
Getting Started: QQPLOT Statement 494
Creating a Normal Quantile-Quantile Plot 494
Adding a Distribution Reference Line 495
Syntax: QQPLOT Statement 497
Summary of Options 498
Dictionary of Options 502
Details: QQPLOT Statement 516
Construction of Quantile-Quantile and Probability Plots 516
Interpretation of Quantile-Quantile and Probability Plots 517
Summary of Theoretical Distributions 518
Graphical Estimation 519
SYMBOL Statement Options 521

ODS Graphics 522
Examples: QQPLOT Statement 523
Example 6.21: Interpreting a Normal Q-Q Plot of Nonnormal Data 523
Example 6.22: Estimating Parameters from Lognormal Plots 525
Example 6.23: Comparing Weibull Q-Q Plots 530
Example 6.24: Estimating Cpk from a Normal Q-Q Plot 532
Dictionary of Common Options: CAPABILITY Procedure **534**
General Options 534
Options for Traditional Graphics 539
Options for Legacy Line Printer Charts 542
References **542**

Introduction: CAPABILITY Procedure

A process capability analysis compares the distribution of output from an in-control process to its specification limits to determine the consistency with which the specifications can be met. The CAPABILITY procedure provides the following:

- process capability indices, such as C_p and C_{pk}
- descriptive statistics based on moments, including skewness and kurtosis. Other descriptive information provided includes quantiles or percentiles (such as the median), frequency tables, and details on extreme values.
- histograms. Optionally, these can be superimposed with specification limits, fitted probability density curves for various distributions, and kernel density estimates.
- cumulative distribution function plots (cdf plots). Optionally, these can be superimposed with specification limits and probability distribution curves for various distributions.
- quantile-quantile plots (Q-Q plots), probability plots, and probability-probability plots (P-P plots). These plots facilitate the comparison of a data distribution with various theoretical distributions. Optionally, Q-Q plots and probability plots can be superimposed with specification limits.
- comparative histograms, cdf plots, Q=Q plots, probability plots, and P-P plots. These are composite graphs that are composed of plots that correspond to the different levels of specified CLASS variables.
- goodness-of-fit tests for a variety of distributions including the normal. The assumption of normality is critical to the interpretation of capability indices.
- statistical intervals (prediction, tolerance, and confidence intervals) for a normal population
- the ability to produce plots either as traditional graphics, ODS Graphics output, or legacy line printer plots. Traditional graphics can be saved, replayed, and annotated.
- the ability to inset summary statistics and capability indices in graphical output

- the ability to analyze data sets with a frequency variable
- the ability to read specification limits from a data set
- the ability to create output data sets containing summary statistics, capability indices, histogram intervals, parameters of fitted curves, and statistical intervals

You can use the PROC CAPABILITY statement, together with the VAR and SPEC statements, to compute summary statistics and process capability indices. See "Getting Started: CAPABILITY Procedure" on page 197 for introductory examples. In addition, you can use the statements summarized in Table 6.1 to request plots and specialized analyses:

Table 6.1 Statements for Plots and Specialized Analyses

Statement	Result
CDFPLOT	cumulative distribution function plot
COMPHISTOGRAM	comparative histogram
HISTOGRAM	histogram
INSET	inset table on plot
INTERVALS	statistical intervals
OUTPUT	output data set with summary statistics and capability indices
PPPLOT	probability-probability plot
PROBPLOT	probability plot
QQPLOT	quantile-quantile plot

You have three alternatives for producing plots with the CAPABILITY procedure:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

You can use the INSET statement with any of the plot statements to enhance the plot with an inset table of summary statistics. The INSET statement is not applicable when you produce line printer plots.

Learning about the CAPABILITY Procedure

To learn about the CAPABILITY procedure, first select the appropriate statement Table 6.1. Then refer to the corresponding "Getting Started" section for introductory examples:

- "Getting Started: CDFPLOT Statement" on page 256

- "Getting Started: COMPHISTOGRAM Statement" on page 275
- "Getting Started: HISTOGRAM Statement" on page 300
- "Getting Started: INSET Statement" on page 386
- "Getting Started: INTERVALS Statement" on page 413
- "Getting Started: OUTPUT Statement" on page 424
- "Getting Started: PPPLOT Statement" on page 440
- "Getting Started: PROBPLOT Statement" on page 462
- "Getting Started: QQPLOT Statement" on page 494

To broaden your knowledge of the procedure, read "PROC CAPABILITY and General Statements" on page 195 which summarizes the syntax for the entire procedure and describes the PROC CAPABILITY statement, the VAR statement, the CLASS statement, and the SPEC statement. Subsequent chapters describe the statements listed in Table 6.1. In addition to introductory examples, each chapter provides syntax summaries, descriptions of options, computational details, and advanced examples. Although the chapters are self-contained, much of what you learn about one plot statement, including the syntax, is transferable to other plot statements.

PROC CAPABILITY and General Statements

Overview: CAPABILITY Procedure

This chapter describes several statements that are generally used with the CAPABILITY procedure:

- The PROC CAPABILITY statement is required to invoke the CAPABILITY procedure. You can use this statement by itself to compute summary statistics.
- The VAR statement, which is optional, specifies the variables in the input data set that are to be analyzed. These are called the analysis or *process* variables. By default, all of the numeric variables are analyzed.
- The CLASS statement, which is optional, specifies one or two variables that group the data into classification levels. A separate analysis is carried out for each combination of levels, and you can use the CLASS statement with plot statements (such as HISTOGRAM) to create comparative displays. [1]
- The SPEC statement, which is optional, provides specification limits for the variables that are to be analyzed. When you use a SPEC statement, the procedure computes process capability indices in addition to summary statistics. Furthermore, the specification limits are displayed in plots created with plot statements that are described in subsequent chapters.

[1] You can use the COMPHISTOGRAM statement to create comparative histograms without applying classification levels to the overall analysis.

You can use the PROC CAPABILITY statement to request a variety of statistics for summarizing the data distribution of each analysis variable:

- sample moments
- basic measures of location and variability
- confidence intervals for the mean, standard deviation, and variance
- tests for location
- tests for normality
- trimmed and Winsorized means
- robust estimates of scale
- quantiles and related confidence intervals
- extreme observations and extreme values
- frequency counts for observations
- missing values

You can use the PROC CAPABILITY and SPEC statements together to request a variety of statistics for process capability analysis:

- percents of measurements within and outside specification limits
- confidence intervals for the probabilities of exceeding the specification limits
- standard capability indices and related confidence intervals
- tests of normality in conjunction with capability indices
- specialized capability indices

In addition, you can use options in the PROC CAPABILITY statement to

- specify the input data set to be analyzed
- specify an input data set containing specification limits
- specify a graphics catalog for saving traditional graphics output
- specify rounding units for variable values
- specify the definition used to calculate percentiles
- specify the divisor used to calculate variances and standard deviations
- request legacy line printer plots and define special printing characters used for features

- suppress tables

You can use options in the SPEC statement to

- provide lower and upper specification limits and target values
- control the appearance of specification lines on plots
- control the appearance of the areas under a histogram outside the specification limits

Getting Started: CAPABILITY Procedure

This section introduces the PROC CAPABILITY, VAR, and SPEC statements with examples that illustrate the most commonly used options.

Computing Descriptive Statistics

NOTE: See *Computing Summary Stats and Capability Indices* in the SAS/QC Sample Library.

The fluid weights of 100 drink cans are measured in ounces. The filling process is assumed to be in statistical control. The measurements are saved in a SAS data set named Cans.

```
data Cans;
   label Weight = "Fluid Weight (ounces)";
   input Weight @@;
   datalines;
12.07  12.02  12.00  12.01  11.98  11.96  12.04  12.05  12.01  11.97
12.03  12.03  12.00  12.04  11.96  12.02  12.06  12.00  12.02  11.91
12.05  11.98  11.91  12.01  12.06  12.02  12.05  11.90  12.07  11.98
12.02  12.11  12.00  11.99  11.95  11.98  12.05  12.00  12.10  12.04
12.06  12.04  11.99  12.06  11.99  12.07  11.96  11.97  12.00  11.97
12.09  11.99  11.95  11.99  11.99  11.96  11.94  12.03  12.09  12.03
11.99  12.00  12.05  12.04  12.05  12.01  11.97  11.93  12.00  11.97
12.13  12.07  12.00  11.96  11.99  11.97  12.05  11.94  11.99  12.02
11.95  11.99  11.91  12.06  12.03  12.06  12.05  12.04  12.03  11.98
12.05  12.05  12.11  11.96  12.00  11.96  11.96  12.00  12.01  11.98
;
```

You can use the PROC CAPABILITY and VAR statements to compute summary statistics for the weights.

```
title 'Process Capability Analysis of Fluid Weight';
proc capability data=Cans normaltest;
   var Weight;
run;
```

The input data set is specified with the DATA= option. The NORMALTEST option requests tests for normality. The VAR statement specifies the variables to analyze. If you omit the VAR statement, all numeric variables in the input data set are analyzed.

The descriptive statistics for Weight are shown in Figure 6.1. For instance, the average weight (labeled *Mean*) is 12.0093. The Shapiro-Wilk test statistic labeled *W* is 0.987876, and the probability of a more extreme

value of W (labeled $Pr < W$) is 0.499. Compared to the usual cutoff value of 0.05, this probability (referred to as a p-value) indicates that the weights are normally distributed.

Figure 6.1 Descriptive Statistics

Process Capability Analysis of Fluid Weight

The CAPABILITY Procedure
Variable: Weight (Fluid Weight (ounces))

Moments			
N	100	**Sum Weights**	100
Mean	12.0093	**Sum Observations**	1200.93
Std Deviation	0.04695269	**Variance**	0.00220456
Skewness	0.05928405	**Kurtosis**	-0.1717404
Uncorrected SS	14422.5469	**Corrected SS**	0.218251
Coeff Variation	0.39096946	**Std Error Mean**	0.00469527

Basic Statistical Measures			
Location		**Variability**	
Mean	12.00930	**Std Deviation**	0.04695
Median	12.00000	**Variance**	0.00220
Mode	12.00000	**Range**	0.23000
		Interquartile Range	0.07000

Tests for Location: Mu0=0				
Test		**Statistic**	**p Value**	
Student's t	**t**	2557.745	**Pr > |t|**	<.0001
Sign	**M**	50	**Pr >= |M|**	<.0001
Signed Rank	**S**	2525	**Pr >= |S|**	<.0001

Tests for Normality				
Test		**Statistic**	**p Value**	
Shapiro-Wilk	**W**	0.987876	**Pr < W**	0.4991
Kolmogorov-Smirnov	**D**	0.088506	**Pr > D**	0.0522
Cramer-von Mises	**W-Sq**	0.079055	**Pr > W-Sq**	0.2179
Anderson-Darling	**A-Sq**	0.457672	**Pr > A-Sq**	>0.2500

Quantiles (Definition 5)	
Level	**Quantile**
100% Max	12.130
99%	12.120
95%	12.090
90%	12.065
75% Q3	12.050
50% Median	12.000
25% Q1	11.980
10%	11.955
5%	11.935
1%	11.905
0% Min	11.900

Figure 6.1 *continued*

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
11.90	28	12.09	59
11.91	83	12.10	39
11.91	23	12.11	32
11.91	20	12.11	93
11.93	68	12.13	71

Computing Capability Indices

NOTE: See *Computing Summary Stats and Capability Indices* in the SAS/QC Sample Library.

This example is a continuation of the previous example and shows how you can provide specification limits with a SPEC statement to request capability indices in addition to descriptive statistics.

```
proc capability data=Cans normaltest freq;
   spec lsl=11.95 target=12 usl=12.05;
   var Weight;
run;
```

The options LSL=, TARGET=, and USL= specify the lower specification limit, target value, and upper specification limit for the weights. These statements produce the output shown in Figure 6.2 in addition to the output shown in Figure 6.1.

Figure 6.2 Capability Indices and Frequency Table

Process Capability Analysis of Fluid Weight

The CAPABILITY Procedure
Variable: Weight (Fluid Weight (ounces))

Specification Limits			
Limit		Percent	
Lower (LSL)	11.95000	% < LSL	7.00000
Target	12.00000	% Between	77.00000
Upper (USL)	12.05000	% > USL	16.00000

Process Capability Indices			
Index	Value	95% Confidence Limits	
Cp	0.354967	0.305565	0.404288
CPL	0.420991	0.332644	0.508117
CPU	0.288943	0.211699	0.365112
Cpk	0.288943	0.212210	0.365677
Cpm	0.348203	0.301472	0.398228

Figure 6.2 *continued*

Frequency Counts			
		Percents	
Value	Count	Cell	Cum
11.90	1	1.0	1.0
11.91	3	3.0	4.0
11.93	1	1.0	5.0
11.94	2	2.0	7.0
11.95	3	3.0	10.0
11.96	8	8.0	18.0
11.97	6	6.0	24.0
11.98	6	6.0	30.0
11.99	10	10.0	40.0
12.00	11	11.0	51.0
12.01	5	5.0	56.0
12.02	6	6.0	62.0
12.03	6	6.0	68.0
12.04	6	6.0	74.0
12.05	10	10.0	84.0
12.06	6	6.0	90.0
12.07	4	4.0	94.0
12.09	2	2.0	96.0
12.10	1	1.0	97.0
12.11	2	2.0	99.0
12.13	1	1.0	100.0

In Figure 6.2, the table labeled *Specification Limits* lists the specification limits and target value, together with the percents of observations outside and between the limits. The table labeled *Process Capability Indices* lists estimates for the standard process capability indices C_p, CPL, CPU, C_{pk}, and C_{pm}, along with 95% confidence limits. The index C_{pm} is not computed unless you specify a TARGET= value. See "Standard Capability Indices" on page 235 for formulas used to compute the indices.

If you specify more than one variable in the VAR statement, you can provide corresponding specification limits and target values by specifying lists of values for the LSL=, USL=, and TARGET= options. As an alternative to the SPEC statement, you can read specification limits and target values from a data set specified with the SPEC= option in the PROC CAPABILITY statement. This is illustrated in Example 6.1.

The FREQ option in the PROC CAPABILITY statement requests the table labeled *Frequency Counts* in Figure 6.2.

Syntax: CAPABILITY Procedure

The following are the primary statements that control the CAPABILITY procedure:

PROC CAPABILITY < *options* > **;**
BY *variables* **;**
CDFPLOT < *variables* > < / *options* > **;**
CLASS *variable-1* < *variable-2* >< / *options* > **;**
COMPHISTOGRAM < *variables* > / **CLASS= (***class-variables***)** < *options* > **;**
FREQ *variable* **;**
HISTOGRAM < *variables* > < / *options* > **;**
ID *variables* **;**
INSET *keyword-list* < / *options* > **;**
INTERVALS < *variables* > < / *options* > **;**
OUTPUT < **OUT=** *SAS-data-set* > < *keyword1=names . . . keywordk=names* > **;**
PPPLOT < *variables* > < / *options* > **;**
PROBPLOT < *variables* > < / *options* > **;**
QQPLOT < *variables* > < / *options* > **;**
SPEC < *options* > **;**
VAR *variables* **;**
WEIGHT *variable* **;**

The PROC CAPABILITY statement invokes the procedure. The VAR statement specifies the numeric variables to be analyzed, and it is required if the OUTPUT statement is used to save summary statistics and capability indices in an output data set. If you do not use the VAR statement, all numeric variables in the data set are analyzed. The SPEC statement provides specification limits.

The plot statements (CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, and QQPLOT) create graphical displays, and the INSET statement enhances these displays by adding a table of summary statistics directly on the graph. The INTERVALS statement computes statistical intervals. You can specify one or more of each of the plot statements, the INSET statement, the INTERVALS statement, and the OUTPUT statement. If you use a VAR statement, the variables listed in a plot statement must be a subset of the variables listed in the VAR statement.

PROC CAPABILITY Statement

The syntax for the PROC CAPABILITY statement is as follows:

PROC CAPABILITY < *options* > **;**

The following section lists all *options*. See the section "Dictionary of Options" on page 204 for detailed information.

Summary of Options

Table 6.2 lists all the PROC CAPABILITY *options* by function.

Table 6.2 PROC CAPABILITY Statement Options

Option	Description
Input Data Set Options	
ANNOTATE=	specifies input data set containing annotation information
DATA=	specifies input data set
EXCLNPWGT	specifies that non-positive weights are to be excluded
NOBYSPECS	specifies that specification limits in SPEC= data set are to be applied to all BY groups
SPEC=	specifies input data set with specification limits
Plotting and Graphics Options	
FORMCHAR(*index*)=	defines characters used for features on legacy line printer plots
GOUT=	specifies catalog for saving traditional graphics output
LINEPRINTER	requests legacy line printer plots
Computational Options	
FORCEQN	forces calculation of the robust estimator of scale Q_n
FORCESN	forces calculation of the robust estimator of scale S_n
PCTLDEF=	specifies definition used to calculate percentiles
ROUND=	specifies units used to round variable values
VARDEF=	specifies divisor used to calculate variances and standard deviations
Data Summary Options	
ALL	requests all tables
FREQ	requests frequency table
MODES	requests table of modes
NEXTROBS=	requests table of n lowest, n highest observations
NEXTRVAL=	requests table of n lowest, n highest values
Output Options	
NOPRINT	suppresses printed output
OUTTABLE=	creates an output data set containing univariate statistics and capability indices in tabular form
Hypothesis Testing Options	
MU0=	specifies mean for null hypothesis in tests for location
LOCCOUNT	requests table of counts used in sign test and signed rank test
NORMALTEST	performs tests for normality
Robust Estimation Options	
ROBUSTSCALE	requests table of robust measures of scale
TRIMMED=(*trimmed-options*)	requests table of trimmed means
WINSORIZED=(*Winsorized-options*)	requests table of Winsorized means
TRIMMED-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit

Table 6.2 *continued*

Option	Description
WINSORIZED-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit
Capability Index Options	
CPMA=	(obsolete) specifies a for Cpm(a)
CHECKINDICES	requests test of normality in conjunction with standard indices
SPECIALINDICES	requests table of specialized indices including Boyles' C_{pm}, C_{jkp}, C_{pmk}, $C_{pm}(a)$, and Wright's C_s
CHECKINDICES-Options	
ALPHA=	specifies cutoff probability for p-values for test for normality used in conjunction with process capability indices
TEST=	specifies test for normality (Shapiro-Wilk, Kolmogorov-Smirnov, Anderson-Darling, Cramér-von Mises, or no test)
Confidence Limit Options	
ALPHA=	specifies level for all confidence limits
CIBASIC	requests confidence limits for the mean, standard deviation, variance
CIINDICES	specifies level and type of confidence limits for capability indices
CIPCTLDF	requests distribution-free confidence limits for percentiles
CIPCTLNORMAL	requests confidence limits for percentiles assuming normality
CIPROBEX	requests confidence limits for the probability of exceeding specifications
CIBASIC-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit
CIINDICES-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit
CIPCTLDF-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit
CIPCTLNORMAL-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit
CIPROBEX-Options	
ALPHA=	specifies confidence level
TYPE=	specifies type of confidence limit

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the PROC CAPABILITY statement.

ALL

requests all of the tables generated by the FREQ, MODES, NEXTRVAL=5, CIBASIC, CIPCTLDF, and CIPCTLNORMAL options. If a WEIGHT statement is not used, the ALL option also requests the tables generated by the LOCCOUNT, NORMALTEST, ROBUSTSCALE, TRIMMED=.25, and WINSORIZED=.25 options. PROC CAPABILITY uses any values that you specify with the ALPHA=, MUO=, NEXTRVAL=, CIBASIC, CIPCTLDF, CIPCTLNORMAL, TRIMMED=, or WINSORIZED= options in conjunction with the ALL option.

ALPHA=*value*

specifies the default confidence level for all confidence limits computed by the CAPABILITY procedure. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 results in 90% confidence limits. The default *value* is 0.05.

Note that specialized ALPHA= options are available for a number of confidence interval options. For example, you can specify CIBASIC(ALPHA=0.10) to request a table of *Basic Confidence Limits* at the 90% level. The default values of these options default to the value of the general ALPHA= option.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set containing annotate variables as described in SAS/GRAPH documentation. You can use this data set to add features to traditional graphics. Use this data set only when creating traditional graphics; it is ignored when the LINEPRINTER option is specified and when ODS Graphics is in effect. Features provided in this data set are added to every plot produced in the current run of the procedure.

CHECKINDICES< (TEST = SW | KS | AD | CVM | NONE > < ALPHA=*value* **>)**

specifies the test of normality used in conjunction with process capability indices that are displayed in the *Process Capability Indices* table. If the p-value for the test is less than the cutoff probability value specified with the ALPHA= option, a warning is added to the table, as illustrated in Figure 6.3. See "Tests for Normality" on page 227 for details concerning the test.

```
proc capability data=Process;
var p2;
specs lsl=10
      usl=275;
run;
```

Figure 6.3 Warning Message Printed with Capability Indices

Process Capability Analysis of Fluid Weight

The CAPABILITY Procedure
Variable: P2

Process Capability Indices			
Index	**Value**	**95% Confidence Limits**	
Cp	0.541072	0.388938	0.692946
CPL	0.642426	0.417087	0.862984
CPU	0.439718	0.257339	0.617184
Cpk	0.439718	0.259310	0.620126

Warning: Normality is rejected for alpha = 0.05 using the Shapiro-Wilk test

ALPHA=*value*

specifies the cutoff probability for p-values for a test for normality used in conjunction with process capability indices. The *value* must be between zero and 0.5. The default value is 0.05.

TEST = SW | KS | AD | CVM | NONE

specifies the test of normality used in conjunction with process capability indices that are displayed in the *Process Capability Indices* table. The tests available are Shapiro-Wilk (SW), Kolmogorov-Smirnov (KS), Anderson-Darling (AD), and Cramér-von Mises (CVM). The default test is the Shapiro-Wilk test if the sample size is less than or equal to 2000 and the Kolmogorov-Smirnov test if the sample size is greater than 2000.

CIBASIC< (< TYPE=*keyword* **> < ALPHA=***value* **>) >**

requests confidence limits for the mean, standard deviation, and variance based on the assumption that the data are normally distributed. With large sample sizes, this assumption is not required for confidence limits for the mean.

ALPHA=*value*

specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

CIINDICES< (TYPE=*keyword* **>< ALPHA=***value* **>)**

specifies the type and level of the confidence limits for standard capability indices displayed in the table labeled *Process Capability Indices*.

ALPHA=*value*

specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

CIPCTLDF< (TYPE=*keyword* **>< ALPHA=***value* **>)**

CIQUANTDF< (TYPE=*keyword* **>< ALPHA=***value* **>)**

requests confidence limits for quantiles computed using a distribution-free method. In other words, no specific parametric distribution (such as the normal) is assumed for the data. Order statistics are used to compute the confidence limits as described in Section 5.2 of Hahn and Meeker (1991). This option is not available if you specify a WEIGHT statement.

ALPHA=*value*

specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, SYMMETRIC, or ASYMMETRIC. The default value is SYMMETRIC.

CIPCTLNORMAL< (TYPE=*keyword* **>< ALPHA=***value* **>)**

CIQUANTNORMAL< (TYPE=*keyword* **> < ALPHA=***value* **>)**

requests confidence limits for quantiles based on the assumption that the data are normally distributed. The computational method is described in Section 4.4.1 of Hahn and Meeker (1991) and uses the noncentral *t* distribution as given by Odeh and Owen (1980). This option is not available if you specify a WEIGHT statement.

ALPHA=*value*

specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

CIPROBEX< (TYPE=*keyword* **>< ALPHA=***value* **>)**

requests confidence limits for $Pr[X \leq \text{LSL}]$ and $Pr[X \geq \text{USL}]$, where X is the analysis variable, LSL is the lower specification limit, and USL is the upper specification limit. The computational method, which assumes that X is normally distributed, is described in Section 4.5 of Hahn and Meeker (1991) and uses the noncentral *t* distribution as given by Odeh and Owen (1980). This option is not available if you specify a WEIGHT statement.

ALPHA=*value*

specifies the confidence level. The coverage percent for the confidence limits is $(1 - value)100$. For example, ALPHA=0.10 requests 90% confidence limits. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

CPMA=*value*

specifies the *value* of the parameter a for the capability index $C_{pm}(a)$. This option has been superseded by the SPECIALINDICES(CPMA=) option.

DATA=*SAS-data-set*
specifies the input data set containing the observations to be analyzed. If the DATA= option is omitted, the procedure uses the most recently created SAS data set.

DEF=*index*
is an alias for the PCTLDEF= option. See the entry for the PCTLDEF= option.

EXCLNPWGT
excludes observations with non-positive weight values (zero or nonnegative) for the analysis. By default, PROC CAPABILITY treats observations with negative weights like those with zero weights and counts them in the total number of observations. This option is applicable only if you specify a WEIGHT statement.

FORCEQN
forces calculation of the robust estimate of scale Q_n. Because this calculation is very computationally intensive, by default Q_n is not computed for a variable that has more than 65,526 nonmissing observations. On some hosts, Q_n cannot be computed at all when there are more than 65,526 nonmissing observations.

FORCESN
forces calculation of the robust estimate of scale S_n. Because this calculation is computationally intensive, by default S_n is not computed for a variable that has more than 1 million nonmissing observations.

FORMCHAR(*index***)='***string***'**
defines characters used for features on legacy line printer plots, where *index* is a number ranging from 1 to 11, and *string* is a character or hexadecimal string. This option is ignored unless you specify the LINEPRINTER option in the PROC CAPABILITY statement.

The *index* identifies which features are controlled with the *string* characters, as discussed in the table that follows. If you specify the FORMCHAR= option omitting the *index*, the *string* controls all 11 features.

By default, the form character list specified with the SAS system option FORMCHAR= is used; otherwise, the default is FORMCHAR='|---|+|--'. If you print to a PC screen or your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9'X
```

As an example, suppose you want to plot the data values of the empirical cumulative distribution function with asterisks (*). You can change the appropriate character by using the following:

```
formchar(2)='*'
```

Note that the FORMCHAR= option in the PROC CAPABILITY statement enables you to temporarily override the values of the SAS system option with the same name. The values of the SAS system option are not altered by using the FORMCHAR= option in PROC CAPABILITY statement.

The features associated with values of *index* are as follows:

Value of *index*	Description of Character	Chart Feature
1	vertical bar	frame, ecdf line, HREF= lines
2	horizontal bar	frame, ecdf line, VREF= lines
3	box character (upper left)	frame, ecdf line, histogram bars
4	box character (upper middle)	histogram bars, tick marks (horizontal axis)
5	box character (upper right)	frame, histogram bars
6	box character (middle left)	histogram bars
7	box character (middle middle)	not used
8	box character (middle right)	histogram bars, tick marks (vertical axis)
9	box character (lower left)	frame
10	box character (lower middle)	histogram bars
11	box character (lower right)	frame, ecdf line

FREQ

requests a frequency table in the printed output that contains the variable values, frequencies, percentages, and cumulative percentages. See Figure 6.2 for an example.

GOUT=*graphics-catalog*

specifies a graphics catalog in which to save traditional graphics output. This option is ignored unless you are producing traditional graphics.

LINEPRINTER

requests that legacy line printer plots be produced by the CDFPLOT, HISTOGRAM, PROBPLOT, PPPLOT, and QQPLOT statements. The CLASS and COMPHISTOGRAM statements cannot be used when the LINEPRINTER option is specified.

LOCCOUNT

requests a table with the number of observations greater than, not equal to, and less than the value of MUO=. PROC CAPABILITY uses these values to construct the sign test and signed rank test. This option is not available if you specify a WEIGHT statement.

MODES

MODE

requests a table of all possible modes. By default, when the data contains multiple modes, PROC CAPABILITY displays the lowest mode in the table of basic statistical measures. When all values are unique, PROC CAPABILITY does not produce a table of modes.

MU0=*value(s)*

LOCATION=*value(s)*

specifies the value of the mean or location parameter (μ_o) in the null hypothesis for the tests summarized in the table labeled *Tests for Location: Mu0=value*. If you specify a single value, PROC CAPABILITY tests the same null hypothesis for all analysis variables. If you specify multiple values, a VAR statement is required, and PROC CAPABILITY tests a different null hypothesis for each analysis variable by matching the VAR variables with the values in the corresponding order. The default value is 0.

NEXTROBS=*n*

specifies the number of extreme observations in the table labeled *Extreme Observations*. The table lists the *n* lowest observations and the *n* highest observations. The default value is 5. The value of *n* must be an integer between 0 and half the number of observations. You can specify NEXTROBS=0 to suppress the table.

NEXTRVAL=*n*

requests the table labeled *Extreme Values* and specifies the number of extreme values in the table. The table lists the *n* lowest unique values and the *n* highest unique values. The value of *n* must be an integer between 0 and half the maximum number of observations. By default, $n = 0$ and no table is displayed.

NOBYSPECS

specifies that specification limits in SPEC= data set be applied to all BY groups. If you use a BY statement and specify a SPEC= data set that does not contain the BY variables, you must specify the NOBYSPECS option.

NOPRINT

suppresses the tables of descriptive statistics and capability indices which are created by the PROC CAPABILITY statement. The NOPRINT option does not suppress the tables created by the INTERVALS or plot statements. You can use the NOPRINT options in these statements to suppress the creation of their tables.

NORMALTEST

NORMAL

requests a table of *Tests for Normality* for each of the analysis variables. The table provides test statistics and *p*-values for the Shapiro-Wilk test (provided the sample size is less than or equal to 2000), the Kolmogorov-Smirnov test, the Anderson-Darling test, and the Cramér-von Mises test. See "Tests for Normality" on page 227 for details. If specification limits are provided, the NORMALTEST option is assumed.

OUTTABLE=*SAS-data-set*

specifies an output data set that contains univariate statistics and capability indices arranged in tabular form. See "OUTTABLE= Data Set" on page 222 for details.

PCTLDEF=*index*

DEF=*index*

specifies one of five definitions used to calculate percentiles. The value of *index* can be 1, 2, 3, 4, or 5. See "Percentile Computations" on page 230 for details. By default, PCTLDEF=5.

ROBUSTSCALE

requests a table of robust measures of scale. These measures include the interquartile range, Gini's mean difference, the median absolute deviation about the median (*MAD*), and two statistics proposed by Rousseeuw and Croux (1993), Q_n, and S_n. This option is not available if you specify a WEIGHT statement.

ROUND=*value-list*

specifies units used to round variable values. The ROUND= option reduces the number of unique values for each variable and hence reduces the memory required for temporary storage. *Values* must be greater than 0 for rounding to occur.

If you use only one value, the procedure uses this unit for all variables. If you use a list of values, you must also use a VAR statement. The procedure then uses the roundoff values for variables in the order given in the VAR statement. For example, the following statements specify a roundoff value of 1 for Yieldstrength and a roundoff value of 0.5 for TENSTREN.

```
proc capability round=1 0.5;
   var Yieldstrength tenstren;
run;
```

When a variable value is midway between the two nearest rounded points, the value is rounded to the nearest even multiple of the roundoff value. For example, with a roundoff value of 1, the variable values of –2.5, –2.2, and –1.5 are rounded to –2; the values of –0.5, 0.2, and 0.5 are rounded to 0; and the values of 0.6, 1.2, and 1.4 are rounded to 1.

SPECIALINDICES

requests a table of specialized process capability indices. These indices include k, Boyles' modified C_{pm} (also denoted as $C_{pm}+$), C_{jkp}, $C_{pm}(a)$, $C_p(5.15)$, $C_{pk}(5.15)$, C_{pmk}, Wright's C_s, Boyles' S_{jkp}, C_{pp}, C''_{pp}, C_{pg}, C_{pq}, C_p^W, C_{pk}^W, C_{pm}^W, C_{pc}, and Vännmann's $C_p(u,v)$ and $C_p(v)$.

You can provide values for the parameters a for $C_{pm}(a)$, u and v for $C_p(u,v)$ and $C_p(v)$, and for the γ multiplier for C_s by specifying the following options in parentheses after the SPECIALINDICES option.

CPMA=*value*

specifies the *value* of the parameter a for the capability index $C_{pm}(a)$ described in Section 3.7 of Kotz and Johnson (1993). The *value* must be positive. The default *value* is 0.5. The existing CPMA= option in the PROC CAPABILITY statement is considered obsolete but still works.

CPU=*value*

specifies the *value* of the parameter u for Vännmann's capability index $C_p(u,v)$. The *value* must be greater than or equal to zero. The default *value* is zero.

CPV=*value*

specifies the *value* of the parameter v for Vännmann's capability indices $C_p(u,v)$ and $C_p(v)$. The *value* must be greater than or equal to zero. The default *value* is 4.

CSGAMMA=*value*

specifies the *value* of the γ multiplier suggested by Chen and Kotz (1996) for Wright's capability index C_s. The *value* must be greater than zero. The default *value* is 1.

SPEC=*SAS-data-set*

SPECS=*SAS-data-set*

specifies an input data set containing specification limits for each of the variables in the VAR statement. This option is an alternative to the SPEC statement, which also provides specification limits. See "SPEC= Data Set" on page 220 for details on SPEC= data sets, and Example 6.1 for an example. If you use both the SPEC= option and a SPEC statement, the SPEC= option is ignored.

TRIMMED=*values(s)* **< (TYPE=***keyword* **>< ALPHA=***value* **>)**

requests a table of trimmed means, where each *value* specifies the number or the proportion of trimmed observations. If the *value* is the number n of trimmed observations, n must be between 0 and half the number of nonmissing observations. If the *value* is a proportion p between 0 and 0.5, the number of observations trimmed is the smallest integer greater than or equal to np, where n is the number of observations. To obtain confidence limits for the mean and the student t-test, you must use the default value of VARDEF= which is DF. The TRIMMED= option is not available if you specify a WEIGHT statement.

ALPHA=*value*

specifies the confidence level. The coverage percent is $(1 - value)100$. For example, ALPHA=0.10 requests a 90% confidence limit. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor used in calculating variances and standard deviations. The values and associated divisors are shown in the following table. By default, VARDEF=DF.

Value	Divisor	Formula
DF	degrees of freedom	$n - 1$
N	number of observations	n
WEIGHT | WGT	sum of weight	$\sum_i w_i$
WDF	sum of weights minus one	$(\sum_i w_i) - 1$

WINSORIZED=*values(s)* **< (< TYPE=***keyword* **> < ALPHA=***value* **>) >**

WINSOR=*values(s)* **< (< TYPE=***keyword* **> < ALPHA=***value* **>) >**

requests a table of winsorized means, where each *value* specifies the number or the proportion of winsorized observations. If the *value* is the number n of winsorized observations, n must be between 0 and half the number of nonmissing observations. If the *value* is a proportion p between 0 and 0.5, the number of observations winsorized is the smallest integer greater than or equal to np, where n is the number of observations. To obtain confidence limits for the mean and the student t-test, you must use the default value of VARDEF= which is DF. The WINSORIZED= option is not available if you specify a WEIGHT statement.

ALPHA=*value*

specifies the confidence level. The coverage percent is $(1 - value)100$. For example, ALPHA=0.10 results in a 90% confidence limit. The default value is 0.05.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default value is TWOSIDED.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CAPABILITY to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the CAPABILITY procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable-1* < **(***v-options***)** > < *variable-2* < **(***v-options***)** > >
< / **KEYLEVEL=** *value1* | **(***value1 value2***)** > **;**

The CLASS statement specifies one or two variables used to group the data into classification levels. Variables in a CLASS statement are referred to as *CLASS variables*. CLASS variables can be numeric or character. Class variables can have floating point values, but they typically have a few discrete values that define levels of the variable. You do not have to sort the data by CLASS variables. PROC CAPABILITY uses the formatted values of the CLASS variables to determine the classification levels.

NOTE: You cannot specify a COMPHISTOGRAM statement together with a CLASS statement.

You can specify the following *v-options* enclosed in parentheses after a CLASS variable:

MISSING
specifies that missing values for the CLASS variable are to be treated as valid classification levels. Special missing values that represent numeric values (‘.A’ through ‘.Z’ and ‘._’) are each considered as a separate value. If you omit MISSING, PROC CAPABILITY excludes the observations with a missing CLASS variable value from the analysis. Enclose this option in parentheses after the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the display order for the CLASS variable values. The default value is INTERNAL. You can specify the following values with the ORDER= option:

DATA
orders values according to their order in the input data set. When you use a plot statement, PROC CAPABILITY displays the rows (columns) of the comparative plot from top to bottom (left to right) in the order that the CLASS variable values first appear in the input data set.

FORMATTED orders values by their ascending formatted values. This order might depend on your operating environment. When you use a plot statement, PROC CAPABILITY displays the rows (columns) of the comparative plot from top to bottom (left to right) in increasing order of the formatted CLASS variable values. For example, suppose a numeric CLASS variable DAY (with values 1, 2, and 3) has a user-defined format that assigns Wednesday to the value 1, Thursday to the value 2, and Friday to the value 3. The rows of the comparative plot will appear in alphabetical order (Friday, Thursday, Wednesday) from top to bottom.

If there are two or more distinct internal values with the same formatted value, then PROC CAPABILITY determines the order by the internal value that occurs first in the input data set. For numeric variables without an explicit format, the levels are ordered by their internal values.

FREQ orders values by descending frequency count so that levels with the most observations are listed first. If two or more values have the same frequency count, PROC CAPABILITY uses the formatted values to determine the order.

When you use a plot statement, PROC CAPABILITY displays the rows (columns) of the comparative plot from top to bottom (left to right) in order of decreasing frequency count for the CLASS variable values.

INTERNAL orders values by their unformatted values, which yields the same order as PROC SORT. This order may depend on your operating environment.

When you use a plot statement, PROC CAPABILITY displays the rows (columns) of the comparative plot from top to bottom (left to right) in increasing order of the internal (unformatted) values of the CLASS variable. The first CLASS variable is used to label the rows of the comparative plots (top to bottom). The second CLASS variable is used to label the columns of the comparative plots (left to right). For example, suppose a numeric CLASS variable DAY (with values 1, 2, and 3) has a user-defined format that assigns Wednesday to the value 1, Thursday to the value 2, and Friday to the value 3. The rows of the comparative plot will appear in day-of-the-week order (Wednesday, Thursday, Friday) from top to bottom.

You can specify the following options after the slash (/) in the CLASS statement.

KEYLEVEL=*value* | (*value1 value2*)

specifies the *key cells* in comparative plots. For each plot, PROC CAPABILITY first determines the horizontal axis scaling for the key cell, and then extends the axis using the established tick interval to accommodate the data ranges for the remaining cells, if necessary. Thus, the choice of the key cell determines the uniform horizontal axis that PROC CAPABILITY uses for all cells.

If you specify only one CLASS variable and use a plot statement, KEYLEVEL=*value* identifies the key cell as the level for which the CLASS variable is equal to *value*. By default, PROC CAPABILITY sorts the levels in the order determined by the ORDER= option, and the key cell is the first occurrence of a level in this order. The cells display in order from top to bottom or left to right. Consequently, the key cell appears at the top (or left). When you specify a different key cell with the KEYLEVEL= option, this cell appears at the top (or left).

If you specify two CLASS variables, use KEYLEVEL= (*value1 value2)* to identify the key cell as the level for which CLASS variable *n* is equal to *valuen*. By default, PROC CAPABILITY sorts the levels of the first CLASS variable in the order that is determined by its ORDER= option. Then, within each

of these levels, it sorts the levels of the second CLASS variable in the order that is determined by its ORDER= option. The default key cell is the first occurrence of a combination of levels for the two variables in this order. The cells display in the order of the first CLASS variable from top to bottom and in the order of the second CLASS variable from left to right. Consequently, the default key cell appears at the upper left corner. When you specify a different key cell with the KEYLEVEL= option, this cell appears at the upper left corner.

The length of the KEYLEVEL= value cannot exceed 16 characters and you must specify a formatted value.

The KEYLEVEL= option has no effect unless you specify a plot statement.

NOKEYMOVE

specifies that the location of the key cell in a comparative plot be unchanged by the CLASS statement KEYLEVEL= option. By default, the key cell is positioned as the first cell in a comparative plot.

The NOKEYMOVE option has no effect unless you specify a plot statement.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the input data set. If *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times. If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

ID Statement

ID *variables* **;**

The ID statement specifies one or more variables to include in the table of extreme observations. The corresponding values of the ID variables appear beside the *n* largest and *n* smallest observations, where *n* is the value of the NEXTROBS= option.

SPEC Statement

The syntax for the SPEC statement is as follows:

SPEC < *options* > **;**

You can use at most one SPEC statement in the CAPABILITY procedure. When you provide specification limits and target values in a SPEC statement, the tabular output produced by the PROC CAPABILITY statement includes process capability indices as well as summary statistics. You can use the SPEC statement in conjunction with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements to add specification limit and target lines to the plots produced with these statements.

options

control features of the specification limits and target values. Table 6.3 lists all options by function.

Summary of Options

Table 6.3 SPEC Statement Options

Option	Description
Lower Specification Limit Options	
CLEFT=	color used to fill area left of lower specification limit (histograms only)
CLSL=	color of lower specification limit line
LLSL=	line type of lower specification limit line
LSL=	lower specification limit values
LSLSYMBOL=	character used for lower specification limit line in line printer plots
PLEFT=	pattern type used to fill area left of lower specification limit (histograms only)
WLSL=	width of lower specification limit line
Target Options	
CTARGET=	color of target line
LTARGET=	line type of target line
TARGET=	target value
TARGETSYMBOL=	character used for target in line printer plots
WTARGET=	width of target line
Upper Specification Limit Options	
CRIGHT=	color used to fill area right of upper specification limit (histograms only)
CUSL=	color of upper specification limit line
LUSL=	line type of upper specification limit line
PRIGHT=	pattern type used to fill area right of upper specification limit (histograms only)
USL=	upper specification limit values
USLSYMBOL=	character used for upper specification limit in line printer plots
WUSL=	width of upper specification limit line

General Options

You can specify the following options whether you are producing ODS Graphics output or traditional graphics:

CLEFT=*color*

CLEFT

determines the *color* used to fill the area under a histogram to the left of the lower specification limit. You can specify the CLEFT option without an argument to fill this area with an appropriate color from the ODS style. If you are producing ODS Graphics output, an explicit color specification is ignored. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. See Output 6.2.1 for an example. The CLEFT= option also applies to the area under a fitted curve;for an example, see Output 6.8.1.

CRIGHT=*color*

CRIGHT

determines the *color* used to fill the area under a histogram to the right of the upper specification limit. You can specify the CRIGHT option without an argument to fill this area with an appropriate color from the ODS style. If you are producing ODS Graphics output, an explicit color specification is ignored. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. See Output 6.2.1 for an example. The CRIGHT= option also applies to the area under a fitted curve; for an example, see Output 6.8.1.

LSL=*value-list*

specifies the lower specification limits for the variables listed in the VAR statement, or for all numeric variables in the input data set if no VAR statement is used. If you specify only one lower limit, it is used for all of the variables; otherwise, the number of limits must match the number of variables. See the section "Computing Capability Indices" on page 199 for an example.

TARGET=*value-list*

specifies target values for the variables listed in the VAR statement, or for all numeric variables in the input data set if no VAR statement is used. If you specify only one target value, it is used for all of the variables; otherwise, the number of values must match the number of variables. See the section "Computing Capability Indices" on page 199 for an example.

USL=*value-list*

specifies the upper specification limits for the variables listed in the VAR statement, or for all numeric variables in the input data set if no VAR statement is used. If you specify only one upper limit, it is used for all of the variables; otherwise, the number of limits must match the number of variables. See the section "Computing Capability Indices" on page 199 for an example.

Options for Traditional Graphics

You can specify the following options if you are producing traditional graphics:

CLSL=*color*

specifies the color of the lower specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements.

CTARGET=*color*

specifies the color of the target line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements.

CUSL=*color*

specifies the color of the upper specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements.

LLSL=*linetype*

specifies the line type for the lower specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 6.2.1 for an example. The default is 1, which produces a solid line.

LTARGET=*linetype*
specifies the line type for the target line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 6.2.1 for an example. The default is 1, which produces a solid line.

LUSL=*linetype*
specifies the line type for the upper specification line displayed in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 6.2.1 for an example. The default is 1, which produces a solid line.

PLEFT=*pattern*
specifies the pattern used to fill the area under a histogram to the left of the lower specification limit. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. For an example, see Output 6.2.1. The PLEFT= option also applies to the area under a fitted curve; for an example, see Output 6.8.1. The default pattern is a solid fill.

PRIGHT=*pattern*
specifies the pattern used to fill the area under a histogram to the right of the upper specification limit. This option is applicable only when the SPEC statement is used in conjunction with a HISTOGRAM or COMPHISTOGRAM statement. For an example, see Output 6.2.1. The PRIGHT= option also applies to the area under a fitted curve; for an example, see Output 6.8.1. The default pattern is a solid fill.

WLSL=*n*
specifies the width in pixels of the lower specification line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 6.2.1 for an illustration. The default is 1.

WTARGET=*n*
specifies the width in pixels of the target line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 6.2.1 for an illustration. The default is 1.

WUSL=*n*
specifies the width in pixels of the upper specification line in plots created with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PROBPLOT, and QQPLOT statements. See Output 6.2.1 for an illustration. The default is 1.

Options for Legacy Line Printer Plots

You can specify the following options if you are producing legacy line printer plots:

LSLSYMBOL='*character*'
specifies the character used to display the lower specification line in line printer plots created with the CDFPLOT, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default character is 'L'.

TARGETSYMBOL='*character*'
TARGETSYM='*character*'
specifies the character used to display the target line in line printer plots created with the CDFPLOT, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default character is 'T'.

USLSYMBOL='*character*'
specifies the character used to display the upper specification line in line printer plots created with the CDFPLOT, HISTOGRAM, PROBPLOT, and QQPLOT statements. The default character is 'U'.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the analysis variables and their order in the results. By default, if you omit the VAR statement, PROC CAPABILITY analyzes all numeric variables that are not listed in the other statements.

You must provide a VAR statement when you use an OUTPUT statement. To store the same statistic for several analysis variables in the OUT= data set, you specify a list of names in the OUTPUT statement. PROC CAPABILITY makes a one-to-one correspondence between the order of the analysis variables in the VAR statement and the list of names that follow a statistic keyword.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names a variable that provides weights for each observation in the input data set. The CAPABILITY procedure uses the values w_i of the WEIGHT variable to modify the computation of a number of summary statistics by assuming that the variance of the ith value X_i of the analysis variable is equal to σ^2/w_i, where σ is an unknown parameter. This assumption is rarely applicable in process capability analysis, and the purpose of the WEIGHT statement is simply to make the CAPABILITY procedure consistent with other data summarization procedures, such as the UNIVARIATE procedure.

The values of the WEIGHT variable do not have to be integers and are typically positive. By default, observations with non-positive or missing values of the WEIGHT variable are handled as follows:

- If the value is zero, the observation is counted in the total number of observations.
- If the value is negative, it is converted to zero, and the observation is counted in the total number of observations.
- If the value is missing, the observation is excluded from the analysis.

To exclude observations that contain negative and zero weights from the analysis, specify the option EXCLNPWGT in the PROC statement. Note that most SAS/STAT® procedures, such as PROC GLM, exclude negative and zero weights by default.

When you specify a WEIGHT variable, the procedure uses its values, w_i, to compute weighted versions of the statistics provided in the *Moments* table. For example, the procedure computes a weighted mean $\overline{X}_w$ and a weighted variance s_w^2 as $\overline{X}_w = \frac{\sum_i w_i x_i}{\sum_i w_i}$ and $s_w^2 = \frac{1}{d}\sum_i w_i (x_i - \overline{X}_w)^2$ where x_i is the ith variable value. The divisor d is controlled by the VARDEF= option in the PROC CAPABILITY statement.

When you use both the WEIGHT and SPEC statements, capability indices are computed using $\overline{X}_w$ and s_w in place of $\overline{X}$ and s. Again, note that weighted capability indices are seldom needed in practice.

When you specify a WEIGHT statement, the procedure also computes a weighted standard error and a weighted version of Student's t test. This test is the only test of location that is provided when weights are specified.

The WEIGHT statement does not affect the determination of the mode, extreme values, extreme observations, or the number of missing values of the analysis variables. However, the weights w_i are used to compute weighted percentiles.

The WEIGHT variable has no effect on the calculation of extreme values, and it has no effect on graphical displays produced with the plot statements.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance printed output. If you are creating traditional graphics, you can also use AXIS, LEGEND, PATTERN, and SYMBOL statements to enhance your plots. For details, see SAS/GRAPH documentation and the chapter for the plot statement that you are using.

Details: CAPABILITY Procedure

This section provides details on the following topics:

- input data sets specified with the DATA= option, the SPEC= option, and the ANNOTATE= option
- the output data set specified with the OUTTABLE= option
- descriptive statistics
- the tests for normality requested with the NORMALTEST option
- percentile definitions controlled using the PCTLDEF= option
- robust estimators
- computing the mode
- assumptions and terminology for capability indices
- standard capability indices
- specialized capability indices

Input Data Sets

DATA= Data Set

The DATA= data set contains a set of variables that represent measurements from a process. The CAPABILITY procedure must have a DATA= data set. If you do not specify one with the DATA= option in the PROC CAPABILITY statement, the procedure uses the last data set created.

SPEC= Data Set

The SPEC= option in the PROC CAPABILITY statement identifies a SPEC= data set, which contains specification limits. This option is an alternative to using the SPEC statement. If you use both the SPEC= option and a SPEC statement, the SPEC= option is ignored. The SPEC= option is especially useful when:

- the number of variables is large
- the same specification limits are referred to in more than one analysis
- a BY statement is used
- batch processing is used

The following variables are read from a SPEC= data set:

Variable	Description
LSL	lower specification limit
TARGET	target value
USL	upper specification limit
VAR	name of the variable

You may omit either _LSL_ or _USL_ but not both. _TARGET_ is optional. If the SPEC= data set contains both _LSL_ and _USL_, you can assign missing values to _LSL_ or _USL_ to indicate one-sided specifications. You can assign missing values to _TARGET_ when the variable does not use a target value. _LSL_, _USL_, and _TARGET_ must be numeric variables. _VAR_ must be a character variable.

You can include the following optional variables in a SPEC= data set to control the appearance of specification limits on charts:

Variable	Description
CLEFT	color used to fill area left of LSL (histograms only)
CLSL	color of LSL line
CRIGHT	color used to fill area right of USL (histograms only)
CTARGET	color of target line
CUSL	color of USL line
LLSL	line type of LSL line
LSLSYM	character used for LSL line in line printer plots
LTARGET	line type of target line
LUSL	line type of USL line
PLEFT	pattern type used to fill area left of LSL (histograms only)
PRIGHT	pattern type used to fill area right of USL (histograms only)
TARGETSYM	character used for target in line printer plots
USLSYM	character used for USL line in line printer plots
WLSL	width of LSL line
WTARGET	width of target line
WUSL	width of USL line

If you are using the HISTOGRAM statement to create "clickable" histograms in HTML, you can also provide the following variables in a SPEC= data set:

Variable	Description
LOURL	URL associated with area to left of lower specification limit
HIURL	URL associated with area to right of upper specification limit
URL	URL associated with area between specification limits

These are character variables whose values are Uniform Resource Locators (URLs) linked to areas on a histogram. When you view the ODS HTML output with a browser, you can click on an area, and the browser will bring up the page specified by the corresponding URL.

If you use a BY statement, the SPEC= data set must also contain the BY variables. The SPEC= data set must be sorted in the same order as the DATA= data set. Within a BY group, specification limits for each variable plotted are read from the first observation where _VAR_ matches the variable name.

See the section "Examples: CAPABILITY Procedure" on page 248 for an example of reading specification limits from a SPEC= data set.

ANNOTATE= Data Sets

In the CAPABILITY procedure, you can add features to traditional graphics plots by specifying ANNOTATE= data sets either in the PROC CAPABILITY statement or in individual plot statements. Depending on where you specify an ANNOTATE= data set, however, the information is used for all plots or only for plots produced by a given statement.

Information contained in the ANNOTATE= data set specified in the PROC CAPABILITY statement is used for all plots produced in a given PROC step; this is a "global" ANNOTATE= data set. By using this global data set, you can keep information common to all plots in one data set.

Information contained in the ANNOTATE= data set specified in a plot statement is used for plots produced by that statement; this is a "local" ANNOTATE= data set. By using this data set, you can add statement-specific features to plots. For example, you can add different features to plots produced by the HISTOGRAM and QQPLOT statements by specifying an ANNOTATE= data set in each plot statement.

In addition, you can specify an ANNOTATE= data set in the PROC CAPABILITY statement and in plot statements. This enables you to add some features to all plots (those given in the data set specified in the PROC statement) and also add statement-specific features to plots (those given in the data set specified in the plot statement).

For complete details on the structure and content of Annotate type data sets, see SAS/GRAPH documentation.

Output Data Set

OUTTABLE= Data Set

The OUTTABLE= data set saves univariate statistics and capability indices. The following variables can be saved:

Table 6.4 OUTTABLE= Data Set

Variable	Description
_CP_	Capability index C_p
CPLCL	Lower confidence limit for C_p
CPUCL	Upper confidence limit for C_p
CPK	Capability index C_{pk}
CPKLCL	Lower confidence limit for C_{pk}
CPKUCL	Upper confidence limit for C_{pk}
CPL	Capability index CPL
CPLLCL	Lower confidence limit for CPL
CPLUCL	Upper confidence limit for CPL
CPM	Capability index C_{pm}
CPMLCL	Lower confidence limit for C_{pm}
CPMUCL	Upper confidence limit for C_{pm}
CPU	Capability index CPU
CPULCL	Lower confidence limit for CPU
CPUUCL	Upper confidence limit for CPU
CSS	Corrected sum of squares
_CV_	Coefficient of variation
GEOMEAN	Geometric mean
GINI	Gini's mean difference
_K_	Capability index K
KURT	Kurtosis
LSL	Lower specification limit
MAD	Median absolute difference about the median
MAX	Maximum
MEAN	Mean
MEDIAN	Median
MIN	Minimum
MODE	Mode
MSIGN	Sign statistic
NMISS	Number of missing observations
NOBS	Number of nonmissing observations
_P1_	1st percentile
_P5_	5th percentile
_P10_	10th percentile
_P90_	90th percentile
_P95_	95th percentile
_P99_	99th percentile
PCTGTR	Percentage of observations greater than upper specification limit
PCTLSS	Percentage of observations less than lower specification limit

Table 6.4 *(continued)*

Variable	Description
PROBM	p-value of sign statistic
PROBN	p-value of test for normality
PROBS	p-value of signed rank test
PROBT	p-value of t statistic
_Q1_	25th percentile (lower quartile)
_Q3_	75th percentile (upper quartile)
_QN_	Q_n (see "Robust Estimates of Scale" on page 233)
QRANGE	Interquartile range (upper quartile minus lower quartile)
RANGE	Range
SGNRNK	Centered sign rank
SKEW	Skewness
_SN_	S_n (see "Robust Estimates of Scale" on page 233)
STD	Standard deviation
STDGINI	Gini's standard deviation
STDMAD	MAD standard deviation
STDMEAN	Standard error of the mean
STDQN	Q_n standard deviation
STDQRANGE	Interquartile range standard deviation
STDSN	S_n standard deviation
SUMWGT	Sum of the weights
SUM	Sum
TARGET	Target value
USL	Upper specification limit
USS	Uncorrected sum of squares
VARI	Variance
VAR	Variable name

NOTE: The variables _CP_, _CPLCL_, _CPUCL_, _CPK_, _CPKLCL_, _CPKUCL_, _CPL_, _CPLLCL_, _CPLUCL_, _CPM_, _CPMLCL_, _CPMUCL_, _CPU_, _CPULCL_, _CPUUCL_, _K_, _LSL_, _PCTGTR_, _PCTLSS_, _TARGET_, and _USL_ are included if you provide specification limits.

The OUTTABLE= data set and the OUT= data set[2] contain essentially the same information. However, the structure of the OUTTABLE= data set may be more appropriate when you are computing summary statistics or capability indices for more than one process variable in the same invocation of the CAPABILITY procedure. Each observation in the OUTTABLE= data set corresponds to a different process variable, and the variables in the data set correspond to summary statistics and indices.

NOTE: See *Tabulating Results for Multiple Variables* in the SAS/QC Sample Library.

For example, suppose you have ten process variables (P1-P10). The following statements create an OUTTABLE= data set named Table, which contains summary statistics and capability indices for each of these variables:

[2] See "OUTPUT Statement: CAPABILITY Procedure" on page 424 for details on the OUT= data set.

```
proc capability data=Process outtable=Table noprint;
   var P1-P10;
   specs lsl=5 10 65 35 35 5 25 25 60 15
         usl=175 275 300 450 550 200 275 425 500 525;
run;
```

The following statements create the table shown in Figure 6.4, which contains the mean, standard deviation, lower and upper specification limits, and capability index C_{pk} for each process variable:

```
proc print data=Table label noobs;
   var _VAR_ _MEAN_ _STD_ _LSL_ _USL_ _CPK_;
   label _VAR_='Process';
run;
```

Figure 6.4 Tabulating Results for Multiple Process Variables

Process Capability Analysis of Fluid Weight

Process	Mean	Standard Deviation	Lower Specification Limit	Upper Specification Limit	Capability Index CPK
P1	90.76	57.024	5	175	0.49242
P2	167.32	81.628	10	275	0.43972
P3	224.56	96.525	65	300	0.26052
P4	258.08	145.218	35	450	0.44053
P5	283.48	157.033	35	550	0.52745
P6	107.48	52.437	5	200	0.58814
P7	153.20	90.031	25	275	0.45096
P8	217.08	130.031	25	425	0.49239
P9	280.68	140.943	60	500	0.51870
P10	243.24	178.799	15	525	0.42551

Descriptive Statistics

This section provides computational details for the descriptive statistics which are computed with the PROC CAPABILITY statement. These statistics can also be saved in the OUT= data set by specifying the keywords listed in Table 6.52 in the OUTPUT statement.

Standard algorithms (Fisher 1973) are used to compute the moment statistics. The computational methods used by the CAPABILITY procedure are consistent with those used by other SAS procedures for calculating descriptive statistics. For details on statistics also calculated by Base SAS software, see *Base SAS Procedures Guide*.

The following sections give specific details on several statistics calculated by the CAPABILITY procedure.

Mean

The sample mean is calculated as

$$\frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

where n is the number of nonmissing values for a variable, x_i is the ith value of the variable, and w_i is the weight associated with the ith value of the variable. If there is no WEIGHT= variable, the formula reduces to $\frac{1}{n}\sum_{i=1}^{n} x_i$.

Sum

The sum is calculated as $\sum_{i=1}^{n} w_i x_i$, where n is the number of nonmissing values for a variable, x_i is the ith value of the variable, and w_i is the weight associated with the ith value of the variable. If there is no WEIGHT= variable, the formula reduces to $\sum_{i=1}^{n} x_i$.

Sum of the Weights

The sum of the weights is calculated as $\sum_{i=1}^{n} w_i$, where n is the number of nonmissing values for a variable and w_i is the weight associated with the ith value of the variable. If there is no WEIGHT= variable, the sum of the weights is n.

Variance

The variance is calculated as

$$\frac{1}{d}\sum_{i=1}^{n} w_i (x_i - \bar{X}_w)^2$$

where n is the number of nonmissing values for a variable, x_i is the ith value of the variable, $\bar{X}_w$ is the weighted mean, w_i is the weight associated with the ith value of the variable, and d is the divisor controlled by the VARDEF= option in the PROC CAPABILITY statement. If there is no WEIGHT= variable, the formula reduces to

$$\frac{1}{d}\sum_{i=1}^{n} (x_i - \bar{X}_w)^2$$

Standard Deviation

The standard deviation is calculated as

$$\sqrt{\frac{1}{d}\sum_{i=1}^{n} w_i (x_i - \bar{X}_w)^2}$$

where n is the number of nonmissing values for a variable, x_i is the ith value of the variable, $\bar{X}_w$ is the weighted mean, w_i is the weight associated with the ith value of the variable, and d is the divisor controlled by the VARDEF= option in the PROC CAPABILITY statement. If there is no WEIGHT= variable, the formula reduces to

$$\sqrt{\frac{1}{d}\sum_{i=1}^{n} (x_i - \bar{X}_w)^2}$$

Skewness

The sample skewness is calculated as

$$\frac{n}{(n-1)(n-2)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{X}}{s} \right)^3$$

where n is the number of nonmissing values for a variable and must be greater than 2, x_i is the ith value of the variable, $\bar{X}$ is the sample average, and s is the sample standard deviation.

The sample skewness can be positive or negative; it measures the asymmetry of the data distribution and estimates the theoretical skewness $\sqrt{\beta_1} = \mu_3 \mu_2^{-\frac{3}{2}}$, where μ_2 and μ_3 are the second and third central moments. Observations that are normally distributed should have a skewness near zero.

Kurtosis

The sample kurtosis is calculated as

$$\frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{X}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$$

where $n > 3$. The sample kurtosis measures the heaviness of the tails of the data distribution. It estimates the adjusted theoretical kurtosis denoted as $\beta_2 - 3$, where $\beta_2 = \frac{\mu_4}{\mu_2{}^2}$, and μ_4 is the fourth central moment. Observations that are normally distributed should have a kurtosis near zero.

Coefficient of Variation (CV)

The coefficient of variation is calculated as

$$CV = \frac{100 \times s}{\bar{X}}$$

Geometric Mean

The geometric mean is calculated as

$$\left(\prod_{i=1}^{n} x_i^{w_i} \right)^{1/\sum_{i=1}^{n} w_i}$$

where n is the number of nonmissing values for a variable, x_i is the ith value of the variable, and w_i is the weight associated with the ith value of the variable.

If there is no WEIGHT variable, the formula reduces to

$$\left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

If any x_i is negative, the geometric mean is set to missing.

Signed Rank Statistic

The signed rank statistic S is computed as

$$S = \sum_{i:x_i>\mu_0} r_i^+ - \frac{n(n+1)}{4}$$

where r_i^+ is the rank of $|x_i - \mu_0|$ after discarding values of $x_i = \mu_0$, and n is the number of x_i values not equal to μ_0. Average ranks are used for tied values.

If $n \leq 20$, the significance of S is computed from the exact distribution of S, where the distribution is a convolution of scaled binomial distributions. When $n > 20$, the significance of S is computed by treating

$$S\sqrt{\frac{n-1}{nV - S^2}}$$

as a Student t variate with $n - 1$ degrees of freedom. V is computed as

$$V = \frac{1}{24}n(n+1)(2n+1) - \frac{1}{48}\sum t_i(t_i+1)(t_i-1)$$

where the sum is over groups tied in absolute value and where t_i is the number of values in the ith group (Iman 1974, Conover 1980). The null hypothesis tested is that the mean (or median) is μ_0, assuming that the distribution is symmetric. Refer to Lehmann and D'Abrera (1975).

Tests for Normality

You can use the NORMALTEST option in the PROC CAPABILITY statement to request several tests of the hypothesis that the analysis variable values are a random sample from a normal distribution. These tests, which are summarized in the table labeled *Tests for Normality*, include the following:

- Shapiro-Wilk test
- Kolmogorov-Smirnov test
- Anderson-Darling test
- Cramér-von Mises test

Tests for normality are particularly important in process capability analysis because the commonly used capability indices are difficult to interpret unless the data are at least approximately normally distributed. Furthermore, the confidence limits for capability indices displayed in the table labeled *Process Capability Indices* require the assumption of normality. Consequently, the tests of normality are always computed when you specify the SPEC statement, and a note is added to the table when the hypothesis of normality is rejected. You can specify the particular test and the significance level with the CHECKINDICES option.

Shapiro-Wilk Test

If the sample size is 2000 or less, the procedure computes the Shapiro-Wilk statistic W (also denoted as W_n to emphasize its dependence on the sample size n). The statistic W_n is the ratio of the best estimator of the variance (based on the square of a linear combination of the order statistics) to the usual corrected sum of squares estimator of the variance. When n is greater than three, the coefficients to compute the linear

combination of the order statistics are approximated by the method of Royston (1992). The statistic W_n is always greater than zero and less than or equal to one ($0 < W \leq 1$).

Small values of W lead to rejection of the null hypothesis. The method for computing the p-value (the probability of obtaining a W statistic less than or equal to the observed value) depends on n. For $n = 3$, the probability distribution of W is known and is used to determine the p-value. For $n > 4$, a normalizing transformation is computed:

$$Z_n = \begin{cases} (-\log(\gamma - \log(1 - W_n)) - \mu)/\sigma & \text{if } 4 \leq n \leq 11 \\ (\log(1 - W_n) - \mu)/\sigma & \text{if } 12 \leq n \leq 2000 \end{cases}$$

The values of σ, γ, and μ are functions of n obtained from simulation results. Large values of Z_n indicate departure from normality, and because the statistic Z_n has an approximately standard normal distribution, this distribution is used to determine the p-values for $n > 4$.

EDF Tests for Normality

The Kolmogorov-Smirnov, Anderson-Darling and Cramér-von Mises tests for normality are based on the empirical distribution function (EDF) and are often referred to as EDF tests. EDF tests for a variety of non-normal distributions are available in the HISTOGRAM statement; see the section "EDF Goodness-of-Fit Tests" on page 351 for details. For a thorough discussion of these tests, refer to D'Agostino and Stephens (1986).

The empirical distribution function is defined for a set of n independent observations $X_1, \ldots, X_n$ with a common distribution function $F(x)$. Under the null hypothesis, $F(x)$ is the normal distribution. Denote the observations ordered from smallest to largest as $X_{(1)}, \ldots, X_{(n)}$. The empirical distribution function, $F_n(x)$, is defined as

$$F_n(x) = \begin{cases} 0, & x < X_{(1)} \\ \frac{i}{n}, & X_{(i)} \leq x < X_{(i+1)}, i = 1, \ldots, n-1 \\ 1, & X_{(n)} \leq x \end{cases}$$

Note that $F_n(x)$ is a step function that takes a step of height $\frac{1}{n}$ at each observation. This function estimates the distribution function $F(x)$. At any value x, $F_n(x)$ is the proportion of observations less than or equal to x, while $F(x)$ is the probability of an observation less than or equal to x. EDF statistics measure the discrepancy between $F_n(x)$ and $F(x)$.

The EDF tests make use of the probability integral transformation $U = F(X)$. If $F(X)$ is the distribution function of X, the random variable U is uniformly distributed between 0 and 1. Given n observations $X_{(1)}, \ldots, X_{(n)}$, the values $U_{(i)} = F(X_{(i)})$ are computed. These values are used to compute the EDF test statistics, as described in the next three sections. The CAPABILITY procedures computes the associated p-values by interpolating internal tables of probability levels similar to those given by D'Agostino and Stephens (1986).

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov statistic (D) is defined as

$$D = \sup_x |F_n(x) - F(x)|$$

The Kolmogorov-Smirnov statistic belongs to the supremum class of EDF statistics. This class of statistics is based on the largest vertical difference between $F(x)$ and $F_n(x)$.

The Kolmogorov-Smirnov statistic is computed as the maximum of D^+ and D^-, where D^+ is the largest vertical distance between the EDF and the distribution function when the EDF is greater than the distribution function, and D^- is the largest vertical distance when the EDF is less than the distribution function.

$$\begin{aligned} D^+ &= \max_i \left(\frac{i}{n} - U_{(i)}\right) \\ D^- &= \max_i \left(U_{(i)} - \frac{i-1}{n}\right) \\ D &= \max\left(D^+, D^-\right) \end{aligned}$$

PROC CAPABILITY uses a modified Kolmogorov D statistic to test the data against a normal distribution with mean and variance equal to the sample mean and variance.

Anderson-Darling Test

The Anderson-Darling statistic and the Cramér-von Mises statistic belong to the quadratic class of EDF statistics. This class of statistics is based on the squared difference $(F_n(x) - F(x))^2$. Quadratic statistics have the following general form:

$$Q = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, \psi(x) dF(x)$$

The function $\psi(x)$ weights the squared difference $(F_n(x) - F(x))^2$.

The Anderson-Darling statistic (A^2) is defined as

$$A^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \left[F(x)\left(1 - F(x)\right)\right]^{-1} dF(x)$$

Here the weight function is $\psi(x) = \left[F(x)\left(1 - F(x)\right)\right]^{-1}$.

The Anderson-Darling statistic is computed as

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^{n} \left[(2i-1) \log U_{(i)} + (2n + 1 - 2i) \log\left(\{1 - U_{(i)}\right)\right]$$

Cramér-von Mises Test

The Cramér-von Mises statistic (W^2) is defined as

$$W^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, dF(x)$$

Here the weight function is $\psi(x) = 1$.

The Cramér-von Mises statistic is computed as

$$W^2 = \sum_{i=1}^{n} \left(U_{(i)} - \frac{2i-1}{2n}\right)^2 + \frac{1}{12n}$$

Percentile Computations

The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th, and 99th percentiles (quantiles), as well as the minimum and maximum of each analysis variable. To compute percentiles other than these default percentiles, use the PCTLPTS= and PCTLPRE= options in the OUTPUT statement.

You can specify one of five definitions for computing the percentiles with the PCTLDEF= option. Let n be the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the variable. Let the tth percentile be y, set $p = \frac{t}{100}$, and let

$$\begin{aligned} np &= j + g \quad \text{when PCTLDEF=1, 2, 3, or 5} \\ (n+1)p &= j + g \quad \text{when PCTLDEF=4} \end{aligned}$$

where j is the integer part of np, and g is the fractional part of np. Then the PCTLDEF= option defines the tth percentile, y, as described in the following table:

PCTLDEF=	Description	Formula
1	weighted average at x_{np}	$y = (1-g)x_j + gx_{j+1}$ where x_0 is taken to be x_1
2	observation numbered closest to np	$y = x_i$ if $g \neq \frac{1}{2}$ $y = x_j$ if $g = \frac{1}{2}$ and j is even $y = x_{j+1}$ if $g = \frac{1}{2}$ and j is odd where i is the integer part of $np + \frac{1}{2}$
3	empirical distribution function	$y = x_j$ if $g = 0$ $y = x_{j+1}$ if $g > 0$
4	weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$ where x_{n+1} is taken to be x_n
5	empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1})$ if $g = 0$ $y = x_{j+1}$ if $g > 0$

Weighted Percentiles

When you use a WEIGHT statement, the percentiles are computed differently. The 100pth weighted percentile y is computed from the empirical distribution function with averaging

$$y = \begin{cases} x_1 & \text{if } w_1 > pW \\ \frac{1}{2}(x_i + x_{i+1}) & \text{if } \sum_{j=1}^{i} w_j = pW \\ x_{i+1} & \text{if } \sum_{j=1}^{i} w_j < pW < \sum_{j=1}^{i+1} w_j \end{cases}$$

where w_i is the weight associated with x_i, and where $W = \sum_{i=1}^{n} w_i$ is the sum of the weights.

Note that the PCTLDEF= option is not applicable when a WEIGHT statement is used. However, in this case, if all the weights are identical, the weighted percentiles are the same as the percentiles that would be computed without a WEIGHT statement and with PCTLDEF=5.

Confidence Limits for Percentiles

You can use the CIPCTLNORMAL option to request confidence limits for percentiles which assume the data are normally distributed. These limits are described in Section 4.4.1 of Hahn and Meeker (1991). When $0.0 < p < 0.5$, the two-sided $100(1-\alpha)\%$ confidence limits for the $100p$-th percentile are

$$\begin{aligned} \text{lower limit} &= \bar{X} - g'(\alpha/2; 1-p, n)s \\ \text{upper limit} &= \bar{X} - g'(1-\alpha/2; p, n)s \end{aligned}$$

where n is the sample size. When $0.5 \leq p < 1.0$, the two-sided $100(1-\alpha)\%$ confidence limits for the $100p$-th percentile are

$$\begin{aligned} \text{lower limit} &= \bar{X} + g'(\alpha/2; 1-p, n)s \\ \text{upper limit} &= \bar{X} + g'(1-\alpha/2; p, n)s \end{aligned}$$

One-sided $100(1-\alpha)\%$ confidence bounds are computed by replacing $\alpha/2$ by α in the appropriate preceding equation. The factor $g'(\gamma, p, n)$ is related to the noncentral t distribution and is described in Owen and Hua (1977) and Odeh and Owen (1980).

You can use the CIPCTLDF option to request confidence limits for percentiles which are distribution free (in particular, it is not necessary to assume that the data are normally distributed). These limits are described in Section 5.2 of Hahn and Meeker (1991). The two-sided $100(1-\alpha)\%$ confidence limits for the $100p$-th percentile are

$$\begin{aligned} \text{lower limit} &= X_{(l)} \\ \text{upper limit} &= X_{(u)} \end{aligned}$$

where $X_{(j)}$ is the jth order statistic when the data values are arranged in increasing order:

$$X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$$

The lower rank l and upper rank u are integers that are symmetric (or nearly symmetric) around $\lfloor np \rfloor + 1$ where $\lfloor np \rfloor$ is the integer part of np, and where n is the sample size. Furthermore, l and u are chosen so that $X_{(l)}$ and $X_{(u)}$ are as close to $X_{\lfloor np \rfloor + 1}$ as possible while satisfying the coverage probability requirement

$$Q(u-1; n, p) - Q(l-1; n, p) \geq 1 - \alpha$$

where $Q(k; n, p)$ is the cumulative binomial probability

$$Q(k; n, p) = \sum_{i=0}^{k} \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i}$$

In some cases, the coverage requirement cannot be met, particularly when n is small and p is near 0 or 1. To relax the requirement of symmetry, you can specify CIPCTLDF(TYPE = ASYMMETRIC). This option requests symmetric limits when the coverage requirement can be met, and asymmetric limits otherwise.

If you specify CIPCTLDF(TYPE = LOWER), a one-sided $100(1-\alpha)\%$ lower confidence bound is computed as X_l, where l is the largest integer that satisfies the inequality

$$1 - Q(l-1; n, p) \geq 1 - \alpha$$

with $0 < l \leq n$. Likewise, if you specify CIPCTLDF(TYPE = UPPER), a one-sided $100(1-\alpha)\%$ lower confidence bound is computed as X_l, where l is the largest integer that satisfies the inequality

$$Q(u-1; n, p) \geq 1-\alpha$$

where $0 < u \leq n$.

Note that confidence limits for percentiles are not computed when a WEIGHT statement is specified.

Robust Estimators

The CAPABILITY procedure provides several methods for computing robust estimates of location and scale, which are insensitive to outliers in the data.

Winsorized Means

The k-times Winsorized mean is a robust estimator of location which is computed as

$$\bar{x}_{wk} = \frac{1}{n}\left((k+1)x_{(k+1)} + \sum_{i=k+2}^{n-k-1} x_{(i)} + (k+1)x_{(n-k)}\right)$$

where n is the number of observations, and $x_{(i)}$ is the ith order statistic when the observations are arranged in increasing order:

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$$

The Winsorized mean is the mean computed after replacing the k smallest observations with the $(k+1)$st smallest observation, and the k largest observations with the $(k+1)$st largest observation.

For data from a symmetric distribution, the Winsorized mean is an unbiased estimate of the population mean. However, the Winsorized mean does not have a normal distribution even if the data are normally distributed.

The Winsorized sum of squared deviations is defined as

$$s^2_{wk} = (k+1)(x_{(k+1)} - \bar{x}_{wk})^2 + \sum_{i=k+2}^{n-k-1}(x_{(i)} - \bar{x}_{wk})^2 + (k+1)(x_{(n-k)} - \bar{x}_{wk})^2$$

A Winsorized t test is given by

$$t_{wk} = \frac{\bar{x}_{wk} - \mu_0}{\mathrm{STDERR}(\bar{x}_{wk})}$$

where the standard error of the Winsorized mean is

$$\mathrm{STDERR}(\bar{x}_{wk}) = \frac{n-1}{n-2k-1}\frac{s_{wk}}{\sqrt{n(n-1)}}$$

When the data are from a symmetric distribution, the distribution of t_{wk} is approximated by a Student's t distribution with $n-2k-1$ degrees of freedom. Refer to Tukey and McLaughlin (1963) and Dixon and Tukey (1968).

A $100(1-\alpha)\%$ Winsorized confidence interval for the mean has upper and lower limits

$$\bar{x}_{wk} \pm t_{1-\alpha/2}\mathrm{STDERR}(\bar{x}_{wk})$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)100$th percentile of the Student's t distribution with $n-2k-1$ degrees of freedom.

Trimmed Means

The k-times trimmed mean is a robust estimator of location which is computed as

$$\bar{x}_{tk} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} x_{(i)}$$

where n is the number of observations, and $x_{(i)}$ is the ith order statistic when the observations are arranged in increasing order:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The trimmed mean is the mean computed after the k smallest observations and the k largest observations in the sample are deleted.

For data from a symmetric distribution, the trimmed mean is an unbiased estimate of the population mean. However, the trimmed mean does not have a normal distribution even if the data are normally distributed.

A robust estimate of the variance of the trimmed mean t_{tk} can be obtained from the Winsorized sum of squared deviations; refer to Tukey and McLaughlin (1963). the corresponding trimmed t test is given by

$$t_{tk} = \frac{\bar{x}_{tk} - \mu_0}{\mathrm{STDERR}(\bar{x}_{tk})}$$

where the standard error of the trimmed mean is

$$\mathrm{STDERR}(\bar{x}_{tk}) = \frac{s_{tk}}{\sqrt{(n-2k)(n-2k-1)}}$$

and s_{wk} is the square root of the Winsorized sum of squared deviations.

When the data are from a symmetric distribution, the distribution of t_{tk} is approximated by a Student's t distribution with $n-2k-1$ degrees of freedom. Refer to Tukey and McLaughlin (1963) and Dixon and Tukey (1968).

A $100(1-\alpha)\%$ trimmed confidence interval for the mean has upper and lower limits

$$\bar{x}_{tk} \pm t_{1-\alpha/2}\mathrm{STDERR}(\bar{x}_{tk})$$

where $t_{1-\alpha/2}$ is the $(1-\alpha/2)100$th percentile of the Student's t distribution with $n-2k-1$ degrees of freedom.

Robust Estimates of Scale

The sample standard deviation, which is the most commonly used estimator of scale, is sensitive to outliers. Robust scale estimators, on the other hand, remain bounded when a single data value is replaced by an arbitrarily large or small value. The CAPABILITY procedure computes several robust measures of scale, including the interquartile range Gini's mean difference G, the median absolute deviation about the median (MAD), Q_n, and S_n. In addition, the procedure computes estimates of the normal standard deviation σ derived from each of these measures.

The interquartile range (IQR) is simply the difference between the upper and lower quartiles. For a normal population, σ can be estimated as IQR/1.34898.

Gini's mean difference is computed as

$$G = \frac{1}{\binom{n}{2}} \sum_{i<j} |x_i - x_j|$$

For a normal population, the expected value of G is $2\sigma/\sqrt{\pi}$. Thus $G\sqrt{\pi}/2$ is a robust estimator of σ when the data are from a normal sample. For the normal distribution, this estimator has high efficiency relative to the usual sample standard deviation, and it is also less sensitive to the presence of outliers.

A very robust scale estimator is the MAD, the median absolute deviation from the median (Hampel 1974), which is computed as

$$\text{MAD} = \text{med}_i(|x_i - \text{med}_j(x_j)|)$$

where the inner median, $\text{med}_j(x_j)$, is the median of the n observations, and the outer median (taken over i) is the median of the n absolute values of the deviations about the inner median. For a normal population, 1.4826MAD is an estimator of σ.

The MAD has low efficiency for normal distributions, and it may not always be appropriate for symmetric distributions. Rousseeuw and Croux (1993) proposed two statistics as alternatives to the MAD. The first is

$$S_n = 1.1926\ \text{med}_i(\text{med}_j(|x_i - x_j|))$$

where the outer median (taken over i) is the median of the n medians of $|x_i - x_j|$, $j = 1, 2, \ldots, n$. To reduce small-sample bias, $c_{sn}S_n$ is used to estimate σ, where c_{sn} is a correction factor; refer to Croux and Rousseeuw (1992).

The second statistic is

$$Q_n = 2.2219\ \{|x_i - x_j|; i < j\}_{(k)}$$

where

$$k = \binom{h}{2}$$

and $h = [n/2] + 1$. In other words, Q_n is 2.2219 times the kth order statistic of the $\binom{n}{2}$ distances between the data points. The bias-corrected statistic $c_{qn}Q_n$ is used to estimate σ, where c_{qn} is a correction factor; refer to Croux and Rousseeuw (1992).

Computing the Mode

The mode is the value that occurs most often in a set of observations. The CAPABILITY procedure counts repetitions of the actual values (or the rounded values, if you specify the ROUND= option). If a tie occurs for the most frequent value, the procedure reports the lowest mode in the table labeled *Basic Statistical Measures*. To list all possible modes, specify the MODES option in the PROC CAPABILITY statement. When no repetitions occur in the data, the procedure does not report the mode. The WEIGHT statement has no effect on the mode.

Assumptions and Terminology for Capability Indices

One of the fundamental assumptions in process capability analysis is that the process must be in statistical control. Without statistical control, the process is not predictable, the concept of a process distribution does not apply, and quantities related to the distribution, such as probabilities, percentiles, and capability indices, cannot be meaningfully estimated. Additionally, all of the standard process capability indices described in the next section require that the process distribution be normal, or at least approximately normal.

In many industries, statistical control is routinely checked with a Shewhart chart (such as an $\bar{X}$ and R chart) before capability indices such as

$$C_{pk} = \min\left(\frac{\mathrm{USL}-\mu}{3\sigma}, \frac{\mathrm{LSL}-\mu}{3\sigma}\right)$$

are computed. The control chart analysis yields estimates for the process mean μ and standard deviation σ, which are based on subgrouped data and can be used to estimate C_{pk}. In particular, σ can be estimated by

$$s_R = \bar{R}/d_2$$

rather than the ungrouped sample standard deviation

$$s = \frac{1}{n-1}\sqrt{\sum_{i=1} n(x_i - \bar{x})^2}$$

You can use the SHEWHART procedure to carry out the control chart analysis and to compute capability indices based on s_R. On the other hand, the CAPABILITY procedure computes indices based on s.

Some industry manuals distinguish these two approaches. For instance, the ASQC/AIAG manual *Fundamental Process Control* uses the notation C_{pk} for the estimate based on s_R, and it uses the notation P_{pk} for the estimate based on s. However, assuming that the process is in control and only common cause variation is present, both s_R and s are estimates of the same parameter σ, and so there is fundamentally no difference in the two approaches[2].

Once control has been established, attention should focus on the distribution of the process measurements, and at this point there is no practical or statistical advantage to working with subgrouped measurements. In fact, the use of s is closely associated with a wide variety of methods that are highly useful for process capability analysis, including tests for normality, graphical displays such as histograms and probability plots, and confidence intervals for parameters and capability indices.

Standard Capability Indices

This section provides computational details for the standard process capability indices computed by the CAPABILITY procedure: C_p, CPL, CPU, C_{pk}, and C_{pm}.

The Index Cp

The process capability index C_p, sometimes called the "process potential index," the "process capability ratio," or the "inherent capability index," is estimated as

$$\widehat{C}_p = \frac{USL - LSL}{6s}$$

[2]Statistically, s is a more efficient estimator of σ than s_R.

where *USL* is the upper specification limit, *LSL* is the lower specification limit, and *s* is the sample standard deviation. If you do not specify both the upper and the lower specification limits in the SPEC statement or the SPEC= data set, then C_p is assigned a missing value.

The interpretation of C_p can depend on the application, on past experience, and on local practice. However, broad guidelines for interpretation have been proposed by several authors. Ekvall and Juran (1974) classify C_p values as

- "not adequate" if $C_p < 1$
- "adequate" if $1 \leq C_p \leq 1.33$, but requiring close control as C_p approaches 1
- "more than adequate" if $C_p > 1.33$

Montgomery (1996) recommends minimum values of C_p as

- 1.33 for existing processes
- 1.50 for new processes or for existing processes when the variable is critical (for example, related to safety or strength)
- 1.67 for new processes when the variable is critical

Exact $100(1-\alpha)\%$ lower and upper confidence limits for C_p (denoted by LCL and UCL) are computed using percentiles of the chi-square distribution, as indicated by the following equations:

$$\begin{aligned} \text{lower limit} &= \hat{C}_p \sqrt{\chi^2_{\alpha/2,n-1}/(n-1)} \\ \text{upper limit} &= \hat{C}_p \sqrt{\chi^2_{1-\alpha/2,n-1}/(n-1)} \end{aligned}$$

Here, $\chi^2_{\alpha,\nu}$ denotes the lower 100αth percentile of the chi-square distribution with ν degrees of freedom. Refer to Chou, Owen, and Borrego (1990) and Kushler and Hurley (1992).

You can specify α with the ALPHA= option in the PROC CAPABILITY statement or with the CIINDICES(ALPHA=*value*) in the PROC CAPABILITY statement. The default value is 0.05. You can save these limits in the OUT= data set by specifying the keywords CPLCL and CPUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying the keywords in the INSET statement.

The Index CPL

The process capability index *CPL* is estimated as

$$\widehat{\mathrm{CPL}} = \frac{\bar{X} - LSL}{3s}$$

where $\bar{X}$ is the sample mean, *LSL* is the lower specification limit, and *s* is the sample standard deviation. If you do not specify the lower specification limit in the SPEC statement or the SPEC= data set, then *CPL* is assigned a missing value.

Montgomery (1996) refers to *CPL* as the "process capability ratio" in the case of one-sided lower specifications and recommends minimum values as follows:

- 1.25 for existing processes
- 1.45 for new processes or for existing processes when the variable is critical
- 1.60 for new processes when the variable is critical

Exact $100(1-\alpha)\%$ lower and upper confidence limits for *CPL* are computed using a generalization of the method of Chou, Owen, and Borrego (1990), who point out that the $100(1-\alpha)$ lower confidence limit for *CPL* (denoted by CPLLCL)satisfies the equation

$$\Pr\{T_{n-1}(\delta = 3\sqrt{n})\ \text{CPLLCL}\ \leq 3\text{CPL}\sqrt{n}\} = 1-\alpha$$

where $T_{n-1}(\delta)$ has a non-central *t* distribution with *n* – 1 degrees of freedom and noncentrality parameter δ. You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. The confidence limits can be saved in an output data set by specifying the keywords CPLLCL and CPLUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these keywords in the INSET statement.

The Index CPU

The process capability index *CPU* is estimated as

$$\widehat{\text{CPU}} = \frac{USL - \bar{X}}{3s}$$

where *USL* is the upper specification limit, $\bar{X}$ is the sample mean, and *s* is the sample standard deviation. If you do not specify the upper specification limit in the SPEC statement or the SPEC= data set, then *CPU* is assigned a missing value.

Montgomery (1996) refers to *CPU* as the "process capability ratio" in the case of one-sided upper specifications and recommends minimum values that are the same as those specified previously for *CPL*.

Exact $100(1-\alpha)\%$ lower and upper confidence limits for *CPU* are computed using a generalization of the method of Chou, Owen, and Borrego (1990), who point out that the $100(1-\alpha)$ lower confidence limit for *CPU* (denoted by CPULCL)satisfies the equation

$$\Pr\{T_{n-1}(\delta = 3\sqrt{n}\ \text{CPULCL}\ \geq 3\text{CPU}\sqrt{n}\} = 1-\alpha$$

where $T_{n-1}(\delta)$ has a non-central *t* distribution with *n* – 1 degrees of freedom and noncentrality parameter δ. You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. The confidence limits can be saved in an output data set by specifying the keywords CPULCL and CPUUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these keywords in the INSET statement.

The Index Cpk

The process capability index C_{pk} is defined as

$$C_{pk} = \frac{1}{3\sigma}\min(USL-\mu,\ \mu-LSL) = \min(CPU,\ CPL)$$

Note that the indices C_{pk}, C_p, and k are related as $C_{pk} = C_p(1-k)$. The CAPABILITY procedure estimates C_{pk} as

$$\widehat{C}_{pk} = \frac{1}{3s} \times \min(USL - \bar{X},\ \bar{X} - LSL) = \min(CPU,\ CPL)$$

where *USL* is the upper specification limit, *LSL* is the lower specification limit, $\bar{X}$ is the sample mean, and s is the sample standard deviation.

If you specify only the upper limit in the SPEC statement or the SPEC= data set, then C_{pk} is computed as *CPU*, and if you specify only the lower limit in the SPEC statement or the SPEC= data set, then C_{pk} is computed as *CPL*.

Bissell (1990) derived approximate two-sided 95% confidence limits for C_{pk} by assuming that the distribution of $\widehat{C}_{pk}$ is normal. Using Bissell's approach, $100(1-\alpha)\%$ lower and upper confidence limits can be computed as

$$\text{lower limit} = \widehat{C}_{pk}\left[1 - \Phi^{-1}(1-\alpha/2)\sqrt{\frac{1}{9n\widehat{C}_{pk}^2} + \frac{1}{2(n-1)}}\ \right]$$

$$\text{upper limit} = \widehat{C}_{pk}\left[1 + \Phi^{-1}(1-\alpha/2)\sqrt{\frac{1}{9n\widehat{C}_{pk}^2} + \frac{1}{2(n-1)}}\ \right]$$

where Φ denotes the cumulative standard normal distribution function. Kushler and Hurley (1992) concluded that Bissell's method gives reasonably accurate results.

You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. These limits can be saved in an output data set by specifying the keywords CPKLCL and CPKUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these same keywords in the INSET statement.

The Index Cpm

The process capability index C_{pm} is intended to account for deviation from the target T in addition to variability from the mean. This index is often defined as

$$C_{pm} = \frac{\text{USL} - \text{LSL}}{6\sqrt{\sigma^2 + (\mu - T)^2}}$$

A closely related version of C_{pm} is the index

$$C_{pm}^* = \frac{\min(\text{USL} - T, T - \text{LSL})}{3\sqrt{\sigma^2 + (\mu - T)^2}} = \frac{d - |T - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $d = (\text{USL} - \text{LSL})/2$ and $m = (\text{USL} + \text{LSL})/2$. If $T = m$, then $C_{pm} = C_{pm}^*$. However, if $T \neq m$, then both indices suffer from problems of interpretation, as pointed out by Kotz and Johnson (1993), and their use should be avoided in this case.

The CAPABILITY procedure computes an estimator of C_{pm} as

$$\widehat{C}_{pm} = \frac{\min(USL - T,\ T - LSL)}{3\sqrt{s^2 + (\bar{X} - T)^2}}$$

where s is the sample standard deviation.

If you specify only a single specification limit SL in the SPEC statement or the SPEC= data set, then C_{pm} is estimated as

$$\widehat{C}_{pm} = \frac{|T - SL|}{3\sqrt{s^2 + (\bar{X} - T)^2}}$$

Boyles (1991) proposed a slightly modified point estimate for C_{pm} computed as

$$\widetilde{C}_{pm} = \frac{(\mathrm{USL} - \mathrm{LSL})/2}{3\sqrt{(\frac{n-1}{n})s^2 + (\bar{X} - T)^2}}$$

Boyles also suggested approximate two-sided $100(1-\alpha)\%$ confidence limits for C_{pm}, which are computed as

$$\begin{aligned} \text{lower limit} &= \widetilde{C}_{pm}\sqrt{\chi^2_{\alpha/2,\nu}/\nu} \\ \text{upper limit} &= \widetilde{C}_{pm}\sqrt{\chi^2_{1-\alpha/2,\nu}/\nu} \end{aligned}$$

Here $\chi^2_{\alpha,\nu}$ denotes the lower 100αth percentile of the chi-square distribution with ν degrees of freedom, where ν equals

$$\frac{n(1 + (\frac{\bar{X}-T}{s})^2)}{1 + 2(\frac{\bar{X}-T}{s})^2}$$

You can specify α with the ALPHA= option in the PROC CAPABILITY statement. The default value is 0.05. These confidence limits can be saved in an output data set by specifying the keywords CPMLCL and CPMUCL in the OUTPUT statement. In addition, you can display these limits on plots produced by the CAPABILITY procedure by specifying these keywords in the INSET statement.

Specialized Capability Indices

This section describes a number of specialized capability indices which you can request with the SPECIALINDICES option in the PROC CAPABILITY statement.

The Index k

The process capability index k (also denoted by K) is computed as

$$k = \frac{2|m - \bar{X}|}{USL - LSL}$$

where $m = \frac{1}{2}(USL + LSL)$ is the midpoint of the specification limits, $\bar{X}$ is the sample mean, *USL* is the upper specification limit, and *LSL* is the lower specification limit.

The formula for k used here is given by Kane (1986). Note that k is sometimes computed without taking the absolute value of $m - \bar{X}$ in the numerator. See Wadsworth, Stephens, and Godfrey (1986).

If you do not specify the upper and lower limits in the SPEC statement or the SPEC= data set, then k is assigned a missing value.

Boyles' Index C_{pm}^{+}

Boyles (1992) proposed the process capability index C_{pm}^{+} which is defined as

$$C_{pm}^{+} = \frac{1}{3}\left[\frac{E_{X<T}\left[(X-T)^2\right]}{(T-\mathrm{LSL})^2} + \frac{E_{X>T}\left[(X-T)^2\right]}{(\mathrm{USL}-T)^2}\right]^{-1/2}$$

He proposed this index as a modification of C_{pm} for use when $\mu \neq T$. The quantities

$$E_{X<T}\left[(X-T)^2\right] = E\left[(X-T)^2 | X < T\right] Pr\left[X < T\right]$$

and

$$E_{X>T}\left[(X-T)^2\right] = E\left[(X-T)^2 | X > T\right] Pr\left[X > T\right]$$

are referred to as semivariances. Kotz and Johnson (1993) point out that if $T = (\mathrm{LSL} + \mathrm{USL})/2$, then $C_{pm}^{+} = C_{pm}$.

Kotz and Johnson (1993) suggest that a natural estimator for C_{pm}^{+} is

$$\widehat{C}_{pm}^{+} = \frac{1}{3}\left[\frac{1}{n}\left\{\frac{\sum_{X_i<T}(X_i-T)^2}{(T-\mathrm{LSL})^2} + \frac{\sum_{X_i>T}(X_i-T)^2}{(\mathrm{USL}-T)^2}\right\}^{-1/2}\right]$$

Note that this index is not defined when either of the specification limits is equal to the target T. Refer to Section 3.5 of Kotz and Johnson (1993) for further details.

The Index C_{jkp}

Johnson, Kotz, and Pearn (1994) introduced a so-called "flexible" process capability index which takes into account possible differences in variability above and below the target T. They defined this index as

$$C_{jkp} = \frac{1}{3\sqrt{2}} \min\left(\frac{\mathrm{USL}-T}{\sqrt{E_{X>T}[(X-T)^2]}}, \frac{T-\mathrm{LSL}}{\sqrt{E_{X<T}[(X-T)^2]}}\right)$$

where $d = (\mathrm{USL} - \mathrm{LSL})/2$.

A natural estimator of this index is

$$\widehat{C}_{jkp} = \frac{1}{3\sqrt{2}} \min\left(\frac{\text{USL} - T}{\sqrt{\sum_{X_i > T}(X_i - T)^2/n}}, \frac{T - \text{LSL}}{\sqrt{\sum_{X_i < T}(X_i - T)^2/n}} \right)$$

For further details, refer to Section 4.4 of Kotz and Johnson (1993).

The Indices $C_{pm}(a)$

The class of capability indices $C_{pm}(a)$, indexed by the parameter a ($a > 0$) allows flexibility in choosing between the relative importance of variability and deviation of the mean from the target value T.

The class defined as

$$C_{pm}(a) = (1 - a\zeta^2)C_p$$

where $\zeta = (\mu - T)/\sigma$. The motivation for this definition is that if $|\zeta|$ is small, then

$$C_{pm} \approx (1 - \frac{1}{2}\zeta^2)C_p$$

A natural estimator of $C_{pm}(a)$ is

$$\frac{d}{3s}\widehat{C}_{pm}(a) = \left\{ 1 - a\left(\frac{\bar{X} - T}{s} \right)^2 \right\}$$

where $d = (\text{USL} - \text{LSL})/2$. You can specify the value of a with the SPECIALINDICES(CPMA=) option in the PROC CAPABILITY statement. By default, $a = 0.5$.

This index is not recommended for situation in which the target T is not equal to the midpoint of the specification limits.

For additional details, refer to Section 3.7 of Kotz and Johnson (1993).

The Index $C_{p(5.15)}$

Johnson *et al.* (1992) suggest the class of process capability indices defined as

$$C_{p(\theta)} = \frac{\text{USL} - \text{LSL}}{\theta\sigma}$$

where θ is chosen so that the proportion of conforming items is robust with respect to the shape of the process distribution. In particular, Kotz and Johnson (1993) recommend use of

$$C_{p(5.15)} = \frac{\text{USL} - \text{LSL}}{5.15\sigma}$$

which is estimated as

$$\widehat{C}_{p(5.15)} = \frac{\text{USL} - \text{LSL}}{5.15s}$$

For details, refer to Section 4.3.2 of Kotz and Johnson (1993).

The Index $C_{pk(5.15)}$

Similarly, Kotz and Johnson (1993) recommend use of the robust capability index

$$C_{pk(5.15)} = \frac{d - |\mu - (\mathrm{USL} + \mathrm{LSL})/2|}{2.575\sigma}$$

where $d = (\mathrm{USL} - \mathrm{LSL})/2$. This index is estimated as

$$\widehat{C}_{pk(5.15)} = \frac{d - |\bar{X} - (\mathrm{USL} + \mathrm{LSL})/2|}{2.575s}$$

For details, refer to Section 4.3.2 of Kotz and Johnson (1993).

The Index C_{pmk}

Pearn, Kotz, and Johnson (1992) proposed the index C_{pmk}

$$C_{pmk} = \frac{(\mathrm{USL} - \mathrm{LSL})/2 - |\mu - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $m = (\mathrm{LCL} + \mathrm{UCL})/2$. A natural estimator for C_{pmk} is

$$\widehat{C}_{pmk} = \frac{(\mathrm{USL} - \mathrm{LSL})/2 - |\bar{X} - m|}{3\sqrt{(\frac{n-1}{n})s^2 + (\bar{X} - T)^2}}$$

where $m = (\mathrm{USL} + \mathrm{LSL})/2$.

For further details, refer to Section 3.6 of Kotz and Johnson (1993).

Wright's Index C_s

Wright (1995) defines the capability index

$$C_s = \frac{\min(\mathrm{USL} - \mu, \mu - \mathrm{LSL})}{3\sqrt{\sigma^2 + (\mu - T)^2 + \mu_3/\sigma}}$$

where $\mu_3 = E(X - \mu)^3$.

A natural estimator of C_s is

$$\widehat{C}_s = \frac{(\mathrm{USL} - \mathrm{LSL})/2 - |\bar{X} - m|}{3\sqrt{\left(\frac{n-1}{n}\right)s^2 + (\bar{X} - T)^2 + |c_4 s^2 b_3|}}$$

where c_4 is an unbiasing constant for the sample standard deviation, and b_3 is a measure of skewness. Wright (1995) shows that C_s compares favorably with C_{pmk} even when skewness is not present, and he advocates the use of C_s for monitoring near-normal processes when loss of capability typically leads to asymmetry.

Chen and Kotz (1996) proposed a modification to Wright's C_s index which introduces a multiplier, $\gamma > 0$, and is estimated as

$$\widehat{C}_s = \frac{(\mathrm{USL} - \mathrm{LSL})/2 - |\bar{X} - m|}{3\sqrt{\left(\frac{n-1}{n}\right)s^2 + (\bar{X} - T)^2 + \gamma|c_4 s^2 b_3|}}$$

If you specify a value for γ with the SPECIALINDICES(CSGAMMA=) option, the index C_s is computed with this modification. Otherwise it is computed using Wright's original definition.

The Index S_{jkp}

Boyles (1994) proposed a smooth version of C_{jkp} defined as

$$S_{jkp} = S\left(\frac{\text{USL} - T}{\sqrt{2E_{X>T}[(X-T)^2]}}, \frac{T - \text{LSL}}{\sqrt{2E_{X<T}[(X-T)^2]}}\right)$$

The CAPABILITY procedure estimates S_{jkp} as

$$\widehat{S}_{jkp} = S\left(\frac{\text{USL} - T}{\sqrt{2\sum_{X_i>T}(X_i-T)^2/n}}, \frac{T - \text{LSL}}{\sqrt{2\sum_{X_i<T}(X_i-T)^2/n}}\right)$$

where $S(x, y) = \Phi^{-1}[\{\Phi(x) + \Phi(y)\}/2]/3$.

The Index C_{pp}

Chen (1998) devised a process incapability index based on the C^*_{pm} index. The first term measures *inaccuracy* and the second measures *imprecision*. The C_{pp} index is estimated as

$$\widehat{C}_{pp} = \left(\frac{\bar{X} - T}{d^*/3}\right)^2 + \left(\frac{s}{d^*/3}\right)^2$$

where $d^* = \min(\text{USL} - T, T - \text{LSL})$.

The Index C''_{pp}

The index C_{pp} does not handle asymmetric tolerances well, as discussed by Kotz and Lovelace (1998). To address that shortcoming, Chen (1998) defined the index C''_{pp}, which is estimated by

$$\widehat{C}''_{pp} = \left(\frac{\widehat{A}}{d^*/3}\right)^2 + \left(\frac{s}{d^*/3}\right)$$

where

$$\widehat{A} = \max\left\{\frac{(\bar{X} - T)d}{T - \text{LSL}}, \frac{(T - \bar{X})d}{\text{USL} - T}\right\}$$

and $d = (USL - LSL)/2$.

The Index C_{pg}

Marcucci and Beazley (1988) defined the index

$$C_{pg} = \frac{1}{C_{pm}^2}$$

which is estimated as

$$\widehat{C}_{pg} = \frac{1}{\widehat{C}_{pm}^2}$$

The Index C_{pq}

Gupta and Kotz (1997) introduced the index C_{pq}, which is estimated by

$$\widehat{C}_{pq} = \widehat{C}_p \left[1 - \frac{1}{2} \left(\frac{\bar{X} - T}{s} \right)^2 \right]$$

The Index C_p^W

Bai and Choi (1997) defined the index

$$C_p^W = \frac{C_p}{\sqrt{1 + |1 - 2P_x|}}$$

where $P_x = \Pr(X \leq \mu)$. It is estimated by

$$\widehat{C}_p^W = \frac{\widehat{C}_p}{\sqrt{1 + |1 - 2\widehat{P}_x|}}$$

where $\widehat{P}_x$ is the fraction of observations less than or equal to $\bar{X}$. For more information about C_p^W, see Kotz and Lovelace (1998).

The Index C_{pk}^W

Bai and Choi (1997) also proposed the index

$$C_{pk}^W = \min \left\{ \frac{\text{USL} - \mu}{3\sigma\sqrt{2P_x}}, \frac{\mu - \text{LSL}}{3\sigma\sqrt{2(1 - P_x)}} \right\}$$

It is estimated by

$$\widehat{C}_{pk}^{W} = \min\left\{\frac{\text{USL} - \bar{X}}{3s\sqrt{2\widehat{P}_x}}, \frac{\bar{X} - \text{LSL}}{3s\sqrt{2(1-\widehat{P}_x)}}\right\}$$

where $\widehat{P}_x$ is the fraction of observations less than or equal to $\bar{X}$. For more information about C_{pk}^{W}, see Kotz and Lovelace (1998).

The Index C_{pm}^{W}

The index C_{pm}^{W}, also introduced by Bai and Choi (1997), is defined as

$$C_{pm}^{W} = \frac{C_{pm}}{\sqrt{1 + |1 - 2P_T|}}$$

where $P_T = \Pr(X \leq T)$. It is estimated by

$$\widehat{C}_{pm}^{W} = \frac{\widehat{C}_{pm}}{\sqrt{1 + |1 - 2\widehat{P}_T|}}$$

where $\widehat{P}_T$ is the fraction of observations less than or equal to T. For more information about C_{pm}^{W}, see Kotz and Lovelace (1998).

The Index C_{pc}

Luceño (1996) proposed the index

$$C_{pc} = \frac{\text{USL} - \text{LSL}}{6\sqrt{\frac{\pi}{2}E|X - M|}}$$

where $M = (\text{USL} + \text{LSL})/2$. It is estimated by

$$\widehat{C}_{pc} = \frac{\text{USL} - \text{LSL}}{6\sqrt{\frac{\pi}{2}c}}$$

where

$$c = \frac{1}{n}\sum_{i=1}^{n}|X_i - M|$$

Vännmann's Index $C_p(u, v)$

Vännmann (1995) introduced the generalized index $C_p(u, v)$, which reduces to the following capability indices given appropriate choices of u and v:

- $C_p(0, 0) = C_p$
- $C_p(0, 1) = C_{pk}$
- $C_p(1, 0) = C_{pm}$
- $C_p(1, 1) = C_{pmk}$

$C_p(u, v)$ is defined as

$$C_p(u, v) = \frac{d - u|\,\mu - M\,|}{3\sqrt{\sigma^2 + v(\,\mu - T)^2}}$$

and estimated by

$$\widehat{C}_p(u, v) = \frac{d - u|\bar{X} - M|}{3\sqrt{(\frac{n-1}{n})s^2 + v(\bar{X} - T)^2}}$$

You can specify u with the SPECIALINDICES(CPU=) option and v with the SPECIALINDICES(CPV=) option. By default, $u = 0$ and $v = 4$.

Vännmann's Index $C_p(v)$

Vännmann (1997) also proposed the index $C_p(v)$, which is equivalent to $C_p(u, v)$ with $u = 1$. It is estimated as

$$\widehat{C}_p(v) = \frac{d - |\bar{X} - M|}{3\sqrt{(\frac{n-1}{n})s^2 + v(\bar{X} - T)^2}}$$

You can specify v with the SPECIALINDICES(CPV=) option. By default, $v = 4$.

Missing Values

If a variable for which statistics are calculated has a missing value, that value is ignored in the calculation of statistics, and the missing values are tabulated separately. A missing value for one such variable does not affect the treatment of other variables in the same observation.

If the WEIGHT variable has a missing value, the observation is excluded from the analysis. If the FREQ variable has a missing value, the observation is excluded from the analysis. If a variable in a BY or ID statement has a missing value, the procedure treats it as it would treat any other value of a BY or ID variable.

ODS Tables

This section describes the ODS tables produced by the CAPABILITY procedure.

Table 6.5 summarizes the ODS tables that you can request with options in the PROC CAPABILITY statement.

Table 6.5 ODS Tables Produced with the PROC CAPABILITY Statement

Table Name	Description	Option
BasicIntervals	confidence intervals for mean, standard deviation, variance	CIBASIC
BasicMeasures	measures of location and variability	default
ExtremeObs	extreme observations	default
ExtremeValues	extreme values	NEXTRVAL=
Frequencies	frequencies	FREQ
LocationCounts	counts used for sign test and signed rank test	LOCCOUNTS
MissingValues	missing values	default
Modes	modes	MODES
Moments	sample moments	default
Quantiles	quantiles	default
RobustScale	robust measures of scale	ROBUSTSCALE
TestsForLocation	tests for location	default
TestsForNormality	tests for normality	NORMALTEST
TrimmedMeans	trimmed means	TRIMMED=
WinsorizedMeans	Winsorized means	WINSORIZED=

Table 6.6 summarizes the ODS tables related to capability indices that you can request with options in the PROC CAPABILITY statement when you provide specification limits with a SPEC statement or with a SPEC= data set.

Table 6.6 ODS Tables Related to Specification Limits

Table Name	Description	Option
CIProbExSpecs	confidence limits for probabilities of exceeding specifications	CIPROBEX
Indices	standard capability indices	default
SpecialIndices	specialized capability indices	SPECIALINDICES
Specifications	percents outside specification limits based on empirical	default

Table 6.7 summarizes the ODS tables related to fitted distributions that you can request with options in the HISTOGRAM statement.

Table 6.7 ODS Tables Produced with the HISTOGRAM Statement

Table Name	Description	Option
Bins	histogram bins	MIDPERCENTS suboption with any distribution option, such as NORMAL(MIDPERCENTS)
FitIndices	capability indices computed from fitted distribution	INDICES suboption with any distribution option, such as LOGNORMAL(INDICES)
FitQuantiles	quantiles of fitted distribution	any distribution option such as NORMAL
GoodnessOfFit	goodness-of-fit tests for fitted distribution	any distribution option such as NORMAL
ParameterEstimates	parameter estimates for fitted distribution	any distribution option such as NORMAL
Specifications	percents outside specification limits based on empirical and fitted distributions	any distribution option such as NORMAL

The following table summarizes the ODS tables that you can request with options in the INTERVALS statement.

Table 6.8 ODS Tables Produced with the INTERVALS Statement

Table Name	Description	Option
Intervals1	prediction interval for future observations	METHODS=1
Intervals2	prediction interval for mean	METHODS=2
Intervals3	tolerance interval for proportion of population	METHODS=3
Intervals4	confidence limits for mean	METHODS=4
Intervals5	prediction interval for standard deviation	METHODS=5
Intervals6	confidence limits for standard deviation	METHODS=6

Examples: CAPABILITY Procedure

This section provides a more advanced example of the PROC CAPABILITY statement.

Example 6.1: Reading Specification Limits

NOTE: See *Reading Spec Limits from an Input Data Set* in the SAS/QC Sample Library.

You can specify specification limits either in the SPEC statement or in a SPEC= data set. In "Computing Capability Indices" on page 199, limits were specified in a SPEC statement. This example illustrates how

to create a SPEC= data set to read specification limits with the SPEC= option in the PROC CAPABILITY statement.

Consider the drink can data presented in "Computing Descriptive Statistics" on page 197. Suppose, in addition to the fluid weight of each drink can, the weight of the can itself is stored in a variable named Cweight, and both variables are saved in a data set called Can2. A partial listing of Can2 follows:

```
proc print data=Can2(obs=5);
run;
```

Output 6.1.1 The Data Set Can2

Process Capability Analysis of Fluid Weight

Obs	Weight	Cweight
1	12.07	1.07
2	12.02	0.86
3	12.00	1.06
4	12.01	1.08
5	11.98	1.02

The following DATA step creates a data set named Limits containing specification limits for the fluid weight and the can weight. Limits has 4 variables (_VAR_, _LSL_, _USL_, and _TARGET_) and 2 observations. The first observation contains the specification limit information for the variable Weight, and the second contains the specification limit information for the variable Cweight.

```
data Limits;
   length _var_ $8;
   _var_    = 'Weight';
   _lsl_    = 11.95;
   _target_ = 12;
   _usl_    = 12.05;
   output;
   _var_    = 'Cweight';
   _lsl_    = 0.90;
   _target_ = 1;
   _usl_    = 1.10;
   output;
run;
```

The following statements read the specification information from the Limits data set into the CAPABILITY procedure by using the SPEC= option. These statements print summary statistics, capability indices, and specification limit information for Weight and Cweight. Figure 6.1 and Figure 6.2 display the output for Weight. Output 6.1.2 displays the output for Cweight.

```
title 'Process Capability Analysis of Drink Can Data';
proc capability data=Can2 specs=Limits;
   var Cweight;
run;
```

Output 6.1.2 Printed Output for Variable Cweight

Process Capability Analysis of Drink Can Data

The CAPABILITY Procedure
Variable: Cweight (Can Weight (ounces))

Moments			
N	100	**Sum Weights**	100
Mean	1.004	**Sum Observations**	100.4
Std Deviation	0.06330941	**Variance**	0.00400808
Skewness	-0.074821	**Kurtosis**	-0.5433858
Uncorrected SS	101.1984	**Corrected SS**	0.3968
Coeff Variation	6.30571767	**Std Error Mean**	0.00633094

Basic Statistical Measures			
Location		**Variability**	
Mean	1.004000	**Std Deviation**	0.06331
Median	1.000000	**Variance**	0.00401
Mode	1.040000	**Range**	0.29000
		Interquartile Range	0.08500

Note: The mode displayed is the smallest of 2 modes with a count of 8.

Tests for Location: Mu0=0				
Test		**Statistic**	**p Value**	
Student's t	**t**	158.5862	**Pr > |t|**	<.0001
Sign	**M**	50	**Pr >= |M|**	<.0001
Signed Rank	**S**	2525	**Pr >= |S|**	<.0001

Tests for Normality				
Test		**Statistic**	**p Value**	
Shapiro-Wilk	**W**	0.987310	**Pr < W**	0.4588
Kolmogorov-Smirnov	**D**	0.061410	**Pr > D**	>0.1500
Cramer-von Mises	**W-Sq**	0.048175	**Pr > W-Sq**	>0.2500
Anderson-Darling	**A-Sq**	0.361939	**Pr > A-Sq**	>0.2500

Quantiles (Definition 5)	
Level	**Quantile**
100% Max	1.150
99%	1.140
95%	1.105
90%	1.080
75% Q3	1.045
50% Median	1.000
25% Q1	0.960
10%	0.910
5%	0.900
1%	0.870
0% Min	0.860

Output 6.1.2 *continued*

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0.86	2	1.11	42
0.88	89	1.12	28
0.88	64	1.12	34
0.90	68	1.13	48
0.90	59	1.15	52

Specification Limits			
Limit		Percent	
Lower (LSL)	0.900000	% < LSL	3.00000
Target	1.000000	% Between	92.00000
Upper (USL)	1.100000	% > USL	5.00000

Process Capability Indices			
Index	Value	95% Confidence Limits	
Cp	0.526515	0.453237	0.599670
CPL	0.547575	0.446607	0.647299
CPU	0.505454	0.408856	0.600808
Cpk	0.505454	0.409407	0.601501
Cpm	0.525467	0.454973	0.601113

Example 6.2: Enhancing Reference Lines

NOTE: See *Controlling the Appearance of Spec Limits* in the SAS/QC Sample Library.

A telecommunications company manufactures amplifiers to be used in telephones. Each amplifier is designed to boost the input signal by 5 decibels (dB). Because it is difficult to make every amplifier's boosting power exactly 5 decibels, the company decides that amplifiers that boost the input signal between 4 and 6 decibels are acceptable. Therefore, the target value is 5 decibels, and the lower and upper specification limits are 4 and 6 decibels, respectively. The following data set contains the boosting powers of a sample of 75 amplifiers:

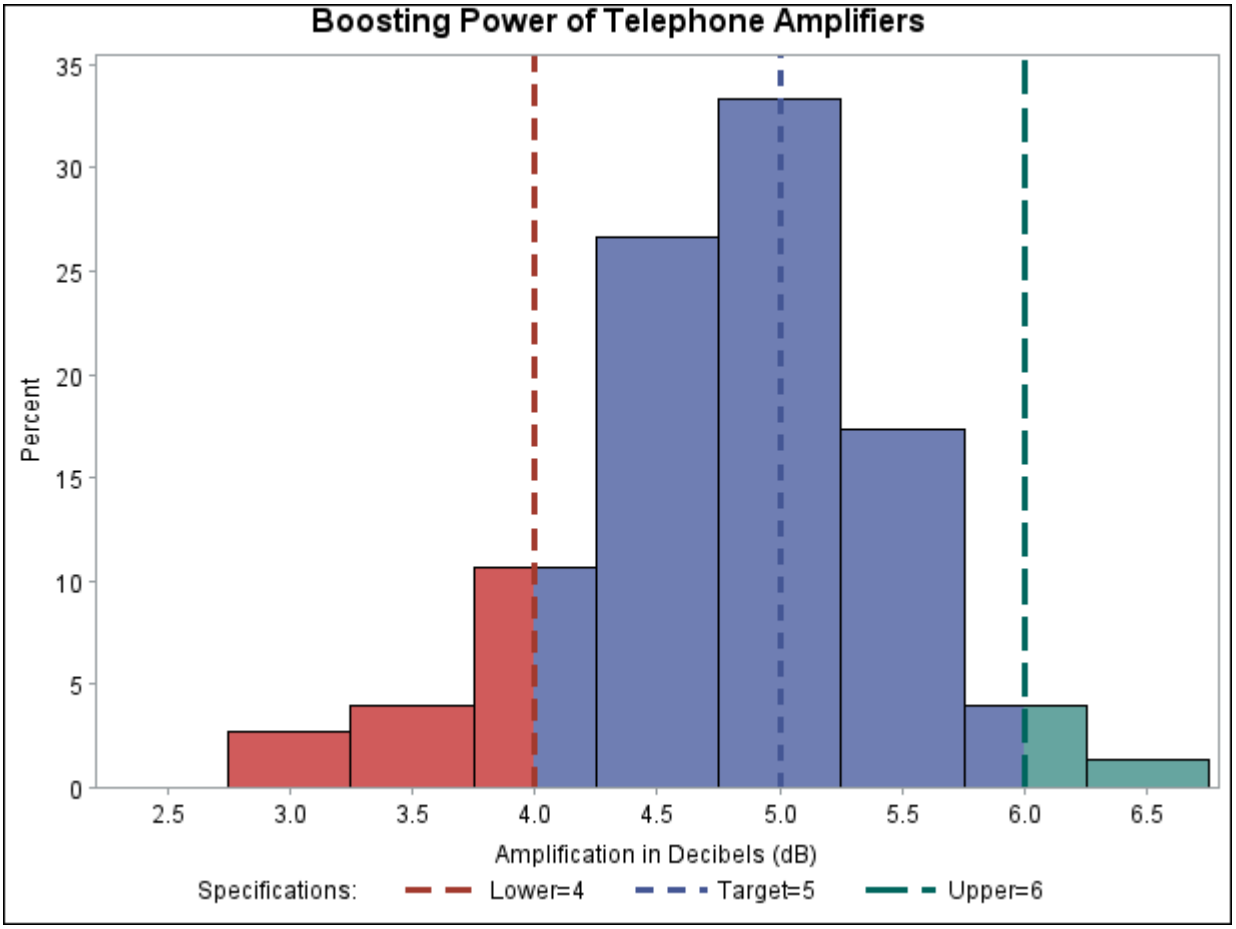
```
data Amps;
   label Decibels = 'Amplification in Decibels (dB)';
   input Decibels @@;
   datalines;
4.54 4.87 4.66 4.90 4.68 5.22 4.43 5.14 3.07 4.22
5.09 3.41 5.75 5.16 3.96 5.37 5.70 4.11 4.83 4.51
4.57 4.16 5.73 3.64 5.48 4.95 4.57 4.46 4.75 5.38
5.19 4.35 4.98 4.87 3.53 4.46 4.57 4.69 5.27 4.67
5.03 4.50 5.35 4.55 4.05 6.63 5.32 5.24 5.73 5.08
5.07 5.42 5.05 5.70 4.79 4.34 5.06 4.64 4.82 3.24
4.79 4.46 3.84 5.05 5.46 4.64 6.13 4.31 4.81 4.98
4.95 5.57 4.11 4.15 5.95
;
```

The SPEC statement provides several options to control the appearance of reference lines for the specification limits and the target value. The following statements use the data set Amps to create a histogram that demonstrates some of these options:

```
ods graphics off;
legend2 FRAME CFRAME=ligr CBORDER=black POSITION=center;
title 'Boosting Power of Telephone Amplifiers';
proc capability data=Amps;
   spec  target = 5      lsl = 4         usl = 6
         ltarget = 2     llsl = 3        lusl = 4
         wtarget = 2     wlsl = 2        wusl = 2
         cleft           cright;
   histogram Decibels / cbarline = black;
run;
```

The resulting histogram is shown in Output 6.2.1. The LTARGET=, LLSL=, and LUSL= options control the line type of the reference lines for the target, lower specification limit, and upper specification limit, respectively. Likewise, the WTARGET=, WLSL=, and WUSL= options control the line widths. The CLEFT= option controls the color used to fill the area to the left of the lower specification limit. Similarly, the CRIGHT= option controls the color used to fill the area to the right of the upper specification limit.

Output 6.2.1 Controlling the Appearance of Specification Limits



Example 6.3: Displaying a Confidence Interval for Cpk

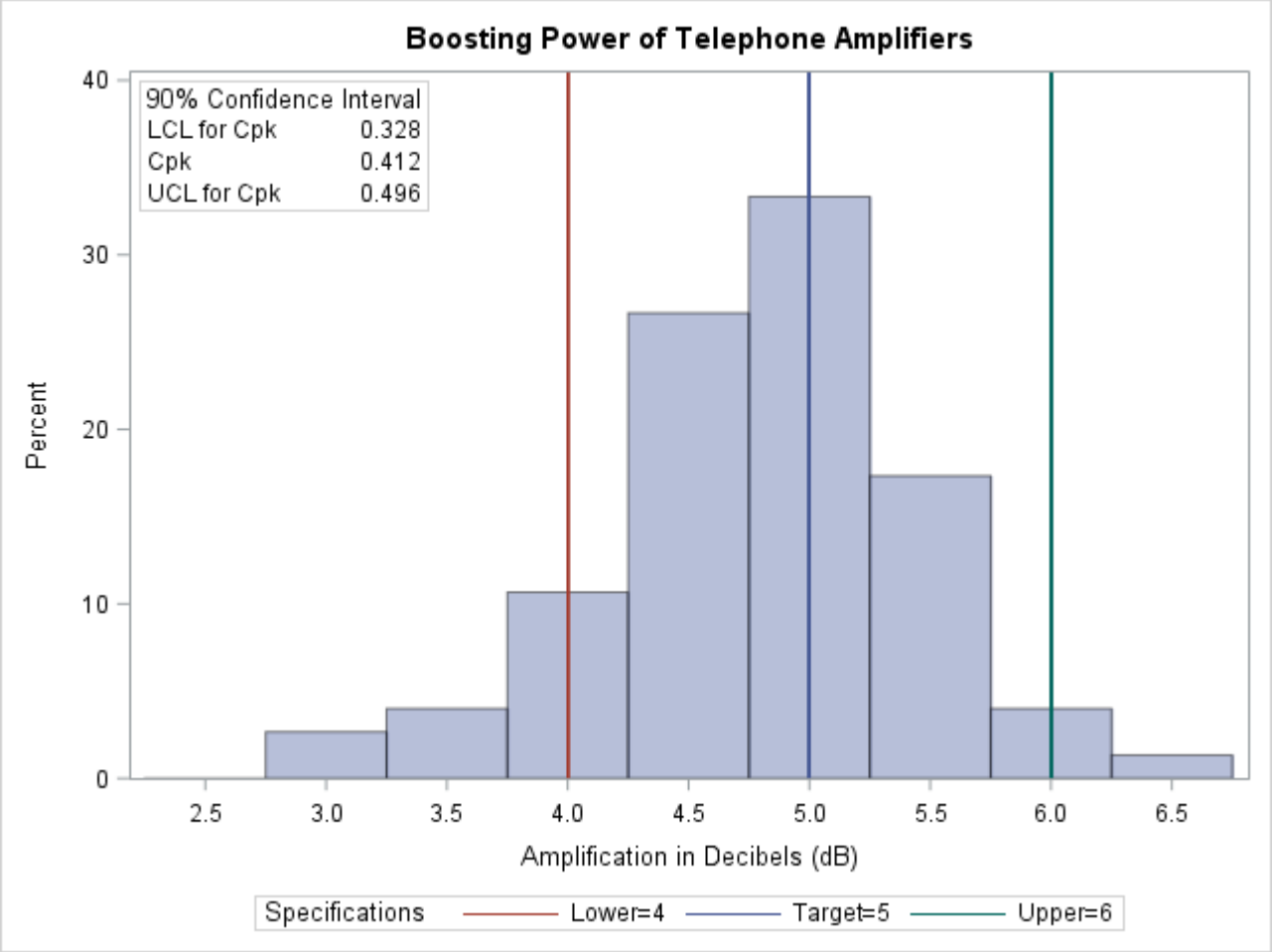
NOTE: See *Displaying a Confidence Interval for Cpm* in the SAS/QC Sample Library.

In this example, the capability index C_{pk} is computed for the amplification data in Amps. To examine the accuracy of this estimate, the following statements calculate a 90% confidence interval for C_{pk}, then display the interval on a histogram (shown in Output 6.3.1) with the INSET statement:

```
title 'Boosting Power of Telephone Amplifiers';
proc capability data=Amps noprint alpha=0.10;
   var Decibels;
   spec  target = 5  lsl = 4  usl = 6
        ltarget = 2 llsl = 3 lusl = 4;
   histogram Decibels / odstitle = title;;
   inset cpklcl cpk cpkucl / header = '90% Confidence Interval'
                             format = 6.3;
run;
```

The ALPHA= option in the PROC CAPABILITY statement controls the level of the confidence interval. In this case, the 90% confidence interval on C_{pk} is wide (from 0.328 to 0.496), indicating that the process may need adjustments in order to improve process variability. Confidence limits for capability indices can be displayed using the INSET statement (as shown in Output 6.3.1) or saved in an output data set by using the OUTPUT statement. For formulas and details about capability indices, see the section "Specialized Capability Indices" on page 239. For more information about the INSET statement, see "INSET Statement: CAPABILITY Procedure" on page 385.

Output 6.3.1 Confidence Interval on C_{pk}



The following statements can be used to produce a table of process capability indices including the index C_{pk}:

```
ods select indices;
proc capability data=Amps alpha=0.10;
   spec  target = 5  lsl = 4  usl = 6
        ltarget = 2 llsl = 3 lusl = 4;
   var Decibels;
run;
```

Output 6.3.2 Process Capability Indices

Boosting Power of Telephone Amplifiers

The CAPABILITY Procedure
Variable: Decibels (Amplification in Decibels (dB))

Process Capability Indices			
Index	**Value**	**90% Confidence Limits**	
Cp	0.508962	0.439538	0.576922
CPL	0.411920	0.326620	0.495136
CPU	0.606004	0.501261	0.708127
Cpk	0.411920	0.327599	0.496241
Cpm	0.488674	0.425292	0.556732

CDFPLOT Statement: CAPABILITY Procedure

Overview: CDFPLOT Statement

The CDFPLOT statement plots the observed cumulative distribution function (*cdf*) of a variable, defined as

$$\begin{aligned} F_N(x) &= \text{percent of nonmissing values} \leq x \\ &= \frac{\text{number of values} \leq x}{N} \times 100\% \end{aligned}$$

where N is the number of nonmissing observations. The cdf is an increasing step function that has a vertical jump of $\frac{1}{N}$ at each value of x equal to an observed value. The cdf is also referred to as the empirical cumulative distribution function (ecdf).

You can use options in the CDFPLOT statement to do the following:

- superimpose specification limits
- superimpose fitted theoretical distributions
- specify graphical enhancements (such as color or text height)

You can also create a comparative cdf plot by using the CDFPLOT statement in conjunction with a CLASS statement.

You have three alternatives for producing cdf plots with the CDFPLOT statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: CDFPLOT Statement

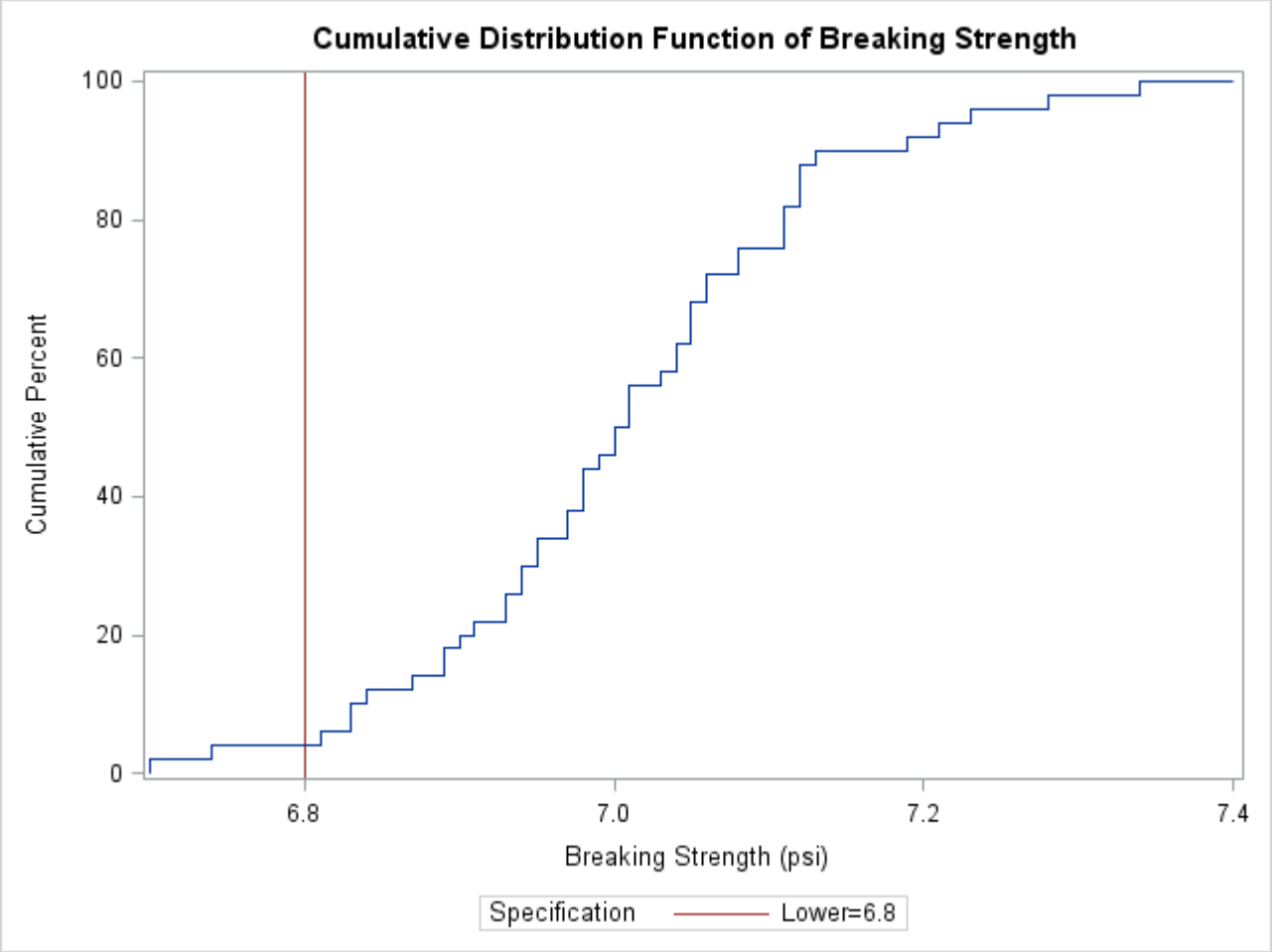
Creating a Cumulative Distribution Plot

NOTE: See *CDF Plot with Superimposed Normal Curve* in the SAS/QC Sample Library.

This section introduces the CDFPLOT statement with a simple example. A company that produces fiber optic cord is interested in the breaking strength of the cord. The following statements create a data set named Cord, which contains 50 breaking strengths measured in pounds per square inch (psi), and they display the cdf plot in Figure 6.5. The plot shows a symmetric distribution with observations concentrated 6.9 and 7.1. The plot also shows that only a small percentage (< 5%) of the observations are below the lower specification limit of 6.8.

```
data Cord;
   label Strength="Breaking Strength (psi)";
   input Strength @@;
   datalines;
6.94 6.97 7.11 6.95 7.12 6.70 7.13 7.34 6.90 6.83
7.06 6.89 7.28 6.93 7.05 7.00 7.04 7.21 7.08 7.01
7.05 7.11 7.03 6.98 7.04 7.08 6.87 6.81 7.11 6.74
6.95 7.05 6.98 6.94 7.06 7.12 7.19 7.12 7.01 6.84
6.91 6.89 7.23 6.98 6.93 6.83 6.99 7.00 6.97 7.01
;

title 'Cumulative Distribution Function of Breaking Strength';
proc capability data=Cord noprint;
   spec lsl=6.8;
   cdf Strength / odstitle=title;
run;
```

Figure 6.5 Cumulative Distribution Function



Syntax: CDFPLOT Statement

The syntax for the CDFPLOT statement is as follows:

CDFPLOT < *variables* > < / *options* > **;**

You can specify the keyword CDF as an alias for CDFPLOT. You can specify any number of CDFPLOT statements after a PROC CAPABILITY statement. The components of the CDFPLOT statement are described as follows:

variables

specify variables for which to create cdf plots. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, the variables can be any numeric variables in the input data set. If you do not specify variables in a CDFPLOT statement, then a cdf plot is created for each variable listed in the VAR statement, or for each numeric variable in the input data set if you do not use a VAR statement.

For example, suppose a data set named steel contains exactly three numeric variables, length, width and height. The following statements create a cdf plot for each of the three variables:

```
proc capability data=steel;
   cdfplot;
run;
```

The following statements create a cdf plot for length and a cdf plot for width:

```
proc capability data=steel;
   var length width;
   cdfplot;
run;
```

The following statements create a cdf plot for width:

```
proc capability data=steel;
   var length width;
   cdfplot width;
run;
```

By default, the horizontal axis of a cdf plot is labeled with the variable name. If you specify a label for a variable, however, the label is used. The default vertical axis label is *Cumulative Percent*, and the axis is scaled in percent of observations.

If you specify a SPEC statement or a SPEC= data set in addition to the CDFPLOT statement, then the specification limits for each variable are displayed as reference lines and are identified in a legend.

options

add features to plots. All options appear after the slash (/) in the CDFPLOT statement. In the following example, the NORMAL option superimposes a normal cdf on the plot, and the CTEXT= option specifies the color of the text.

```
proc capability data=steel;
   cdfplot length / normal ctext=yellow;
run;
```

Summary of Options

The following tables list all options by function. The section "Dictionary of Options" on page 263 describes each option in detail.

Distribution Options

You can use the options listed in Table 6.9 to superimpose a fitted theoretical distribution function on your cdf plot.

Table 6.9 Options for Specifying a Theoretical Distribution

Option	Description
BETA(*beta-options*)	plots beta distribution with threshold parameter θ, scale parameter σ, and shape parameters α and β
EXPONENTIAL(*exponential-options*)	plots exponential distribution with threshold parameter θ and scale parameter σ
GAMMA(*gamma-options*)	plots gamma distribution with threshold parameter θ, scale parameter σ, and shape parameter α
GUMBEL(*Gumbel-options*)	plots Gumbel distribution with location parameter μ and scale parameter σ
IGAUSS(*iGauss-options*)	plots inverse Gaussian distribution with mean μ and shape parameter λ
LOGNORMAL(*lognormal-options*)	plots lognormal distribution with threshold parameter θ, scale parameter ζ, and shape parameter σ,
NORMAL(*normal-options*)	plots normal distribution with mean μ and standard deviation σ
PARETO(*Pareto-options*)	plots generalized Pareto distribution with threshold parameter θ, scale parameter σ, and shape parameter α
POWER(*power-options*)	plots power function distribution with threshold parameter θ, scale parameter σ, and shape parameter α
RAYLEIGH(*Rayleigh-options*)	plots Rayleigh distribution with threshold parameter θ and scale parameter σ
WEIBULL(*Weibull-options*)	plots Weibull distribution function with threshold parameter θ, scale parameter σ, and shape parameter c

Table 6.10 summarizes options that specify distribution parameters and control the display of the theoretical distribution curve. You can specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to superimpose a normal distribution:

```
proc capability;
   cdfplot / normal(mu=10 sigma=0.5 color=red);
run;
```

The COLOR= option specifies the color for the curve, and the *normal-options* MU= and SIGMA= specify the parameters $\mu = 10$ and $\sigma = 0.5$ for the distribution function. If you do not specify these parameters, maximum likelihood estimates are computed.

Table 6.10 Distribution Options

Option	Description
Options Used with All Distributions	
COLOR=	specifies color of theoretical distribution function
L=	specifies line type of theoretical distribution function
SYMBOL=	specifies *character* used to plot theoretical distribution function on line printer plots
W=	specifies width of theoretical distribution function
Beta-Options	
ALPHA=	specifies first shape parameter α for beta distribution function
BETA=	specifies second shape parameter β for beta distribution function
SIGMA=	specifies scale parameter σ for beta distribution function
THETA=	specifies lower threshold parameter θ for beta distribution function
Exponential-Options	
SIGMA=	specifies scale parameter σ for exponential distribution function
THETA=	specifies threshold parameter θ for exponential distribution function
Gamma-Options	
ALPHA=	specifies shape parameter α for gamma distribution function
ALPHADELTA=	specifies change in successive estimates of α at which the Newton-Raphson approximation of $\hat{\alpha}$ terminates
ALPHAINITIAL=	specifies initial value for α in the Newton-Raphson approximation of $\hat{\alpha}$
MAXITER=	specifies maximum number of iterations in the Newton-Raphson approximation of $\hat{\alpha}$
SIGMA=	specifies scale parameter σ for gamma distribution function
THETA=	specifies threshold parameter θ for gamma distribution function
Gumbel-Options	
MU=	specifies location parameter μ for Gumbel distribution function
SIGMA=	specifies scale parameter σ for Gumbel distribution function
IGauss-Options	
LAMBDA=	specifies shape parameter λ for inverse Gaussian distribution function
MU=	specifies mean μ for inverse Gaussian distribution function
Lognormal-Options	
SIGMA=	specifies shape parameter σ for lognormal distribution function
THETA=	specifies threshold parameter θ for lognormal distribution function
ZETA=	specifies scale parameter ζ for lognormal distribution function
Normal-Options	
MU=	specifies mean μ for normal distribution function
SIGMA=	specifies standard deviation σ for normal distribution function
Pareto-Options	
ALPHA=	specifies shape parameter α for generalized Pareto distribution function
SIGMA=	specifies scale parameter σ for generalized Pareto distribution function

Table 6.10 (*continued*)

Option	Description
THETA=	specifies threshold parameter θ for generalized Pareto distribution function
Power-Options	
ALPHA=	specifies shape parameter α for power function distribution
SIGMA=	specifies scale parameter σ for power function distribution
THETA=	specifies threshold parameter θ for power function distribution
Rayleigh-Options	
SIGMA=	specifies scale parameter σ for Rayleigh distribution function
THETA=	specifies threshold parameter θ for Rayleigh distribution function
Weibull-Options	
C=	specifies shape parameter c for Weibull distribution function
CDELTA=	specifies change in successive estimates of c at which the Newton-Raphson approximation of $\hat{c}$ terminates
CINITIAL=	specifies initial value for c in the Newton-Raphson approximation of $\hat{c}$
MAXITER=	specifies maximum number of iterations in the Newton-Raphson approximation of $\hat{c}$
SIGMA=	specifies scale parameter σ for Weibull distribution function
THETA=	specifies threshold parameter θ for Weibull distribution function

General Options

Table 6.11 General CDFPLOT Statement Options

Option	Description
General Plot Layout Options	
CONTENTS=	specifies table of contents entry for cdf plot grouping
HREF=	specifies reference lines perpendicular to the horizontal axis
HREFLABELS=	specifies labels for HREF= lines
NOCDFLEGEND	suppresses legend for superimposed theoretical cdf
NOECDF	suppresses plot of empirical (observed) distribution function
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOSPECLEGEND	suppresses specifications legend
VREF=	specifies reference lines perpendicular to the vertical axis
VREFLABELS=	specifies labels for VREF= lines
VSCALE=	specifies scale for vertical axis
Graphics Options	
ANNOTATE=	specifies annotate data set
CAXIS=	specifies color for axis
CFRAME=	specifies color for frame
CHREF=	specifies colors for HREF= lines
CSTATREF=	specifies colors for STATREF= lines

Table 6.11 (*continued*)

Option	Description
CTEXT=	specifies color for text
CVREF=	specifies colors for VREF= lines
DESCRIPTION=	specifies description for graphics catalog member
FONT=	specifies text font
HAXIS=	specifies AXIS statement for horizontal axis
HEIGHT=	specifies height of text used outside framed areas
HMINOR=	specifies number of horizontal axis minor tick marks
HREFLABPOS=	specifies position for HREF= line labels
INFONT=	specifies software font for text inside framed areas
INHEIGHT=	specifies height of text inside framed areas
LHREF=	specifies line styles for HREF= lines
LSTATREF=	specifies line styles for STATREF= lines
LVREF=	specifies line styles for VREF= lines
NAME=	specifies name for plot in graphics catalog
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
STATREF=	specifies reference lines at values of summary statistics
STATREFLABELS=	specifies labels for STATREF= lines
STATREFSUBCHAR=	specifies substitution character for displaying statistic values in STATREFLABELS= labels
TURNVLABELS	turns and vertically strings out characters in labels for vertical axis
VAXIS=	specifies AXIS statement for vertical axis
VAXISLABEL=	specifies label for vertical axis
VMINOR=	specifies number of vertical axis minor tick marks
VREFLABPOS=	specifies position for VREF= line labels
WAXIS=	specifies line thickness for axes and frame
Options for ODS Graphics Output	
ODSFOOTNOTE=	specifies footnote displayed on cdf plot
ODSFOOTNOTE2=	specifies secondary footnote displayed on cdf plot
ODSTITLE=	specifies title displayed on cdf plot
ODSTITLE2=	specifies secondary title displayed on cdf plot
Options for Comparative Plots	
ANNOKEY	applies annotation requested in ANNOTATE= data set to key cell only
CFRAMESIDE=	specifies color for filling row label frames
CFRAMETOP=	specifies color for filling column label frames
CPROP=	specifies color for proportion of frequency bar
CTEXTSIDE=	specifies color for row labels
CTEXTTOP=	specifies color for column labels
INTERTILE=	specifies distance between tiles in comparative plot
NCOLS=	specifies number of columns in comparative plot
NROWS=	specifies number of rows in comparative plot
OVERLAY	overlays plots for different class levels (ODS Graphics only)

Table 6.11 (*continued*)

Option	Description
Options for Line Printer Charts	
CDFSYMBOL=	specifies character for plotted points
HREFCHAR=	specifies line character for HREF= lines
VREFCHAR=	specifies line character for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the options specific to the CDFPLOT statement. See "Dictionary of Common Options: CAPABILITY Procedure" on page 534 for detailed descriptions of options common to all the plot statements.

ALPHA=*value*

specifies the shape parameter α for distribution functions requested with the BETA, GAMMA, PARETO, and POWER options. Enclose the ALPHA= option in parentheses after the distribution keyword. If you do not specify a value for α, the procedure calculates a maximum likelihood estimate. For examples, see the entries for the distribution options.

BETA<(*beta-options***)>**

displays a fitted beta distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ I_{\frac{x-\theta}{\sigma}}(\alpha, \beta) & \text{for } \theta < x < \theta + \sigma \\ 1 & \text{for } x \ge \sigma + \theta \end{cases}$$

where $I_y(\alpha, \beta)$ is the incomplete beta function, and

θ = lower threshold parameter (lower endpoint)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
β = shape parameter ($\beta > 0$)

The beta distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. You can specify θ and σ by using the THETA= and SIGMA= *beta-options*, as illustrated in the following statements, which fit a beta distribution bounded between 50 and 75. The default values for θ and σ are 0 and 1, respectively.

```
proc capability;
   cdfplot / beta(theta=50 sigma=25);
run;
```

The beta distribution has two shape parameters, α and β. If these parameters are known, you can specify their values with the ALPHA= and BETA= *beta-options*. If you do not specify values for α and β, the procedure calculates maximum likelihood estimates.

The BETA option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the BETA distribution option.

BETA=*value*

B=*value*

specifies the second shape parameter β for beta distribution functions requested by the BETA option. Enclose the BETA= option in parentheses after the BETA keyword. If you do not specify a value for β, the procedure calculates a maximum likelihood estimate. For examples, see the preceding entry for the BETA option.

C=*value*

specifies the shape parameter c for Weibull distribution functions requested with the WEIBULL option. Enclose the C= option in parentheses after the WEIBULL keyword. If you do not specify a value for c, the procedure calculates a maximum likelihood estimate. You can specify the SHAPE= option as an alias for the C= option.

CDFSYMBOL='*character*'

specifies the character used to plot the points on legacy line printer cdf plots. The default is the plus sign (+). This option is ignored unless you specify the LINEPRINTER option in the PROC CAPABILITY statement. Use the SYMBOL statement to control the plotting symbol in traditional graphics output.

EXPONENTIAL<(*exponential-options***)>**

EXP<(*exponential-options* **)>**

displays a fitted exponential distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ 1 - \exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x > \theta \end{cases}$$

where
θ = threshold parameter
σ = scale parameter ($\sigma > 0$)

The parameter θ must be less than or equal to the minimum data value. You can specify θ with the THETA= *exponential-option*. The default value for θ is 0. You can specify σ with the SIGMA= *exponential-option*. By default, a maximum likelihood estimate is computed for σ. For example, the following statements fit an exponential distribution with $\theta = 10$ and a maximum likelihood estimate for σ:

```
proc capability;
   cdfplot / exponential(theta=10 l=2 color=green);
run;
```

The exponential curve is green and has a line type of 2.

The EXPONENTIAL option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the EXPONENTIAL option.

GAMMA<(*gamma-options***)>**

displays a fitted gamma distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ \frac{1}{\Gamma(\alpha)\sigma} \int_\theta^x \left(\frac{t-\theta}{\sigma}\right)^{\alpha-1} \exp\left(-\frac{t-\theta}{\sigma}\right) dt & \text{for } x > \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

The parameter θ for the gamma distribution must be less than the minimum data value. You can specify θ with the THETA= *gamma-option*. The default value for θ is 0. In addition, the gamma distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *gamma-options*. By default, maximum likelihood estimates are computed for α and σ. For example, the following statements fit a gamma distribution function with $\theta = 4$ and maximum likelihood estimates for α and σ:

```
proc capability;
   cdfplot / gamma(theta=4);
run;
```

Note that the maximum likelihood estimate of α is calculated iteratively using the Newton-Raphson approximation. The *gamma-options* ALPHADELTA=, ALPHAINITIAL=, and MAXITER= control the approximation.

The GAMMA option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the GAMMA option.

GUMBEL< (*Gumbel-options*) >
displays a fitted Gumbel distribution (also known as Type 1 extreme value distribution) function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \exp\left(-e^{-(x-\mu)/\sigma}\right)$$

where

μ = location parameter
σ = scale parameter ($\sigma > 0$)

You can specify known values for μ and σ with the MU= and SIGMA= *Gumbel-options*. By default, maximum likelihood estimates are computed for μ and σ.

The GUMBEL option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the GUMBEL option.

IGAUSS< (*iGauss-options*) >
displays a fitted inverse Gaussian distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \Phi\left\{\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu} - 1\right)\right\} + e^{2\lambda/\mu}\Phi\left\{-\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu} + 1\right)\right\}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

μ = mean parameter ($\mu > 0$)
λ = shape parameter ($\lambda > 0$)

You can specify known values for μ and λ with the MU= and LAMBDA= *iGauss-options*. By default, maximum likelihood estimates are computed for μ and λ.

The IGAUSS option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the IGAUSS option.

LAMBDA=*value*

specifies the shape parameter λ for distribution functions requested with the IGAUSS option. Enclose the LAMBDA= option in parentheses after the IGAUSS distribution keyword. If you do not specify a value for λ, the procedure calculates a maximum likelihood estimate.

LEGEND=*name* | **NONE**

specifies the name of a LEGEND statement describing the legend for specification limit reference lines and superimposed distribution functions. Specifying LEGEND=NONE, which suppresses all legend information, is equivalent to specifying the NOLEGEND option. This option is ignored unless you are producing traditional graphics.

LOGNORMAL<(*lognormal-options***)>**

displays a fitted lognormal distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ \Phi\left(\frac{\log(x-\theta)-\zeta}{\sigma}\right) & \text{for } x > \theta \end{cases}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

θ = threshold parameter
ζ = scale parameter
σ = shape parameter ($\sigma > 0$)

The parameter θ for the lognormal distribution must be less than the minimum data value. You can specify θ with the THETA= *lognormal-option*. The default value for θ is 0. In addition, the lognormal distribution has a shape parameter σ and a scale parameter ζ. You can specify these parameters with the SIGMA= and ZETA= *lognormal-options*. By default, maximum likelihood estimates are computed for σ and ζ. For example, the following statements fit a lognormal distribution function with $\theta = 10$ and maximum likelihood estimates for σ and ζ:

```
proc capability;
   cdfplot / lognormal(theta = 10);
run;
```

The LOGNORMAL option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the LOGNORMAL option.

MU=*value*

specifies the parameter μ for distribution functions requested with the GUMBEL, IGAUSS, and NORMAL options. Enclose the MU= option in parentheses after the distribution keyword. For the normal and inverse Gaussian distributions, the default value of μ is the sample mean. If you do not specify a value for μ for the Gumbel distribution, the procedure calculates a maximum likelihood estimate.

NOCDFLEGEND
suppresses the legend for the superimposed theoretical cumulative distribution function.

NOECDF
suppresses the observed distribution function (the empirical cumulative distribution function) of the variable, which is drawn by default. This option enables you to create theoretical cdf plots without displaying the data distribution. The NOECDF option can be used only with a theoretical distribution (such as the NORMAL option).

NOLEGEND
suppresses legends for specification limits, theoretical distribution functions, and hidden observations. Specifying the NOLEGEND option is equivalent to specifying LEGEND=NONE.

NORMAL< (*normal-options*) >
displays a fitted normal distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) \quad \text{for } -\infty < x < \infty$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

μ = mean
σ = standard deviation ($\sigma > 0$)

You can specify known values for μ and σ with the MU= and SIGMA= *normal-options*, as shown in the following statements:

```
proc capability;
   cdfplot / normal(mu=14 sigma=.05);
run;
```

By default, the sample mean and sample standard deviation are calculated for μ and σ. The NORMAL option can appear only once in a CDFPLOT statement. For an example, see Output 6.4.1. See Table 6.10 for a list of secondary options you can specify with the NORMAL option.

NOSPECLEGEND
NOSPECL
suppresses the portion of the legend for specification limit reference lines.

PARETO< (*Pareto-options*) >
displays a fitted generalized Pareto distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = 1 - \left(1 - \frac{\alpha(x-\theta)}{\sigma}\right)^{\frac{1}{\alpha}}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter

The parameter θ for the generalized Pareto distribution must be less than the minimum data value. You can specify θ with the THETA= *Pareto-option*. The default value for θ is 0. In addition, the generalized Pareto distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *Pareto-options*. By default, maximum likelihood estimates are computed for α and σ.

The PARETO option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the PARETO option.

POWER< (*power-options*) >
displays a fitted power function distribution on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ \left(\frac{x-\theta}{\sigma}\right)^{\alpha} & \text{for } \theta < x < \theta + \sigma \\ 1 & \text{for } x \ge \theta + \sigma \end{cases}$$

where

θ = lower threshold parameter (lower endpoint)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

The power function distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. You can specify θ and σ by using the THETA= and SIGMA= *power-options*. The default values for θ and σ are 0 and 1, respectively.

You can specify a value for the shape parameter, α, with the ALPHA= *power-option*. If you do not specify a value for α, the procedure calculates a maximum likelihood estimate.

The power function distribution is a special case of the beta distribution with its second shape parameter, $\beta = 1$.

The POWER option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the POWER option.

RAYLEIGH< (*Rayleigh-options*) >
displays a fitted Rayleigh distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = 1 - e^{-(x-\theta)^2/(2\sigma^2)}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)

The parameter θ for the Rayleigh distribution must be less than the minimum data value. You can specify θ with the THETA= *Rayleigh-option*. The default value for θ is 0. You can specify σ with the SIGMA= *Rayleigh-option*. By default, a maximum likelihood estimate is computed for σ.

The RAYLEIGH option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the RAYLEIGH option.

SIGMA=*value*

specifies the parameter σ for distribution functions requested by the BETA, EXPONENTIAL, GAMMA, GUMBEL, LOGNORMAL, NORMAL, PARETO, POWER, RAYLEIGH, and WEIBULL options. Enclose the SIGMA= option in parentheses after the distribution keyword. The following table summarizes the use of the SIGMA= option:

Distribution Option	SIGMA= Specifies	Default Value	Alias
BETA POWER	scale parameter σ	1	SCALE=
EXPONENTIAL GAMMA WEIBULL	scale parameter σ	maximum likelihood estimate	SCALE=
GUMBEL PARETO RAYLEIGH	scale parameter σ	maximum likelihood estimate	
LOGNORMAL	shape parameter σ	maximum likelihood estimate	SHAPE=
NORMAL	scale parameter σ	standard deviation	

SYMBOL='*character***'**

specifies the *character* used to plot the theoretical distribution function on legacy line printer plots. Enclose the SYMBOL= option in parentheses after the distribution option. The default character is the first letter of the distribution option keyword. This option is ignored unless you specify the LINEPRINTER option in the PROC CAPABILITY statement.

THETA=*value*

THRESHOLD=*value*

specifies the lower threshold parameter θ for theoretical cumulative distribution functions requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, PARETO, POWER, RAYLEIGH, and WEIBULL options. Enclose the THETA= option in parentheses after the distribution keyword. The default *value* is 0.

VSCALE=PERCENT | PROPORTION

specifies the scale of the vertical axis. The value PERCENT scales the data in units of percent of observations per data unit. The value PROPORTION scales the data in units of proportion of observations per data unit. The default is PERCENT.

WEIBULL< (*Weibull-options***) >**

displays a fitted Weibull distribution function on the cdf plot. The equation of the fitted cdf is

$$F(x) = \begin{cases} 0 & \text{for } x \le \theta \\ 1 - \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^c\right) & \text{for } x > \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)

The parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Weibull-option*. The default value for θ is 0. In addition, the Weibull distribution has a shape parameter c and a scale parameter σ. You can specify these parameters with the SIGMA= and C= *Weibull-options*. By default, maximum likelihood estimates are computed for c and σ. For example, the following statements fit a Weibull distribution function with $\theta = 15$ and maximum likelihood estimates for σ and c:

```
proc capability;
   cdfplot / weibull(theta=15);
run;
```

Note that the maximum likelihood estimate of c is calculated iteratively using the Newton-Raphson approximation. The *Weibull-options* CDELTA=, CINITIAL=, and MAXITER= control the approximation.

The WEIBULL option can appear only once in a CDFPLOT statement. See Table 6.10 for a list of secondary options you can specify with the WEIBULL option.

ZETA=*value*

specifies a value for the scale parameter ζ for a lognormal distribution function requested with the LOGNORMAL option. Enclose the ZETA= option in parentheses after the LOGNORMAL keyword. If you do not specify a *value* for ζ, a maximum likelihood estimate is computed. You can specify the SCALE= option as an alias for the ZETA= option.

Details: CDFPLOT Statement

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. CDFPLOT options used to control the appearance of traditional graphics are ignored for ODS Graphics output.

When ODS Graphics is in effect, the CDFPLOT statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 6.12.

Table 6.12 ODS Graphics Produced by the CDFPLOT Statement

ODS Graph Name	Plot Description
CDFPlot	cumulative distribution function plot

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Examples: CDFPLOT Statement

This section illustrates how to display a fitted distribution function, inset tables, and display reference lines on your cdf plot.

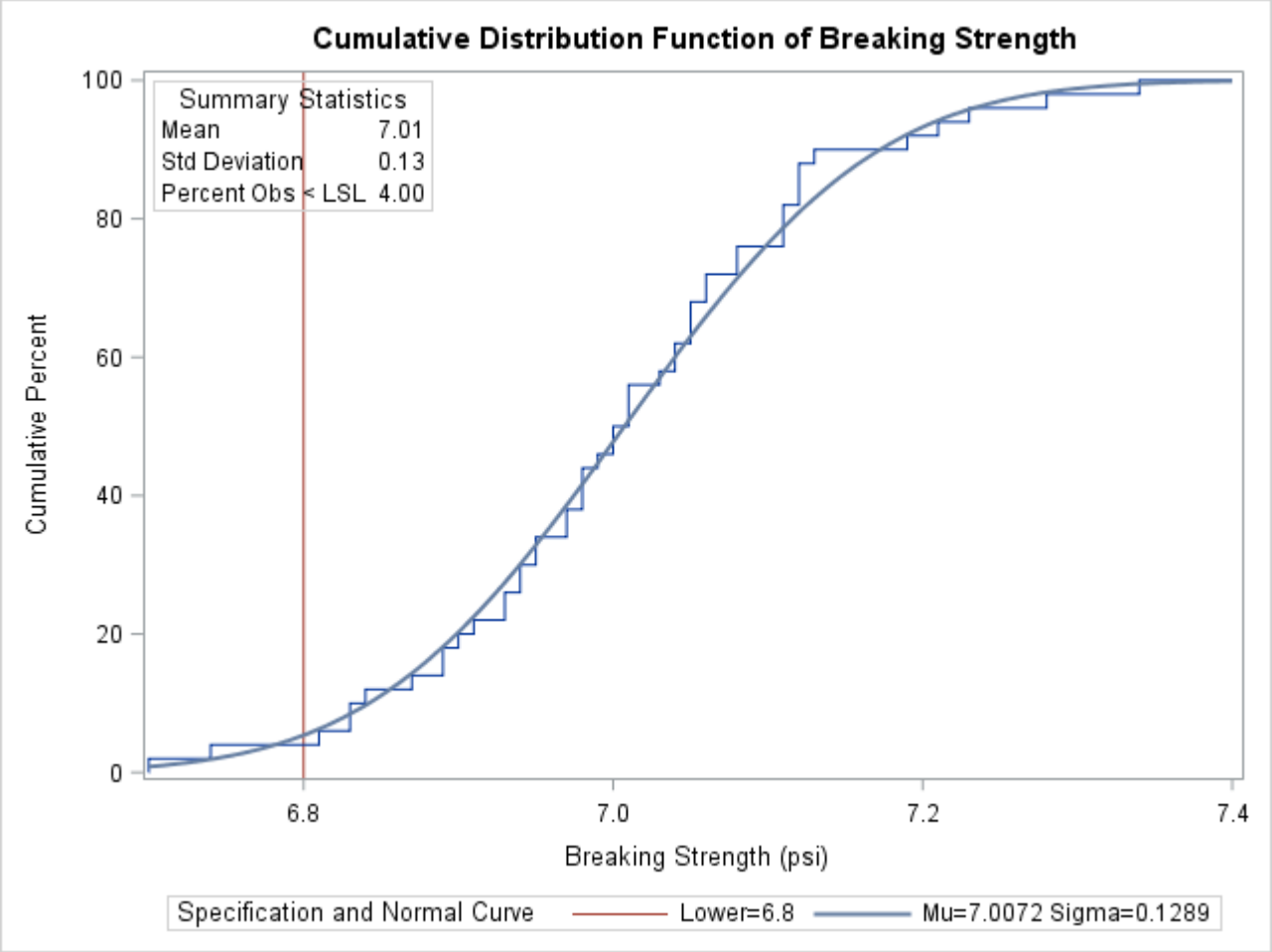
Example 6.4: Fitting a Normal Distribution

NOTE: See *CDF Plot with Superimposed Normal Curve* in the SAS/QC Sample Library.

You can use the CDFPLOT statement to fit any of eleven theoretical distributions (beta, exponential, gamma, Gumbel, inverse Gaussian, lognormal, normal, generalized Pareto, power function, Rayleigh, and Weibull) and superimpose them on the cdf plot. The following statements use the NORMAL option to display a fitted normal distribution function on a cdf plot of breaking strengths. The data set Cord is given in Figure 6.5, and the plot is shown in Output 6.4.1.

```
title 'Cumulative Distribution Function of Breaking Strength';
proc capability data=Cord noprint;
   spec lsl=6.8;
   cdf Strength / normal
                  odstitle=title;
   inset mean std pctlss / format = 5.2
                           header = "Summary Statistics";
run;
```

Output 6.4.1 Superimposed Normal Distribution Function



The NORMAL option requests the fitted curve. The INSET statement requests an inset containing the mean, the standard deviation, and the percent of observations below the lower specification limit. For more information about the INSET statement, see "INSET Statement: CAPABILITY Procedure" on page 385. The SPEC statement requests a lower specification limit at 6.8. For more information about the SPEC statement, see "SPEC Statement" on page 214.

The agreement between the empirical and the normal distribution functions in Output 6.4.1 is evidence that the normal distribution is an appropriate model for the distribution of breaking strengths.

The CAPABILITY procedure provides a variety of other tools for assessing goodness of fit. Goodness-of-fit tests (see "Printed Output" on page 348) provide a quantitative assessment of a proposed distribution. Probability and Q-Q plots, created with the PROBPLOT ("PROBPLOT Statement: CAPABILITY Procedure" on page 461), QQPLOT ("QQPLOT Statement: CAPABILITY Procedure" on page 493), and PPPLOT ("PPPLOT Statement: CAPABILITY Procedure" on page 439) statements, provide effective graphical diagnostics.

Example 6.5: Using Reference Lines with CDF Plots

NOTE: See *CDF Plot with Superimposed Normal Curve* in the SAS/QC Sample Library.

Customer requirements dictate that the breaking strengths in the previous example have upper and lower specification limits of 7.2 and 6.8 psi, respectively. Moreover, less than 5% of the cords can have breaking strengths outside the limits.

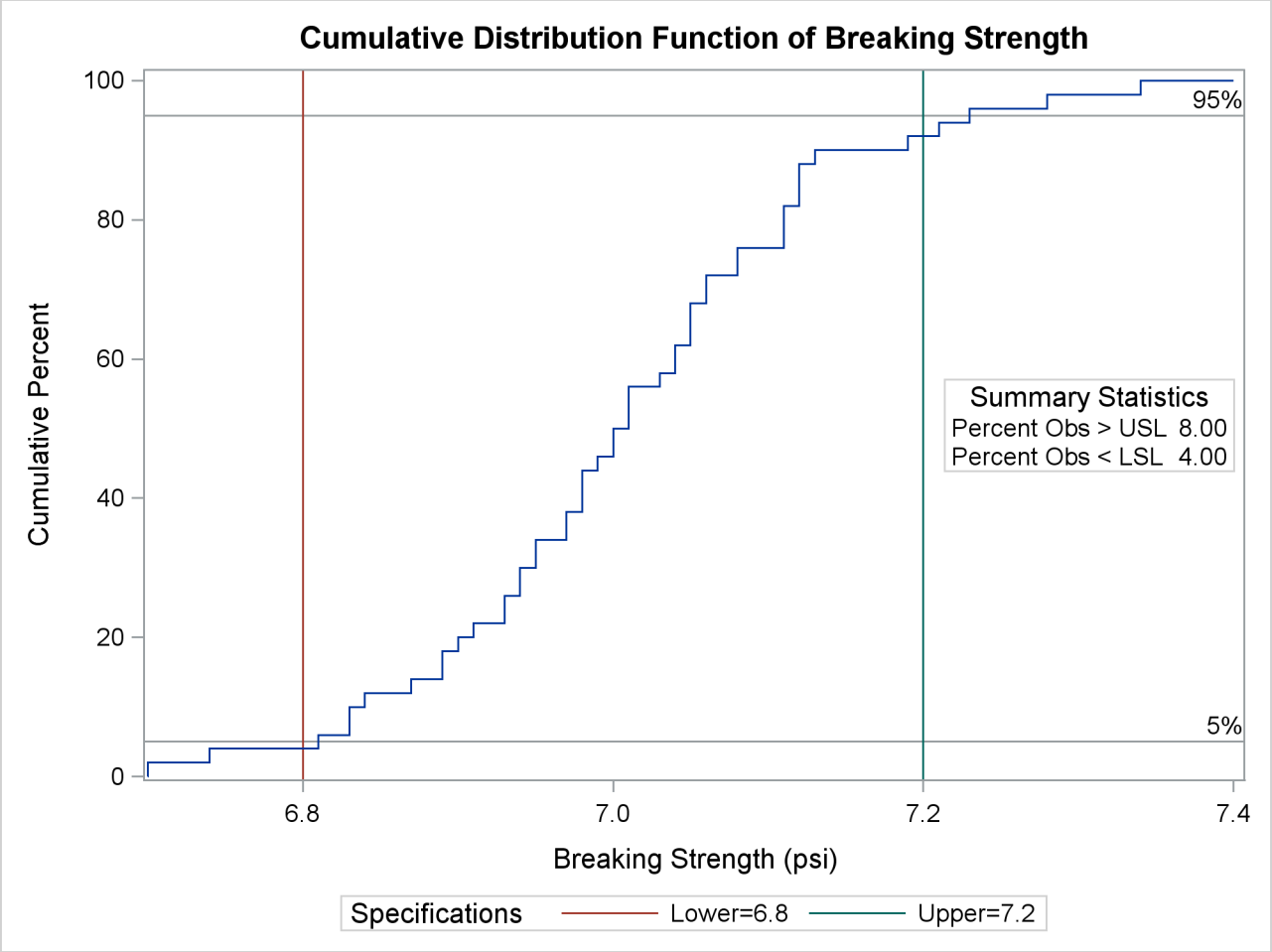
The following statements create a cdf plot with reference lines at the 5% and 95% cumulative percent levels:

```
proc capability data=Cord noprint;
   spec lsl=6.8 usl=7.2;
   cdf Strength / vref         = 5 95
                  vreflabels   = '5%' '95%'
                  odstitle     = title;
   inset pctgtr pctlss / format = 5.2
                         pos    = e
                         header = "Summary Statistics";
run;
```

The INSET statement requests an inset with the percentages of measurements above the upper limit and below the lower limit. For more information about the INSET statement, see "INSET Statement: CAPABILITY Procedure" on page 385.

In Output 6.5.1, the empirical cdf is below the intersection between the lower specification limit line and the 5% line, so less than 5% of the measurements are below the lower limit. The ecdf, however, is *also* below the intersection between the upper specification limit line and the 95% line, implying that *more* than 5% of the measurements are greater than the upper limit. Thus, the goal of having less than 5% of the measurements above the upper specification limit has not been met.

Output 6.5.1 Reference Lines with a Cumulative Distribution Function Plot



COMPHISTOGRAM Statement: CAPABILITY Procedure

Overview: COMPHISTOGRAM Statement

Comparative histograms are useful for comparing the distribution of a process variable across levels of classification variables. You can use the COMPHISTOGRAM statement to create one-way and two-way comparative histograms. When used with a single classification variable, the COMPHISTOGRAM statement displays an array of component histograms (stacked or side-by-side), one for each level of the classification variable. When used with two classification variables, the COMPHISTOGRAM statement displays a matrix of component histograms, one for each combination of levels of the classification variables.

In quality improvement applications, typical uses of comparative histograms include

- comparing the capability of a process before and after an improvement
- comparing process capabilities of two or more suppliers
- exploring stratification in process data due to different lots, machines, manufacturing methods, and so forth
- studying the evolution of process capability over successive time periods

You can use options in the COMPHISTOGRAM statement to

- specify the midpoints or endpoints for histogram intervals
- specify the number of rows and/or columns of component histograms
- display specification limits on the component histograms
- display density curves for fitted normal distributions
- display kernel density estimates
- request graphical enhancements
- inset summary statistics and process capability indices on the component histograms

You have two alternatives for producing comparative histograms with the COMPHISTOGRAM statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced if SAS/GRAPH® is licensed.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: You cannot use the COMPHISTOGRAM statement together with the CLASS statement.

Getting Started: COMPHISTOGRAM Statement

This section introduces the COMPHISTOGRAM statement with examples that illustrate commonly used options. Complete syntax for the COMPHISTOGRAM statement is presented in the section "Syntax: COMPHISTOGRAM Statement" on page 278, and advanced examples are given in the section "Examples: COMPHISTOGRAM Statement" on page 294.

Creating a One-Way Comparative Histogram

NOTE: See *Comparative Histograms with Normal Curves* in the SAS/QC Sample Library.

The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable Length in a SAS data set named Channel:

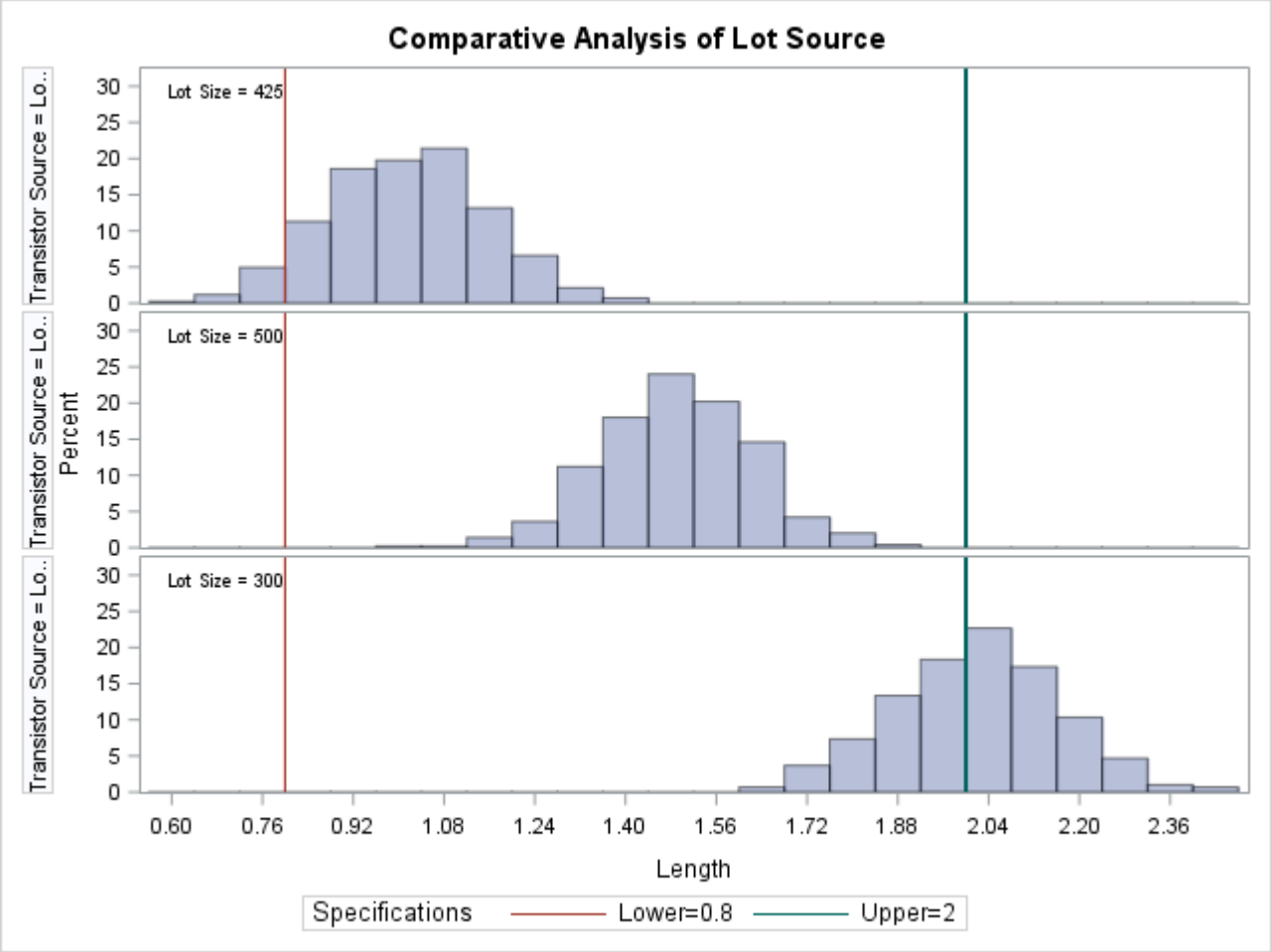
```
data Channel;
   length Lot $ 16;
   input Length @@;
   select;
      when (_n_ <= 425) Lot='Lot 1';
      when (_n_ >= 926) Lot='Lot 3';
      otherwise Lot='Lot 2';
   end;
   datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15
1.11 0.98 0.78 1.09 0.94 1.05 0.89 1.16 0.88 1.19
1.01 1.08 1.19 0.94 0.92 1.27 0.90 0.88 1.38 1.02

   ... more lines ...

2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

The data set Channel is also used in Example 6.12, where a kernel density estimate is superimposed on the histogram of channel lengths. The display in Output 6.12.1 reveals that there are three distinct peaks in the process distribution. To investigate whether these peaks (modes) in the histogram are related to the lot source, you can create a comparative histogram that uses Lot as a classification variable. The following statements create the comparative histogram shown in Figure 6.6:

```
title "Comparative Analysis of Lot Source";
proc capability data=Channel noprint;
   specs lsl = 0.8 usl = 2.0;
   comphist Length / class      = Lot
                     nrows      = 3
                     nlegend    = 'Lot Size'
                     nlegendpos = nw
                     odstitle   = title;
   label Lot = 'Transistor Source';
run;
```

The COMPHISTOGRAM statement requests a comparative histogram for the process variable Length. The CLASS= option requests a component histogram for each level (distinct value) of the classification variable Lot. The option NROWS=3 stacks the histograms three to a page. The NLEGEND= option adds a sample size legend to each component histogram, and the option NLEGENDPOS=NW positions each legend in the northwest corner. The SPEC statement provides the specification limits displayed as vertical reference lines. See the section "Dictionary of Options" on page 284 for descriptions of these options, and see the section "SPEC Statement" on page 214 for details of the SPEC statement.

Figure 6.6 Comparison by Lot Source



Adding Fitted Normal Curves to a Comparative Histogram

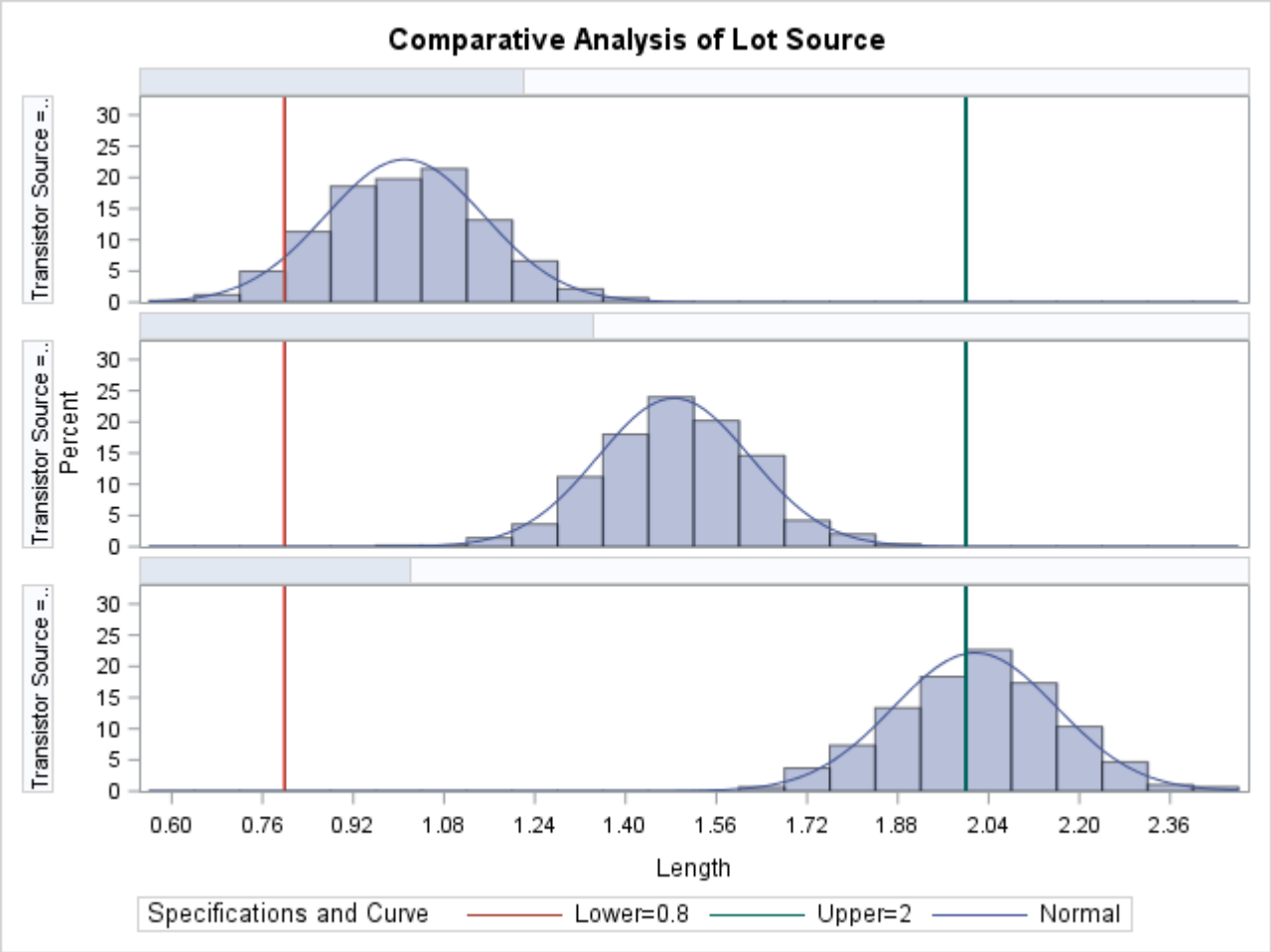
NOTE: See *Comparative Histograms with Normal Curves* in the SAS/QC Sample Library.

In Figure 6.6, it appears that each lot produces transistors with channel lengths that are normally distributed. The following statements use the NORMAL option to fit a normal distribution to the data for each lot (the observations corresponding to a specific level of the classification variable are referred to as a *cell*). The normal parameters μ and σ are estimated from the data for each lot, and the curves are superimposed on each component histogram.

```
title "Comparative Analysis of Lot Source";
proc capability data=Channel noprint;
   specs lsl = 0.8 usl = 2.0;
   comphist Length / class     = Lot
                     nrows     = 3
                     intertile = 1
                     odstitle  = title
                     cprop
                     normal;
   label Lot = 'Transistor Source';
run;
```

The comparative histogram is displayed in Figure 6.7.

Figure 6.7 Fitting Normal Curves



Specifying INTERTILE=1 inserts a space of one percent screen unit between the framed areas, which are referred to as *tiles*. The shaded bars, added with the CPROP= option, represent the relative frequency of observations in each cell. See "Dictionary of Options" on page 284 for details concerning these options.

Syntax: COMPHISTOGRAM Statement

The syntax for the COMPHISTOGRAM statement is as follows:

COMPHISTOGRAM < *variables* > / **CLASS=(***class-variables***)** < *options* > **;**

You can specify the keyword COMPHIST as an alias for COMPHISTOGRAM. You can use any number of COMPHISTOGRAM statements after a PROC CAPABILITY statement.

To create a comparative histogram, you must specify at least one *variable* and either one or two *class-variables* (also referred to as *classification variables*). The COMPHISTOGRAM statement displays a component histogram of the values of the *variable* for each level of the *class-variables*. The observations in a given level are referred to as a *cell*.

The components of the COMPHISTOGRAM statement are described as follows:

variables
are the process variables for which comparative histograms are to be created. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, variables can be any numeric variables in the input data set that are not also listed as *class-variables*. If you do not specify variables in a COMPHISTOGRAM statement or a VAR statement, then by default a comparative histogram is created for each numeric variable in the DATA= data set that is not used as a class-variable. If you use a VAR statement and do not specify variables in the COMPHISTOGRAM statement, then by default a comparative histogram is created for each variable listed in the VAR statement.

For example, suppose a data set named steel contains two process variables named length and width, a numeric classification variable named lot, and a character classification variable named day. The following statements create two comparative histograms, one for length and one for Width:

```
proc capability data=steel;
   comphist / class = lot;
run;
```

Likewise, the following statements create comparative histograms for length and width:

```
proc capability data=steel;
   var length width;
   comphist / class = day;
run;
```

The following statements create three comparative histograms (for length, width, and lot):

```
proc capability data=steel;
   comphist / class = day;
run;
```

The following statements create a comparative histogram for Width only:

```
proc capability data=steel;
   var length width;
   comphist width / class=lot;
run;
```

class-variables
are one or two required classification variables. For example, the following statements create a one-way comparative histogram for width by using the classification variable lot:

```
proc capability data=steel;
   comphist width / class=lot;
run;
```

The following statements create a two-way comparative histogram for width classified by lot and day:

```
proc capability data=steel;
   comphist width / class=(lot day);
run;
```

Note that the parentheses surrounding the *class-variables* are needed only if two classification variables are specified. See Output 6.6.1 and Output 6.7.1 for further examples.

options

control the features of the comparative histogram. All options are specified after the slash (/) in the COMPHIST statement. In the following example, the CLASS= option specifies the classification variable, the NORMAL option fits a normal density curve in each cell, and the CTEXT= option specifies the color of the text:

```
proc capability data=steel;
   comphist length / class = lot
                     normal
                     ctext = yellow;
run;
```

Summary of Options

The following tables list the COMPHIST statement options by function. For complete descriptions, see "Dictionary of Options" on page 284.

Distribution Options

Table 6.13 lists the options for requesting that a fitted normal distribution or a kernel density estimate be overlaid on the comparative histogram.

Table 6.13 Density Estimation Options

Option	Description
KERNEL(*kernel-options*)	fits kernel density estimates
NORMAL(*normal-options*)	fits normal distribution with mean μ and standard deviation σ

You can specify the secondary options listed in Table 6.14 in parentheses after the KERNEL option to control features of kernel density estimates.

Table 6.14 Kernel-Options

Option	Description
C=	specifies standardized bandwidth parameter c for kernel density estimate
COLOR=	specifies color of the kernel density curve
FILL	fills area under kernel density curve
K=	specifies NORMAL, TRIANGULAR, or QUADRATIC kernel
L=	specifies line type used for kernel density curve
LOWER=	specifies lower bound for kernel density curve
UPPER=	specifies upper bound for kernel density curve
W=	specifies line width for kernel density curve

You can specify the secondary options listed in Table 6.15 in parentheses after the NORMAL option to control features of fitted normal distributions.

Table 6.15 Normal-Options

Option	Description
COLOR=	specifies color of normal curve
FILL	fills area under normal curve
L=	specifies line type of normal curve
MU=	specifies mean μ for fitted normal curve
SIGMA=	specifies standard deviation σ for fitted normal curve
W=	specifies width of normal curve

For example, the following statements use the NORMAL option to fit a normal curve in each cell of the comparative histogram:

```
proc capability;
   comphistogram / class = machine
                   normal(color=red l=2);
run;
```

The COLOR= *normal-option* draws the curve in red, and the L= *normal-option* specifies a line style of 2 (a dashed line) for the curve. In this example, maximum likelihood estimates are computed for the normal parameters μ and σ for each cell because these parameters are not specified.

General Options

Table 6.16 General COMPHISTOGRAM Statement Options

Option	Description
Classification Options	
CLASS=	specifies classification variables
CLASSKEY=	specifies key cell

Table 6.16 (*continued*)

Option	Description
MISSING1	requests that missing values of first CLASS= variable be treated as a level of that CLASS= variable
MISSING2	requests that missing values of second CLASS= variable be treated as a level of that CLASS= variable
ORDER1=	specifies display order for values of the first CLASS= variable
ORDER2=	specifies display order for values of the second CLASS= variable
Layout Options	
BARLABEL=	produces labels above histogram bars
BARWIDTH=	specifies width for the bars
CLIPSPEC=	clips histogram bars at specification limits if there are no observations beyond the limits
ENDPOINTS=	labels interval endpoints and specifies how they are determined
HOFFSET=	specifies offset for horizontal axis
INTERTILE=	specifies distance between tiles
MAXNBIN=	specifies maximum number of bins displayed
MAXSIGMAS=	limits number of bins displayed to range of *value* standard deviations above and below mean of data in key cell
MIDPOINTS=	specifies how midpoints are determined
NCOLS=	specifies number of columns in comparative histogram
NOBARS	suppresses histogram bars
NOFRAME	suppresses frame around plotting area
NOKEYMOVE	suppresses rearrangement of cells that occurs by default with the CLASSKEY= option
NOPLOT	suppresses plot
NROWS=	specifies number of rows in comparative histogram
RTINCLUDE	includes right endpoint in interval
WBARLINE=	specifies line thickness for bar outlines
Axis and Legend Options	
GRID	adds grid corresponding to vertical axis
LGRID=	specifies line style for grid requested with GRID option
NLEGEND	specifies form of the legend displayed inside tiles
NLEGENDPOS=	specifies position of legend displayed inside tiles
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
TILELEGLABEL=	specifies label displayed when _CTILE_ and _TILELG_ variables are provided in the CLASSSPEC= data set
TURNVLABELS	turns and strings out vertically characters in vertical axis labels
VAXIS=	specifies tick mark values for vertical axis
VAXISLABEL=	specifies label for vertical axis
VOFFSET=	specifies length of offset at upper end of vertical axis
VSCALE=	specifies scale for vertical axis
WAXIS=	specifies line thickness for axes and frame
WGRID=	specifies line thickness for grid

Table 6.16 (*continued*)

Option	Description
Reference Line Options	
FRONTREF	draws reference lines in front of histogram bars
HREF=	specifies reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies vertical position of labels for HREF= lines
LHREF=	specifies line style for HREF= lines
LVREF=	specifies line style for VREF= lines
VREF=	specifies reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	specifies horizontal position of labels for VREF= lines
Text Enhancement Options	
FONT=	specifies software font for text
HEIGHT=	specifies height of text used outside framed areas
INFONT=	specifies software font for text inside framed areas
INHEIGHT=	specifies height of text inside framed areas
Color and Pattern Options	
CAXIS=	specifies color for axis
CBARLINE=	specifies color for outline of the bars
CFILL=	specifies color for filling bars
CFRAME=	specifies color for frame
CFRAMENLEG=	specifies the color for the frame requested by the NLEGEND option
CFRAMESIDE=	specifies color for filling frame for row labels
CFRAMETOP=	specifies color for filling frame for column labels
CGRID=	specifies color for grid lines
CHREF=	specifies color for HREF= lines
CPROP=	specifies color for proportion of frequency bar
CTEXT=	specifies color for text
CTEXTSIDE=	specifies color for row labels
CTEXTTOP=	specifies color for column labels
CVREF=	specifies color for VREF= lines
PFILL=	specifies pattern used to fill bars
Input and Output Data Set Options	
ANNOKEY	applies annotation requested in ANNOTATE= data set to key cell only
ANNOTATE=	annotate data set
CLASSSPEC=	data set with specification limit information for each cell
OUTHISTOGRAM=	information on histogram intervals
Graphics Catalog Options	
DESCRIPTION=	specifies description for graphics catalog member
NAME=	specifies name for plot in graphics catalog

Dictionary of Options

The following sections describe in detail the options specific to the COMPHISTOGRAM statement. See "Dictionary of Common Options: CAPABILITY Procedure" on page 534 for detailed descriptions of options common to all the plot statements.

General Options

You can specify the following options whether you are producing ODS Graphics output or traditional graphics:

BARLABEL=COUNT | PERCENT | PROPORTION

displays labels above the histogram bars. If you specify BARLABEL=COUNT, the label shows the number of observations associated with a given bar. BARLABEL=PERCENT shows the percent of observations represented by that bar. If you specify BARLABEL=PROPORTION, the label displays the proportion of observations associated with the bar.

C=*value-list* **| MISE**

specifies the standardized bandwidth parameter c for kernel density estimates requested with the KERNEL option. You can specify up to five *values* to display multiple estimates in each cell. You can also specify the keyword MISE to request the bandwidth parameter that minimizes the estimated mean integrated square error (MISE). For example, consider the following statements (for more information, see "Kernel Density Estimates" on page 347):

```
proc capability;
   comphist length / class=batch kernel(c = 0.5 1.0 mise);
run;
```

The KERNEL option displays three density estimates. The first two have standardized bandwidths of 0.5 and 1.0, respectively. The third has a bandwidth parameter that minimizes the MISE. You can also use the C= and K= options (K= specifies kernel type) to display multiple estimates. For example, consider the following statements:

```
proc capability;
   comphist length / class = batch
                     kernel(c = 0.75 k = normal triangular);
run;
```

Here two estimates are displayed. The first uses a normal kernel and bandwidth parameter of 0.75, and the second uses a triangular kernel and a bandwidth parameter of 0.75. In general, if more kernel types are specified than bandwidth parameters, the last bandwidth parameter in the list will be repeated for the remaining estimates. Likewise, if more bandwidth parameters are specified than kernel types, the last kernel type will be repeated for the remaining estimates. The default is MISE.

CLASS=*variable*

CLASS=(*variable1 variable2***)**

specifies that a comparative histogram is to be created using the levels of the *variables* (also referred to as *class-variables* or *classification variables*).

If you specify a single *variable*, a one-way comparative histogram is created. The observations in the input data set are sorted by the formatted values (levels) of the variable. A separate histogram is

created for the process variable values in each level, and these component histograms are arranged in an array to form the comparative histogram. Uniform horizontal and vertical axes are used to facilitate comparisons. For an example, see Figure 6.6.

If you specify two *classification variables*, a two-way comparative histogram is created. The observations in the input data set are cross-classified according to the values (levels) of these variables. A separate histogram is created for the process variable values in each cell of the cross-classification, and these component histograms are arranged in a matrix to form the comparative histogram. The levels of *variable1* are used to label the rows of the matrix, and the levels of *variable2* are used to label the columns of the matrix. Uniform horizontal and vertical axes are used to facilitate comparisons. For an example, see Output 6.7.1.

Classification variables can be numeric or character. Formatted values are used to determine the levels. You can specify whether missing values are to be treated as a level with the MISSING1 and MISSING2 options.

If a label is associated with a classification variable, the label is displayed on the comparative histogram. The variable label is displayed parallel to the column (or row) labels. For an example, see Figure 6.6.

CLASSKEY='*value*'

CLASSKEY=('*value1*' '*value2*'**)**

specifies the *key cell* in a comparative histogram requested with the CLASS= option. The bin size and midpoints are first determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. Thus, the choice of the key cell determines the uniform horizontal axis used for all cells.

If you specify CLASS=*variable*, you can specify CLASSKEY=*'value'* to identify the key cell as the level for which *variable* is equal to *value*. You must specify a formatted *value*. By default, the levels are sorted in the order determined by the ORDER1= option, and the key cell is the level that occurs first in this order. The cells are displayed in this order from top to bottom (or left to right), and, consequently, the key cell is displayed at the top or at the left. If you specify a different key cell with the CLASSKEY= option, this cell is displayed at the top or at the left unless you also specify the NOKEYMOVE option.

If you specify CLASS=(*variable1 variable2*), you can specify CLASSKEY=(*'value1' 'value2'*) to identify the key cell as the level for which *variable1* is equal to *value1* and *variable2* is equal to *value2*. Here, *value1* and *value2* must be formatted values, and they must be enclosed in quotes. For an example of the CLASSKEY= option with a two-way comparative histogram, see Output 6.7.1. By default, the levels of *variable1* are sorted in the order determined by the ORDER1= option, and within each of these levels, the levels of *variable2* are sorted in the order determined by the ORDER2= option. The default key cell is the combination of levels of *variable1* and *variable2* that occurs first in this order. The cells are displayed in order of *variable1* from top to bottom and in order of *variable2* from left to right. Consequently, the default key cell is displayed in the upper left corner. If you specify a different key cell with the CLASSKEY= option, this cell is displayed in the upper left corner unless you also specify the NOKEYMOVE option.

CLASSSPEC=*SAS-data-set*

CLASSSPECS=*SAS-data-set*

specifies a data set that provides distinct specification limits for each cell, as well as a color, legend, and label for the corresponding tile. The following table lists the variables that are read from a CLASSSPEC= data set:

Variable Name	Description
BY variables	subsets the data set
Classification variables	specifies the structure of the comparative histogram
VAR	specifies name of process variable (must be character variable of length 8)
LSL	specifies lower specification limit for tile
TARGET	specifies target value for tile
USL	specifies upper specification limit for tile
CTILE	specifies background color for tiles (must be character variable of length 8)
TILELG	specifies text displayed in color tile legend at bottom of comparative histogram (character variable of length not greater than 16)
TILELB	specifies text displayed in corner of each tile (character variable of length not greater than 16)

If you specify a CLASSSPEC= data set, you cannot use the SPEC statement or a SPEC= data set. If you use a BY statement, the CLASSSPEC= data set must contain one observation for each unique combination of process and classification variables within each BY group. See Example 6.6 for an example of a CLASSSPEC= data set.

Also note that

- you can suppress the background color for a tile by assigning the value 'EMPTY' or a blank value to the variable _CTILE_
- you can use the NLEGENDPOS= option to specify the corner of the tile in which the _TILELB_ label is displayed. You can frame the label with the CFRAMENLEG= option.
- you cannot use the variable _TILELG_ unless you specify the variable _CTILE_
- the variable _TILELB_ takes precedence over the NLEGEND option

ENDPOINTS=*value-list* **| KEY | UNIFORM**

specifies that histogram interval endpoints, rather than midpoints, are aligned with horizontal axis tick marks, and specifies how the endpoints are determined. The method you specify is used for all process variables analyzed with the COMPHISTOGRAM statement.

If you specify ENDPOINTS=*value-list*, the *values* must be listed in increasing order and must be evenly spaced. The difference between consecutive endpoints is used as the width of the histogram bars. The first value is the lower bound of the first histogram bin and the last value is the upper bound of the last bin. Thus, the number of values in the list is one greater than the number of bins it specifies. If the range of the *values* does not cover the range of the data as well as any specification limits (LSL and USL) that are given, the list is extended in either direction as necessary.

If you specify ENDPOINTS=KEY, the procedure first determines the endpoints for the data in the key cell. The initial number of endpoints is based on the number of observations in the key cell by using the method of Terrell and Scott (1985). The endpoint list for the key cell is then extended in either direction as necessary until it spans the data in the remaining cells. If the key cell contains no observations, the method of determining bins reverts to ENDPOINTS=UNIFORM.

If you specify ENDPOINTS=UNIFORM, the procedure determines the endpoints by using all the observations as if there were no cells. In other words, the number of endpoints is computed from the total sample size by using the method of Terrell and Scott (1985).

FILL

fills areas under a fitted density curve with colors and patterns. Enclose the FILL option in parentheses after the keyword NORMAL or KERNEL. Depending on the area to be filled (outside or between the specification limits), you can specify the color and pattern with options in the SPEC statement and the COMPHISTOGRAM statement, as summarized in the following table:

Area Under Curve	Statement	Option
between specification	COMPHIST	CFILL=*color*
limits	COMPHIST	PFILL=*pattern*
left of lower	SPEC	CLEFT=*color*
specification limit	SPEC	PLEFT=*pattern*
right of upper	SPEC	CRIGHT=*color*
specification limit	SPEC	PRIGHT=*pattern*

If you do not display specification limits, you can use the CFILL= and PFILL= options to specify the color and pattern for the entire area under the curve. Solid fills are used by default if patterns are not specified. You can specify the FILL option with only one fitted curve. For an example, see Output 6.6.1. Refer to *SAS/GRAPH: Reference* for a list of available patterns and colors. If you do not specify the FILL option but you do specify the options in the preceding table, the colors and patterns are applied to the corresponding areas under the histogram.

GRID

adds a grid to the comparative histogram. Grid lines are horizontal lines positioned at major tick marks on the vertical axis.

INTERTILE=*value*

specifies the distance in horizontal percent screen units between tiles. For an example, see Figure 6.7. By default, the tiles are contiguous.

K=NORMAL | TRIANGULAR | QUADRATIC

specifies the type of kernel (normal, triangular, or quadratic) used to compute kernel density estimates requested with the KERNEL option. Enclose the K= option in parentheses after the keyword KERNEL. You can specify a single type or a list of types. If you specify more estimates than types, the last kernel type in the list is used for the remaining estimates. By default, a normal kernel is used.

KERNEL< (*kernel-options*) >

requests a kernel density estimate for each cell of the comparative histogram. You can specify the *kernel-options* described in the following table:

Option	Description
FILL	specifies that the area under the curve is to be filled
COLOR=	specifies the color of the curve
L=	specifies the line style for the curve
W=	specifies the width of the curve
K=	specifies the type of kernel
C=	specifies the smoothing parameter
LOWER=	specifies the lower bound for the curve
UPPER=	specifies the upper bound for the curve

See Output 6.6.1 for an example. By default, the estimate is based on the AMISE method. For more information, see "Kernel Density Estimates" on page 347.

LOWER=*value*

specifies the lower bound for a kernel density estimate curve. Enclose the LOWER= option in parentheses after the KERNEL option. You can specify a single lower bound or a list of lower bounds. By default, a kernel density estimate curve has no lower bound.

MAXNBIN=*n*

specifies the maximum number of bins to be displayed. This option is useful in situations where the scales or ranges of the data distributions differ greatly from cell to cell. By default, the bin size and midpoints are determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. However, if the cell scales differ considerably, the resulting number of bins may be so great that each cell histogram is scaled into a narrow region. By limiting the number of bins with the MAXNBIN= option, you can narrow the window about the data distribution in the key cell. Note that the MAXNBIN= option provides an alternative to the MAXSIGMAS= option.

MAXSIGMAS=*value*

limits the number of bins to be displayed to a range of *value* standard deviations (of the data in the key cell) above and below the mean of the data in the key cell. This option is useful in situations where the scales or ranges of the data distributions differ greatly from cell to cell. By default, the bin size and midpoints are determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. If the cell scales differ considerably, however, the resulting number of bins may be so great that each cell histogram is scaled into a narrow region. By limiting the number of bins with the MAXSIGMAS= option, you narrow the window about the data distribution in the key cell. Note that the MAXSIGMAS= option provides an alternative to the MAXNBIN= option.

MIDPOINTS=*value-list* | KEY | UNIFORM

specifies how midpoints are determined for the bins in the comparative histogram. The method you specify is used for all process variables analyzed with the COMPHISTOGRAM statement.

If you specify MIDPOINTS=*value-list*, the *values* must be listed in increasing order and must be evenly spaced. The difference between consecutive midpoints is used as the width of the histogram bars. If the range of the *values* does not cover the range of the data as well as any specification limits (LSL and USL) that are given, the list is extended in either direction as necessary. See Example 6.6 for an illustration.

If you specify MIDPOINTS=KEY, the procedure first determines the midpoints for the data in the key cell. The initial number of midpoints is based on the number of observations in the key cell by using

the method of Terrell and Scott (1985). The midpoint list for the key cell is then extended in either direction as necessary until it spans the data in the remaining cells.

If you specify MIDPOINTS=UNIFORM, the procedure determines the midpoints using all the observations as if there were no cells. In other words, the number of midpoints is computed from the total sample size by using the method of Terrell and Scott (1985).

By default, MIDPOINTS=KEY. However, if the key cell contains no observations, the default is MIDPOINTS=UNIFORM.

MISSING1
specifies that missing values of the first CLASS= variable are to be treated as a level of the CLASS= variable. If the first CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS System missing values. If you do not specify MISSING1, observations for which the first CLASS= variable is missing are excluded from the analysis.

MISSING2
specifies that missing values of the second CLASS= variable are to be treated as a level of the CLASS= variable. If the second CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS System missing values. If you do not specify MISSING2, observations for which the second CLASS= variable is missing are excluded from the analysis.

MU=*value*
specifies the parameter μ for the normal density curves requested with the NORMAL option. Enclose the MU= option in parentheses after the NORMAL option. The default value is the sample mean of the observations in the cell.

NOBARS
suppresses the display of the bars in a comparative histogram.

NOCHART
suppresses the creation of a comparative histogram. This is an alias for NOPLOT.

NOKEYMOVE
suppresses the rearrangement of cells that occurs by default when you use the CLASSKEY= option to specify the key cell. For details, see the entry for the CLASSKEY= option.

NOPLOT
suppresses the creation of a comparative histogram. This option is useful when you are using the COMPHISTOGRAM statement solely to create an output data set.

NORMAL< (*normal-options*) >
displays a normal density curve for each cell of the comparative histogram. The equation of the normal density curve is

$$p(x) = \frac{hv}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

where

μ = mean
σ = standard deviation ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

If you specify values for μ and σ with the MU= and SIGMA= *normal-options*, the same curve is displayed for each cell. By default, a distinct curve is displayed for each cell based on the sample mean and standard deviation for that cell. For example, the following statements display a distinct curve for each level of the variable Supplier:

```
proc capability noprint;
   comphist width / class=supplier normal(color=red l=2);
run;
```

The curves are drawn in red with a line style of 2 (a dashed line). See Figure 6.7 for another illustration. Table 6.15 lists options that can be specified in parentheses after the NORMAL option.

ORDER1=INTERNAL | FORMATTED | DATA | FREQ

specifies the display order for the values of the first CLASS= variable.

The levels of the first CLASS= variable are always constructed using the *formatted* values of the variable, and the formatted values are always used to label the rows (columns) of a comparative histogram. You can use the ORDER1= option to determine the order of the rows (columns) corresponding to these values, as follows:

- **If you specify ORDER1=INTERNAL**, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the internal (unformatted) values of the first CLASS= variable. If there are two or more distinct internal values with the same formatted value, then the order is determined by the internal value that occurs first in the input data set.

  For example, suppose that you specify a numeric CLASS= variable called Day (with values 1, 2, and 3). Suppose also that a format (created with the FORMAT procedure) is associated with Day and that the formatted values are as follows: 1 = 'Wednesday', 2 = 'Thursday', and 3 = 'Friday'. If you specify ORDER1=INTERNAL, the rows of the comparative histogram will appear in day-of-the-week order (*Wednesday*, *Thursday*, *Friday*) from top to bottom.
- **If you specify ORDER1=FORMATTED**, the rows (columns) are displayed from top to bottom (left to right) in increasing order of the formatted values of the first CLASS= variable. In the preceding illustration, if you specify ORDER1=FORMATTED, the rows will appear in alphabetical order (*Friday*, *Thursday*, *Wednesday*) from top to bottom.
- **If you specify ORDER1=DATA**, the rows (columns) are displayed from top to bottom (left to right) in the order in which the values of the first CLASS= variable first appear in the input data set.
- **If you specify ORDER1=FREQ**, the rows (columns) are displayed from top to bottom (left to right) in order of *decreasing* frequency count. If two or more classes have the same frequency count, the order is determined by the formatted values.

By default, ORDER1=INTERNAL.

ORDER2=INTERNAL | FORMATTED | DATA | FREQ
specifies the display order for the values of the second CLASS= variable.

The levels of the second CLASS= variable are always constructed using the *formatted* values of the variable, and the formatted values are always used to label the columns of a two-way comparative histogram. You can use the ORDER2= option to determine the order of the columns.

The layout of a two-way comparative histogram is determined by using the ORDER1= option to obtain the order of the rows from top to bottom (recall that ORDER1=INTERNAL by default). Then the ORDER2= option is applied to the observations corresponding to the first row to obtain the order of the columns from left to right. If any columns remain unordered (that is, the categories are *unbalanced*), the ORDER2= option is applied to the observations in the second row, and so on, until all the columns have been ordered.

The values of the ORDER2= option are interpreted as described for the ORDER1= option. By default, ORDER2=INTERNAL.

OUTHISTOGRAM=*SAS-data-set*
creates a SAS data set that saves the midpoints or endpoints of the histogram intervals, the observed percent of observations in each interval, and (optionally) the percent of observations in each interval estimated from a fitted normal distribution. By default, interval midpoint values are saved in the variable _MIDPT_. If the ENDPOINTS= option is specified, intervals are identified by endpoint values instead. If RTINCLUDE is specified, the _MAXPT_ variable contains upper endpoint values. Otherwise, lower endpoint values are saved in the _MINPT_ variable.

RTINCLUDE
includes the right endpoint of each histogram interval in that interval. The left endpoint is included by default.

SIGMA=*value*
specifies the parameter σ for normal density curves requested with the NORMAL option. Enclose the SIGMA= option in parentheses after the NORMAL option. The default value is the sample standard deviation of the observations in the cell.

UPPER=*value*
specifies the upper bound for a kernel density estimate curve. Enclose the UPPER= option in parentheses after the KERNEL option. You can specify a single upper bound or a list of upper bounds. By default, a kernel density estimate curve has no upper bound.

VSCALE=PERCENT | COUNT | PROPORTION
specifies the scale of the vertical axis. The value COUNT scales the data in units of the number of observations per data unit. The value PERCENT scales the data in units of percent of observations per data unit. The value PROPORTION scales the data in units of proportion of observations per data unit. The default is PERCENT.

Options for Traditional Graphics

You can specify the following options if you are producing traditional graphics:

BARWIDTH=*value*
specifies the width of the histogram bars in screen percent units.

CBARLINE=*color*
specifies the color of the outline of the histogram bars. This option overrides the C= option in the SYMBOL1 statement.

CFILL=*color*
specifies a color used to fill the bars of the histograms (or the areas under a fitted curve if you also specify the FILL option). See the entry for the FILL option for additional details. See Output 6.6.1 and Example 6.7 for examples. Refer to *SAS/GRAPH: Reference* for a list of colors. By default, bars and curve areas are not filled.

CFRAMENLEG=*color* | **EMPTY**
CFRAMENLEG
specifies that the legend requested with the NLEGEND option (or the variable _TILELB_ in a CLASSSPEC= data set) is to be framed and that the frame is to be filled with the color indicated. If you specify CFRAMENLEG=EMPTY, a frame is drawn but not filled with a color.

CGRID=*color*
specifies the color for grid lines requested with the GRID option. By default, grid lines are the same color as the axes. If you use CGRID=, you do not need to specify the GRID option.

CLIPSPEC=CLIP | **NOFILL**
specifies that histogram bars are clipped at the upper and lower specification limit lines when there are no observations outside the specification limits. The bar intersecting the lower specification limit is clipped if there are no observations less than the lower limit; the bar intersecting the upper specification limit is clipped if there are no observations greater than the upper limit. If you specify CLIPSPEC=CLIP, the histogram bar is truncated at the specification limit. If you specify CLIPSPEC=NOFILL, the portion of a filled histogram bar outside the specification limit is left unfilled. Specifying CLIPSPEC=NOFILL when histogram bars are not filled has no effect.

FRONTREF
draws reference lines requested with the HREF= and VREF= options in front of the histogram bars. By default, reference lines are drawn behind the histogram bars and can be obscured by them.

HOFFSET=*value*
specifies the offset in percent screen units at both ends of the horizontal axis. Specify HOFFSET=0 to eliminate the default offset.

LGRID=*n*
specifies the line type for the grid requested with the GRID option. If you use the LGRID= option, you do not need to specify the GRID option. The default is 1, which produces a solid line.

NLEGEND<=*'label'*>
specifies the form of a legend that is displayed inside each tile and indicates the sample size of the cell. The following two forms are available:

- If you specify the NLEGEND option, the form is $N = n$ where n is the cell sample size.
- If you specify the NLEGEND='*label*' option, the form is *label* = n where n is the cell sample size. The label can be up to 16 characters and must be enclosed in quotes. For instance, you might specify NLEGEND='Number of Parts' to request a label of the form *Number of Parts* = n.

See Figure 6.6 for an example. You can use the CFRAMENLEG= option to frame the sample size legend. The variable _TILELB_ in a CLASSSPEC= data set overrides the NLEGEND option. By default, no legend is displayed.

NLEGENDPOS=NW | NE
specifies the position of the legend requested with the NLEGEND option or the variable _TILELB_ in a CLASSSPEC= data set. If NLEGENDPOS=NW, the legend is displayed in the northwest corner of the tile; if NLEGENDPOS=NE, the legend is displayed in the northeast corner of the tile. See Figure 6.6 for an illustration. The default is NE.

PFILL=*pattern*
specifies a pattern used to fill the bars of the histograms (or the areas under a fitted curve if you also specify the FILL option). See the entries for the CFILL= and FILL options for additional details. Refer to *SAS/GRAPH: Reference* for a list of pattern values. By default, the bars and curve areas are not filled.

TILELEGLABEL=*'label'*
specifies a label displayed to the left of the legend that is created when you provide _CTILE_ and _TILELG_ variables in a CLASSSPEC= data set. The *label* can be up to 16 characters and must be enclosed in quotes. The default *label* is *Tiles:*.

VOFFSET=*value*
specifies the offset in percent screen units at the upper end of the vertical axis.

WBARLINE=*n*
specifies the width of bar outlines. By default, $n = 1$.

WGRID=*n*
specifies the width of the grid lines requested with the GRID option. By default, grid lines are the same width as the axes. If you use the WGRID= option, you do not need to specify the GRID option.

Details: COMPHISTOGRAM Statement

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. COMPHISTOGRAM options used to control the appearance of traditional graphics are ignored for ODS Graphics output.

When ODS Graphics is in effect, the COMPHISTOGRAM statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 6.17.

Table 6.17 ODS Graphics Produced by the COMPHISTOGRAM Statement

ODS Graph Name	Plot Description
Histogram	comparative histogram

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Examples: COMPHISTOGRAM Statement

This section provides advanced examples of comparative histograms.

Example 6.6: Adding Insets with Descriptive Statistics

NOTE: See *Machine Study with Comparative Histogram* in the SAS/QC Sample Library.

Three similar machines are used to attach a part to an assembly. One hundred assemblies are sampled from the output of each machine, and a part position is measured in millimeters. The following statements save the measurements in a SAS data set named Machines:

```
data Machines;
   input position @@;
   label position='Position in Millimeters';
   if      (_n_ <= 100) then Machine = 'Machine 1';
   else if (_n_ <= 200) then Machine = 'Machine 2';
   else                      Machine = 'Machine 3';
   datalines;
-0.17  -0.19  -0.24  -0.24  -0.12   0.07  -0.61   0.22   1.91  -0.08
-0.59   0.05  -0.38   0.82  -0.14   0.32   0.12  -0.02   0.26   0.19
-0.07   0.13  -0.49   0.07   0.65   0.94  -0.51  -0.61  -0.57  -0.51
 0.01  -0.51   0.07  -0.16  -0.32  -0.42  -0.42  -0.34  -0.34  -0.35
-0.49   0.11  -0.42   0.76   0.02  -0.59  -0.28   1.12  -0.02  -0.60
-0.64   0.13  -0.32  -0.77  -0.02  -0.07  -0.49  -0.53  -0.22   0.61
-0.23   0.02   0.53   0.23  -0.44  -0.05   0.37  -0.42   0.70  -0.35

   ... more lines ...

 0.58   0.46   0.58   0.92   0.70   0.81   0.07   0.33   0.82   0.62
 0.48   0.41   0.78   0.58   0.43   0.07   0.27   0.49   0.79   0.92
 0.79   0.66   0.22   0.71   0.53   0.57   0.90   0.48   1.17   1.03
;
```

Distinct specification limits for the three machines are provided in a data set named speclims.

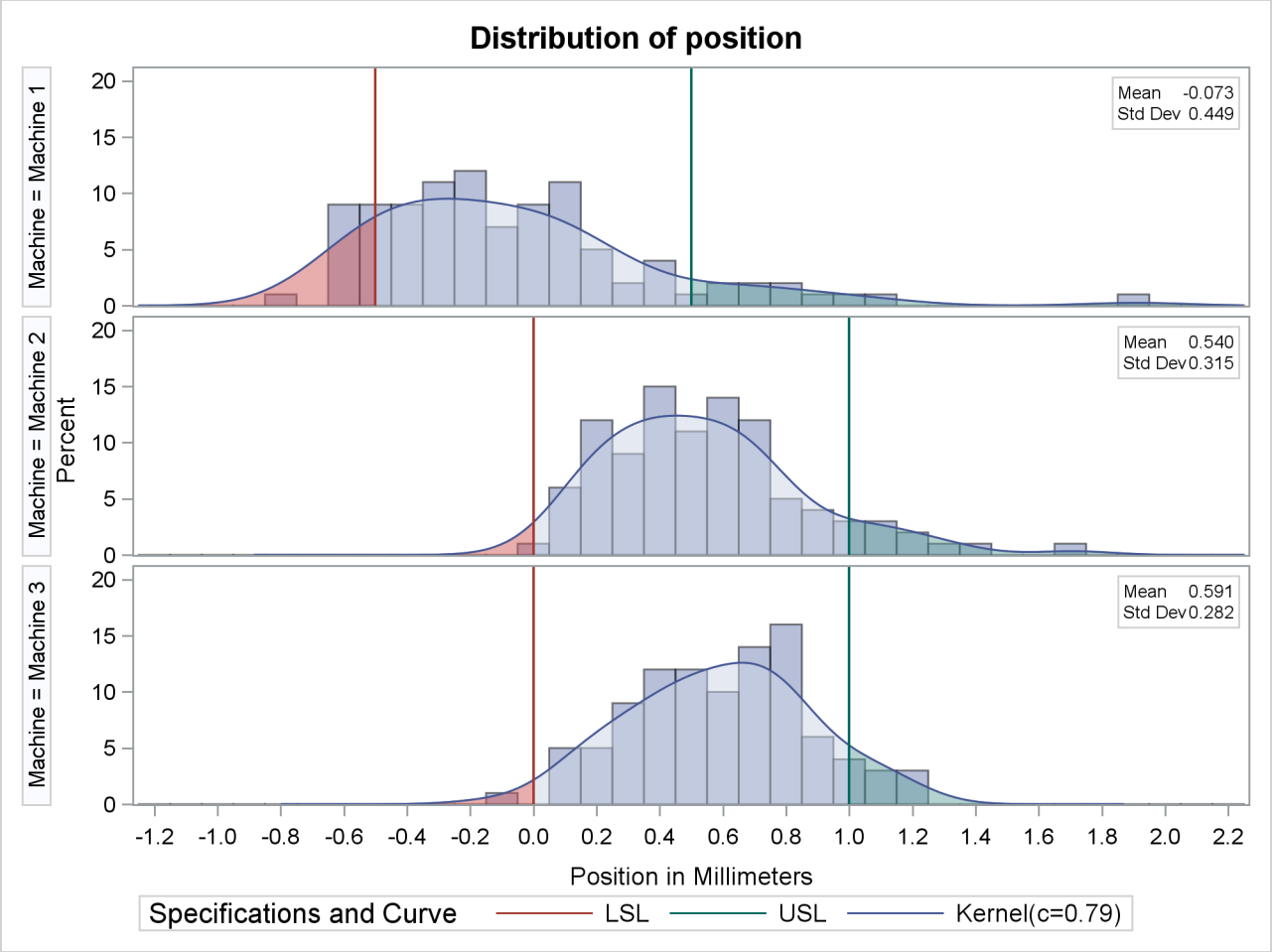
```
data speclims;
   input Machine $9. _lsl_ _usl_;
   _var_ = 'position';
   datalines;
Machine 1 -0.5 0.5
Machine 2  0.0 1.0
Machine 3  0.0 1.0
;
```

The following statements create a comparative histogram for the measurements in Machines that displays the specification limits in speclims.

```
proc capability data=Machines noprint;
   spec cleft cright;
   comphist position / class      = Machine
                       nrows      = 3
                       intertile  = 1
                       midpoints  = -1.2 to 2.2 by 0.1
                       kernel(fill)
                       classspecs = speclims;
   inset mean std="Std Dev" / pos = ne format = 6.3;
run;
```

The display is shown in Output 6.6.1.

Output 6.6.1 Comparative Histogram



The INSET statement is used to inset the sample mean and standard deviation for each machine in the corresponding tile. The MIDPOINTS= option specifies the midpoints of the histogram bins. Kernel density estimates are displayed using the KERNEL option. The curve areas outside the specification limits are filled using the CLEFT and CRIGHT options in the SPEC statement, and the area between the limits is filled using the CFILL= option in COMPHISTOGRAM statement.

Example 6.7: Creating a Two-Way Comparative Histogram

NOTE: See *Two-Way Comparative Histogram* in the SAS/QC Sample Library.

Two suppliers (A and B) provide disk drives for a computer manufacturer. The manufacturer measures the disk drive opening width to compare the process capabilities of the suppliers and determine whether there has been an improvement from 1992 to 1993.

The following statements save the measurements in a data set named Disk. There are two classification variables, Supplier and Year, and a format is associated with Year.

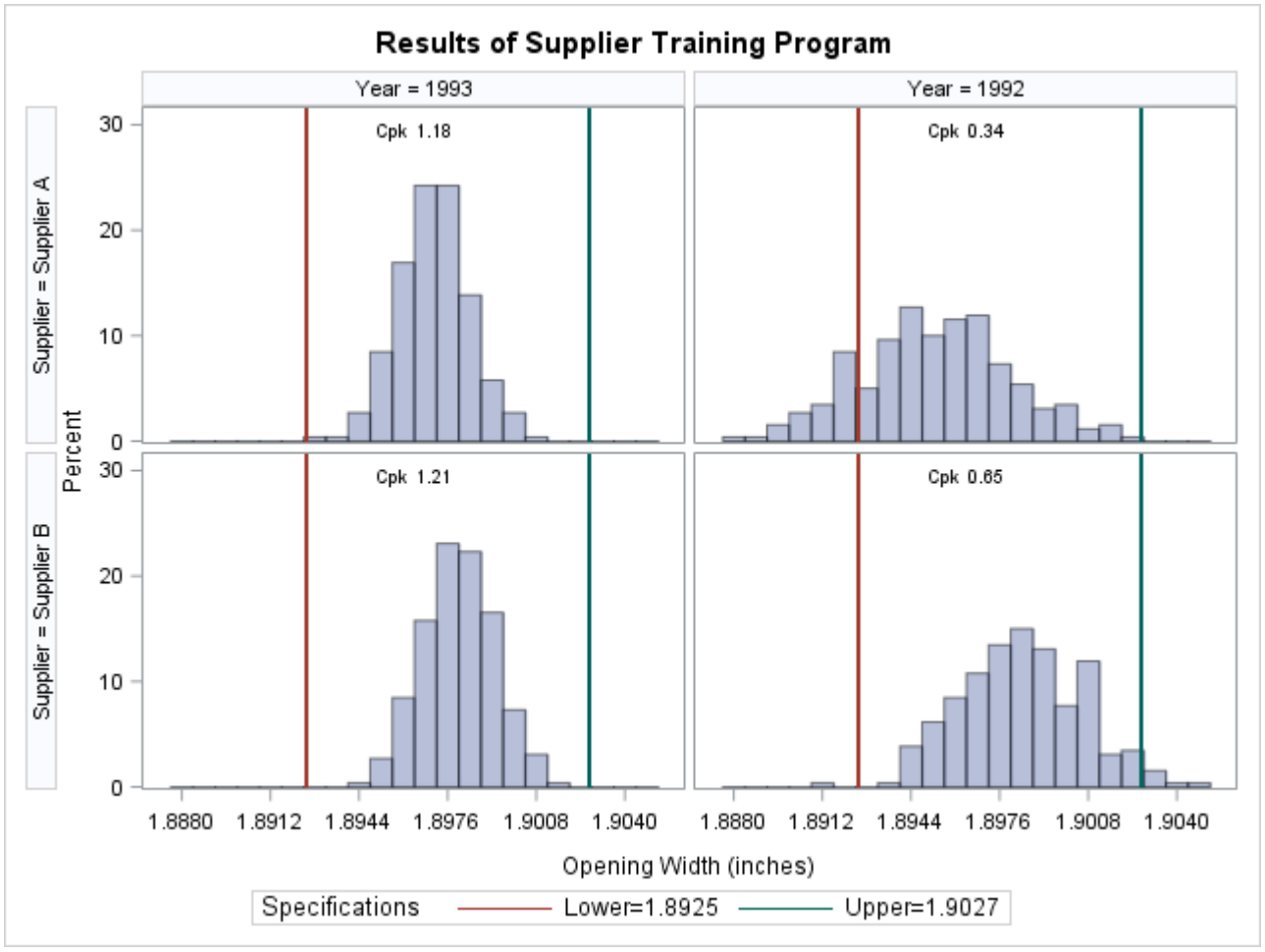
```
proc format ;
   value mytime  1 = '1992'
                 2 = '1993' ;

data disk;
   input @1 supplier $10. year width;
   label width = 'Opening Width (inches)';
   format year mytime.;
   datalines;
Supplier A   1   1.8932
Supplier A   1   1.8952
    .        .    .
    .        .    .
Supplier B   1   1.8980
Supplier B   1   1.8986
Supplier A   2   1.8978
Supplier A   2   1.8966
    .        .    .
    .        .    .
Supplier B   2   1.8967
Supplier B   2   1.8997
;
```

The following statements create the comparative histogram in Output 6.7.1:

```
* Define a format for time periods;
proc format ;
   value mytime
      1 = '1992'
      2 = '1993'
      3 = 'Target for 1994'
   ;
* Simulate the data;
data Disk;
   keep Supplier Year Width;
   length Supplier $ 16;
   label  Width = 'Opening Width (inches)';
   format Year mytime. ;
   Year = 1;
   Supplier = 'Supplier A';
   avg      = 1.895 ;
   std      = 0.0027 ;
   do i = 1 to 260;
      Width = avg + std * rannor(15535); output;
   end;
   Supplier = 'Supplier B';
   avg      = 1.8983 ;
   std      = 0.0024 ;
   do i = 1 to 260;
      Width = avg + std * rannor(15535); output;
   end;
   Year = 2;
   Supplier = 'Supplier A';
   avg      = 1.8970 ;
   std      = 0.0013 ;
   do i = 1 to 260;
```

```
      Width = avg + std * rannor(15535); output;
   end;
   Supplier = 'Supplier B';
   avg      = 1.8980 ;
   std      = 0.0013 ;
   do i = 1 to 260;
      Width = avg + std * rannor(15535); output;
   end;
run;

title "Results of Supplier Training Program";
proc capability data=Disk noprint;
   specs  lsl = 1.8925
          usl = 1.9027;
   comphist Width / class      = ( Supplier Year )
                    classkey   = ('Supplier A' '1993')
                    intertile  = 1.0
                    vaxis      = 0 10 20 30
                    ncols      = 2
                    nrows      = 2
                    odstitle   = title;
   inset cpk (4.2) / noframe pos = n;
run;
```

Output 6.7.1 Two-Way Comparative Histogram



The CLASSKEY= option specifies the key cell as the observations for which Supplier is equal to 'SUPPLIER A' and Year is equal to 2. This cell determines the binning for the other cells, and (because the NOKEYMOVE option is not specified) the columns are interchanged so that this cell is displayed in the upper left corner. Note that if the CLASSKEY= option were not specified, the default key cell would be the observations for which Supplier is equal to 'SUPPLIER A' and Year is equal to 1. If the CLASSKEY= option were not specified (or if the NOKEYMOVE option were specified), the column labeled *1992* would be displayed to the left of the column labeled *1993*. See the entry for the CLASSKEY= option on page 285 for details.

The VAXIS= option specifies the tick mark labels for the vertical axis, while NROWS=2 and NCOLS=2 specify a 2×2 arrangement for the tiles. The INSET statement is used to display the capability index C_{pk} for each cell. Output 6.7.1 provides evidence that both suppliers have reduced variability from 1992 to 1993.

HISTOGRAM Statement: CAPABILITY Procedure

Overview: HISTOGRAM Statement

Histograms are typically used in process capability analysis to compare the distribution of measurements from an in-control process with its specification limits. In addition to creating histograms, you can use the HISTOGRAM statement to do the following:

- specify the midpoints or endpoints for histogram intervals
- display specification limits on histograms
- display density curves for fitted theoretical distributions on histograms
- request goodness-of-fit tests for fitted distributions
- display kernel density estimates on histograms
- inset summary statistics and process capability indices on histograms
- save histogram intervals and parameters of fitted distributions in output data sets
- create hanging histograms
- request graphical enhancements
- create comparative histograms by using the HISTOGRAM statement together with a CLASS statement

You have three alternatives for producing histograms with the HISTOGRAM statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.

- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: HISTOGRAM Statement

This section introduces the HISTOGRAM statement with examples that illustrate commonly used options. Complete syntax for the HISTOGRAM statement is presented in the section "Syntax: HISTOGRAM Statement" on page 305, and advanced examples are given in the section "Examples: HISTOGRAM Statement" on page 363.

Creating a Histogram with Specification Limits

NOTE: See *Histogram with Fitted Normal Curve* in the SAS/QC Sample Library.

A semiconductor manufacturer produces printed circuit boards that are sampled to determine whether the thickness of their copper plating lies between a lower specification limit of 3.45 mils and an upper specification limit of 3.55 mils. The plating process is assumed to be in statistical control. The plating thicknesses of 100 boards are saved in a data set named Trans, created by the following statements:

```
data Trans;
   input Thick @@;
   label Thick='Plating Thickness (mils)';
   datalines;
3.468 3.428 3.509 3.516 3.461 3.492 3.478 3.556 3.482 3.512
3.490 3.467 3.498 3.519 3.504 3.469 3.497 3.495 3.518 3.523
3.458 3.478 3.443 3.500 3.449 3.525 3.461 3.489 3.514 3.470
3.561 3.506 3.444 3.479 3.524 3.531 3.501 3.495 3.443 3.458
3.481 3.497 3.461 3.513 3.528 3.496 3.533 3.450 3.516 3.476
3.512 3.550 3.441 3.541 3.569 3.531 3.468 3.564 3.522 3.520
3.505 3.523 3.475 3.470 3.457 3.536 3.528 3.477 3.536 3.491
3.510 3.461 3.431 3.502 3.491 3.506 3.439 3.513 3.496 3.539
3.469 3.481 3.515 3.535 3.460 3.575 3.488 3.515 3.484 3.482
3.517 3.483 3.467 3.467 3.502 3.471 3.516 3.474 3.500 3.466
;
```

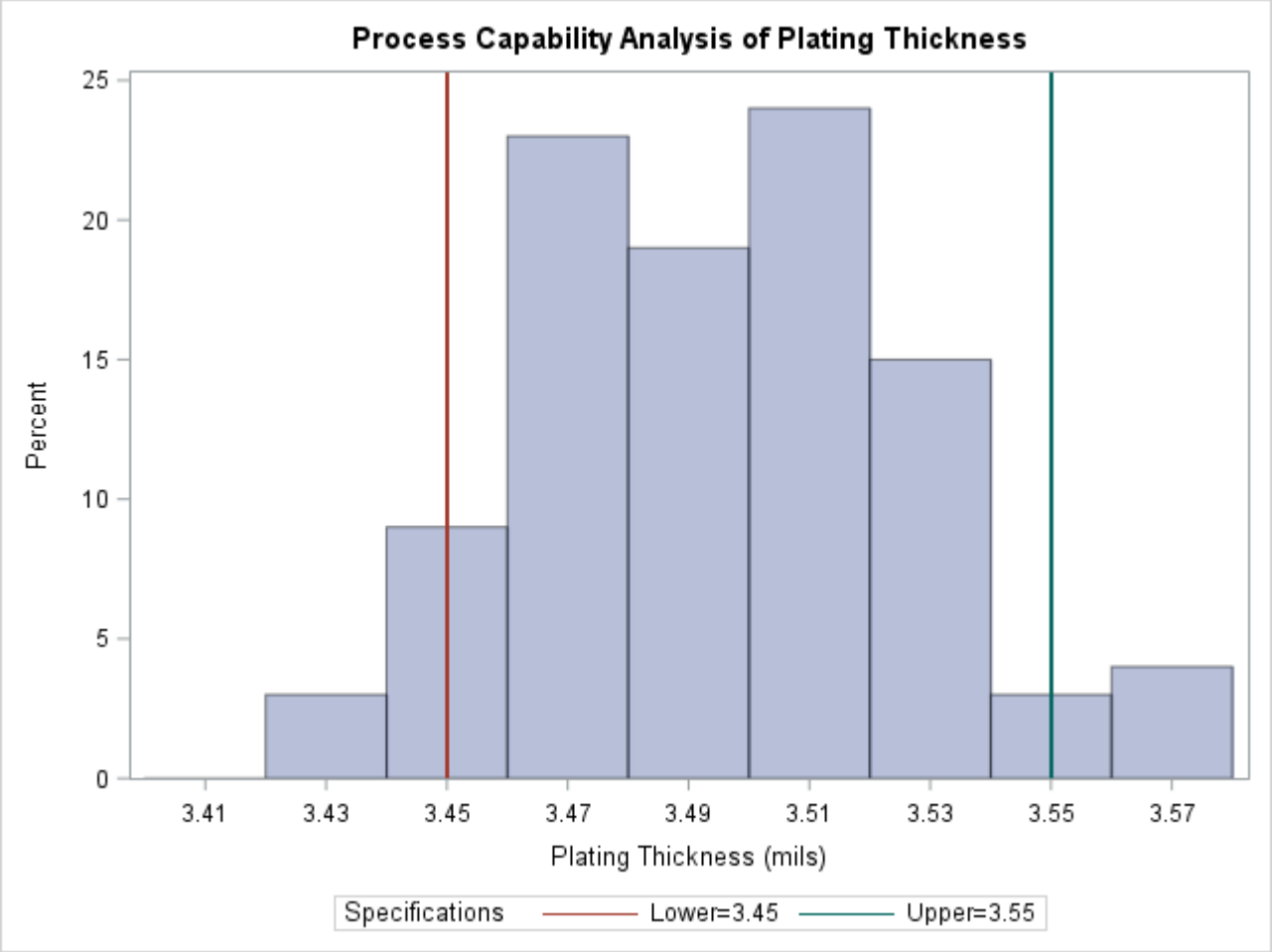
The following statements create the histogram shown in Figure 6.8:

```
title 'Process Capability Analysis of Plating Thickness';
proc capability data=Trans noprint;
   spec lsl = 3.45 usl = 3.55;
   histogram Thick / odstitle = title;
run;
```

A histogram is created for each variable listed after the keyword HISTOGRAM. The SPEC statement, which is optional, provides the specification limits that are displayed on the histogram. For more information about the SPEC statement, see "SPEC Statement" on page 214.

The NOPRINT option suppresses printed output with summary statistics for the variable Thick that would be displayed by default. See "Computing Descriptive Statistics" on page 197 for an example of this output.

Figure 6.8 Histogram Created with Traditional Graphics



Adding a Normal Curve to the Histogram

NOTE: See *Histogram with Fitted Normal Curve* in the SAS/QC Sample Library.

This example is a continuation of the preceding example.

The following statements fit a normal distribution from the thickness measurements and superimpose the fitted density curve on the histogram:

```
proc capability data=Trans;
   spec lsl = 3.45 usl = 3.55;
   histogram / normal;
run;
```

The ODS GRAPHICS ON statement specified before the PROC CAPABILITY statement enables ODS Graphics, so the histogram is created using ODS Graphics instead of traditional graphics.

The NORMAL option summarizes the fitted distribution in the printed output shown in Figure 6.9, and it specifies that the normal curve be displayed on the histogram shown in Figure 6.10.

Figure 6.9 Summary for Fitted Normal Distribution

Process Capability Analysis of Plating Thickness

The CAPABILITY Procedure
Fitted Normal Distribution for Thick (Plating Thickness (mils))

Parameters for Normal Distribution

Parameter	Symbol	Estimate
Mean	Mu	3.49533
Std Dev	Sigma	0.032117

Goodness-of-Fit Tests for Normal Distribution

Test		Statistic	DF		p Value
Kolmogorov-Smirnov	**D**	0.05563823		**Pr > D**	>0.150
Cramer-von Mises	**W-Sq**	0.04307548		**Pr > W-Sq**	>0.250
Anderson-Darling	**A-Sq**	0.27840748		**Pr > A-Sq**	>0.250
Chi-Square	**Chi-Sq**	6.96953022	5	**Pr > Chi-Sq**	0.223

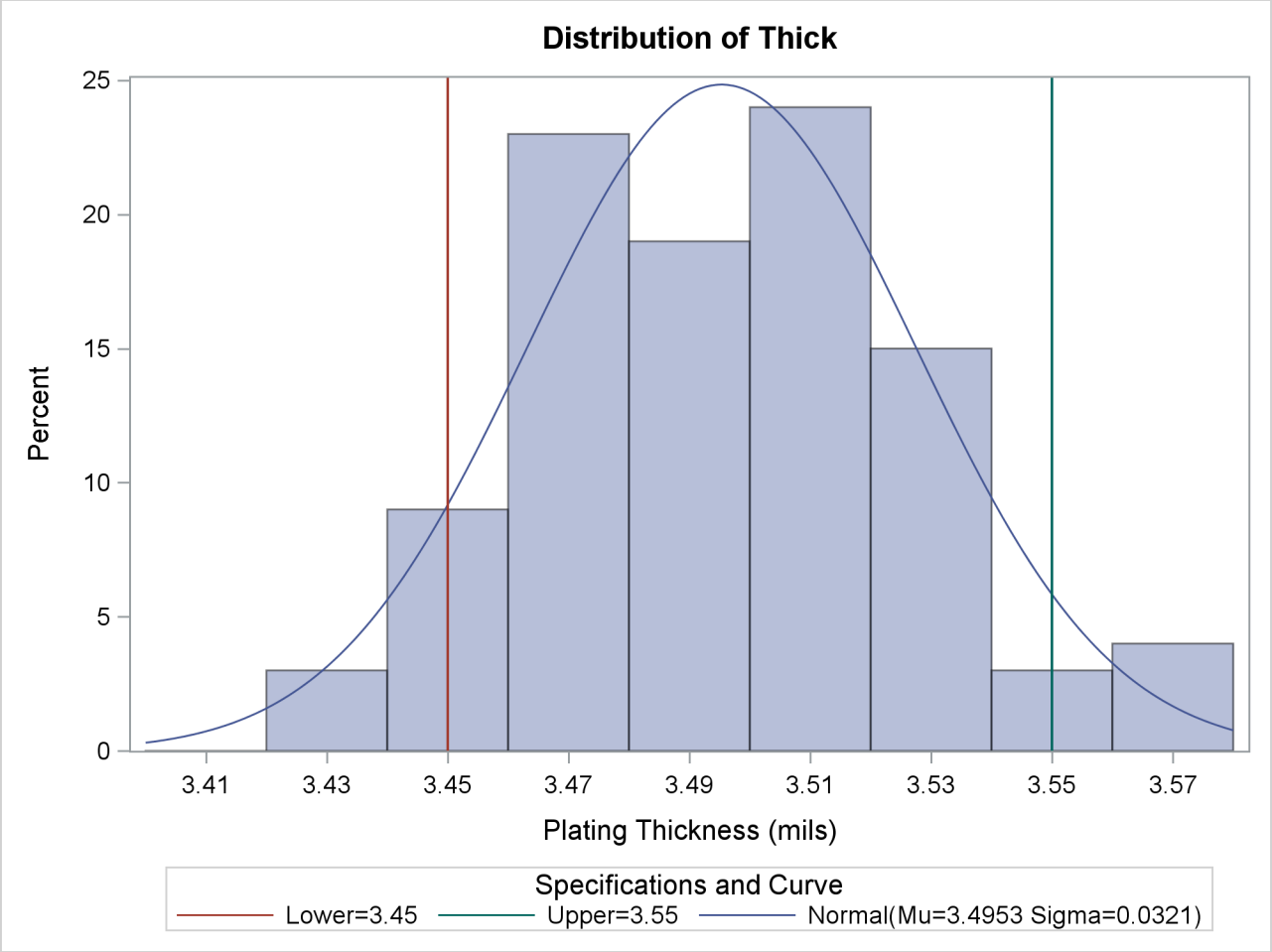
Percent Outside Specifications for Normal Distribution

Lower Limit		Upper Limit	
LSL	3.450000	**USL**	3.550000
Obs Pct < LSL	8.000000	**Obs Pct > USL**	5.000000
Est Pct < LSL	7.906248	**Est Pct > USL**	4.435722

Quantiles for Normal Distribution

	Quantile	
Percent	**Observed**	**Estimated**
1.0	3.42950	3.42061
5.0	3.44300	3.44250
10.0	3.45750	3.45417
25.0	3.46950	3.47367
50.0	3.49600	3.49533
75.0	3.51650	3.51699
90.0	3.53550	3.53649
95.0	3.55300	3.54816
99.0	3.57200	3.57005

Figure 6.10 Histogram Superimposed with Normal Curve



The printed output includes the following:

- parameters for the normal curve. The normal parameters μ and σ are estimated by the sample mean ($\hat{\mu} = 3.49533$) and the sample standard deviation ($\hat{\sigma} = 0.032117$).
- goodness-of-fit tests based on the empirical distribution function (EDF): the Anderson-Darling, Cramer-von Mises, and Kolmogorov-Smirnov tests. The p-values for these tests are greater than the usual cutoff values of 0.05 and 0.10, indicating that the thicknesses are normally distributed.
- a chi-square goodness-of-fit test. The p-value of 0.223 for this test indicates that the thicknesses are normally distributed. In general EDF tests (when available) are preferable to chi-square tests. See the section "EDF Goodness-of-Fit Tests" on page 351 for details.
- observed and estimated percentages outside the specification limits
- observed and estimated quantiles

For details, including formulas for the goodness-of-fit tests, see "Printed Output" on page 348. Note that the NOPRINT option in the PROC CAPABILITY statement suppresses only the printed output with summary statistics for the variable Thick. To suppress the printed output in Figure 6.9, specify the NOPRINT option enclosed in parentheses after the NORMAL option as in "Customizing a Histogram" on page 304.

The NORMAL option is one of many options that you can specify in the HISTOGRAM statement. See the section "Syntax: HISTOGRAM Statement" on page 305 for a complete list of options or the section "Dictionary of Options" on page 314 for detailed descriptions of options.

Customizing a Histogram

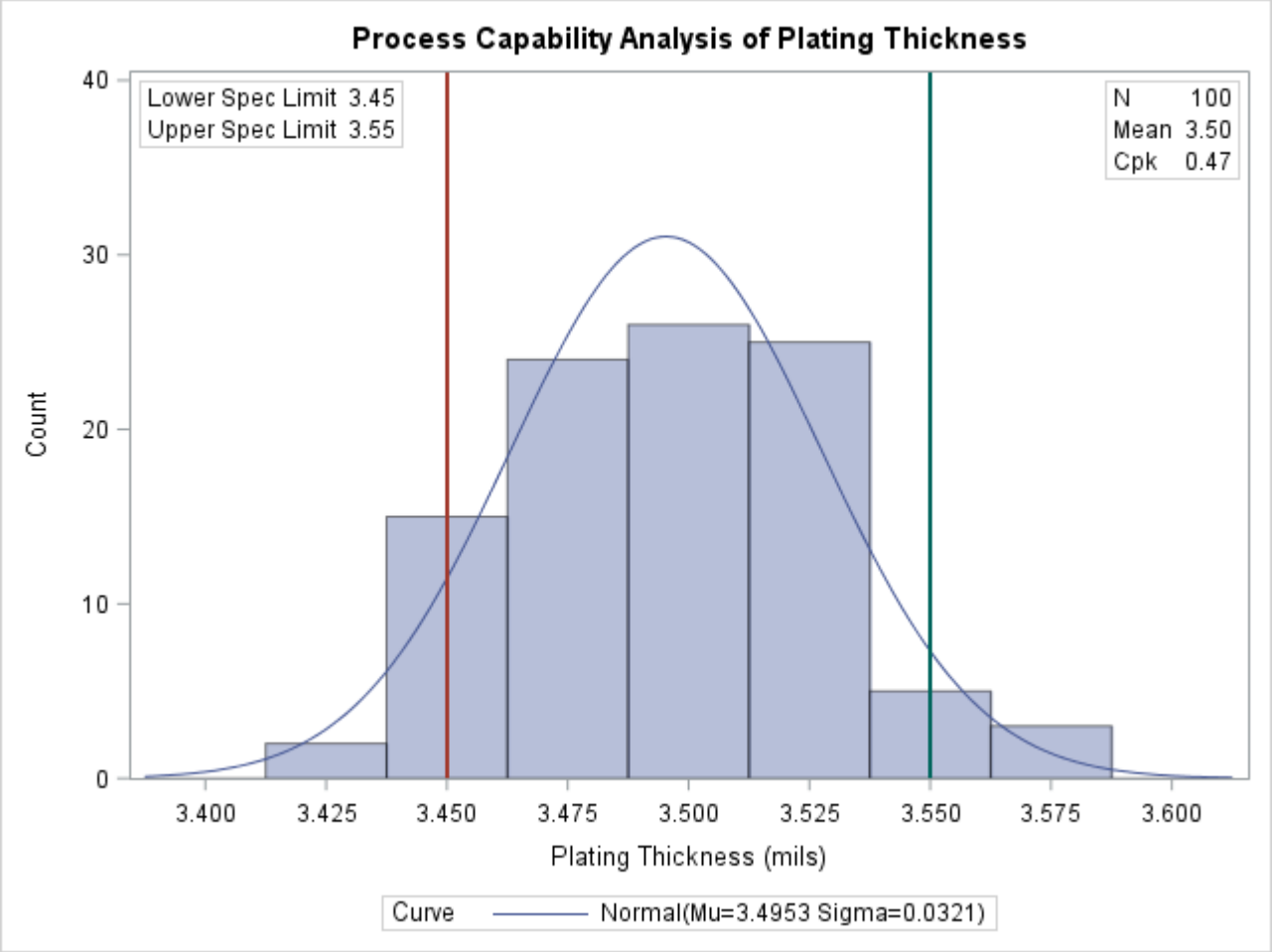
NOTE: See *Histogram with Fitted Normal Curve* in the SAS/QC Sample Library.

This example is a continuation of the preceding example. The following statements show how you can use HISTOGRAM statement options and INSET statements to customize a histogram:

```
title 'Process Capability Analysis of Plating Thickness';
proc capability data=Trans noprint;
   spec lsl = 3.45 usl = 3.55;
   histogram Thick / normal
                     midpoints = 3.4 to 3.6 by 0.025
                     vscale    = count
                     odstitle  = title
                     nospeclegend;
   inset lsl usl;
   inset n mean (5.2) cpk (5.2);
run;
```

The histogram is displayed in Figure 6.11.

Figure 6.11 Customizing the Appearance of the Histogram



The MIDPOINTS= option specifies a list of values to use as bin midpoints. The VSCALE=COUNT option requests a vertical axis scaled in counts rather than percents. The INSET statements inset the specification limits and summary statistics. The NOSPECLEGEND option suppress the default legend for the specification limits that is shown in Figure 6.8.

For more information about HISTOGRAM statement options, see the section "Dictionary of Options" on page 314. For details on the INSET statement, see "INSET Statement: CAPABILITY Procedure" on page 385.

Syntax: HISTOGRAM Statement

The syntax for the HISTOGRAM statement is as follows:

HISTOGRAM < *variables* > < / *options* > **;**

You can specify the keyword HIST as an alias for HISTOGRAM. You can use any number of HISTOGRAM statements after a PROC CAPABILITY statement. The components of the HISTOGRAM statement are described as follows.

variables

are the process variables for which histograms are to be created. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, the variables can be any numeric variables in the input data set. If you do not specify variables in a VAR statement or in the HISTOGRAM statement, then by default, a histogram is created for each numeric variable in the DATA= data set. If you use a VAR statement and do not specify any variables in the HISTOGRAM statement, then by default, a histogram is created for each variable listed in the VAR statement.

For example, suppose a data set named steel contains exactly two numeric variables named length and width. The following statements create two histograms, one for length and one for width:

```
proc capability data=steel;
   histogram;
run;
```

The following statements also create histograms for length and width:

```
proc capability data=steel;
   var length width;
   histogram;
run;
```

The following statements create a histogram for length only:

```
proc capability data=steel;
   var length width;
   histogram length;
run;
```

options

add features to the histogram. Specify all options after the slash (/) in the HISTOGRAM statement.

For example, in the following statements, the NORMAL option displays a fitted normal curve on the histogram, the MIDPOINTS= option specifies midpoints for the histogram, and the CTEXT= option specifies the color of the text:

```
proc capability data=steel;
   histogram length / normal
                      midpoints = 5.6 5.8 6.0 6.2 6.4
                      ctext     = yellow;
run;
```

Summary of Options

The following tables list the HISTOGRAM statement options by function. For detailed descriptions, see "Dictionary of Options" on page 314.

Parametric Density Estimation Options

Table 6.18 lists options that display a parametric density estimate on the histogram.

Table 6.18 Parametric Distribution Options

Option	Description
BETA(*beta-options*)	fits beta distribution with threshold parameter θ, scale parameter σ, and shape parameters α and β
EXPONENTIAL(*exponential-options*)	fits exponential distribution with threshold parameter θ and scale parameter σ
GAMMA(*gamma-options*)	fits gamma distribution with threshold parameter θ, scale parameter σ, and shape parameter α
GUMBEL(*Gumbel-options*)	plots Gumbel distribution with location parameter μ and scale parameter σ
IGAUSS(*iGauss-options*)	plots inverse Gaussian distribution with mean μ and shape parameter λ
LOGNORMAL(*lognormal-options*)	fits lognormal distribution with threshold parameter θ, scale parameter ζ, and shape parameter σ
NORMAL(*normal-options*)	fits normal distribution with mean μ and standard deviation σ
PARETO(*Pareto-options*)	plots Pareto distribution with threshold parameter θ, scale parameter σ, and shape parameter α
POWER(*power-options*)	plots power function distribution with threshold parameter θ, scale parameter σ, and shape parameter α
RAYLEIGH(*Rayleigh-options*)	plots Rayleigh distribution with threshold parameter θ and scale parameter σ
SB(*SB-options*)	fits Johnson S_B distribution with threshold parameter θ, scale parameter σ, and shape parameters δ and γ
SU(*SU-options*)	fits Johnson S_U distribution with location parameter θ, scale parameter σ, and shape parameters δ and γ
WEIBULL(*Weibull-options*)	fits Weibull distribution with threshold parameter θ, scale parameter σ, and shape parameter c

Table 6.19 lists secondary options that specify parameters for fitted parametric distributions and that control the display of fitted curves. Specify these secondary options in parentheses after the distribution keyword. For example, the following statements fit a normal curve by using the NORMAL option:

```
proc capability;
   histogram / normal(color=red mu=10 sigma=0.5);
run;
```

The COLOR= *normal-option* draws the curve in red, and the MU= and SIGMA= *normal-options* specify the parameters $\mu = 10$ and $\sigma = 0.5$ for the curve. Note that the sample mean and sample standard deviation are used to estimate μ and σ, respectively, when the MU= and SIGMA= options are not specified.

You can specify lists of values for distribution parameters to display more than one fitted curve from the same distribution family on a histogram. Option values are matched by list position. You can specify the value *EST* in a list of distribution parameter values to use an estimate of the parameter.

For example, the following code displays two normal curves on a histogram:

```
proc capability;
   histogram / normal(color=(red blue) mu=10 est sigma=0.5 est);
run;
```

The first curve is red, with $\mu = 10$ and $\sigma = 0.5$. The second curve is blue, with μ equal to the sample mean and σ equal to the sample standard deviation.

See the section "Formulas for Fitted Curves" on page 336 for detailed information about the families of parametric distributions that you can fit with the HISTOGRAM statement.

Table 6.19 Distribution Options

Option	Description
Options Used with All Parametric Distributions	
COLOR=	specifies color of fitted density curve
FILL	fills area under fitted density curve
INDICES	calculates capability indices based on fitted distribution
L=	specifies line type of fitted curve
MIDPERCENTS	prints table of midpoints of histogram intervals
NOPRINT	suppresses printed output summarizing fitted curve
PERCENTS=	lists percents for which quantiles calculated from data and quantiles estimated from fitted curve are tabulated
SYMBOL=	specifies character used for fitted density curve in line printer plots
W=	specifies width of fitted density curve
Beta-Options	
ALPHA=	specifies first shape parameter α for fitted beta curve
BETA=	specifies second shape parameter β for fitted beta curve
SIGMA=	specifies scale parameter σ for fitted beta curve
THETA=	specifies lower threshold parameter θ for fitted beta curve
Exponential-Options	
SIGMA=	specifies scale parameter σ for fitted exponential curve
THETA=	specifies threshold parameter θ for fitted exponential curve
Gamma-Options	
ALPHA=	specifies shape parameter α for fitted gamma curve
ALPHADELTA=	specifies change in successive estimates of α at which the Newton-Raphson approximation of $\hat{\alpha}$ terminates

Table 6.19 (*continued*)

Option	Description
ALPHAINITIAL=	specifies initial value for α in Newton-Raphson approximation of $\hat{\alpha}$
MAXITER=	specifies maximum number of iterations in Newton-Raphson approximation of $\hat{\alpha}$
SIGMA=	specifies scale parameter σ for fitted gamma curve
THETA=	specifies threshold parameter θ for fitted gamma curve
Gumbel-Options	
EDFNSAMPLES=	specifies number of samples for EDF goodness-of-fit simulation
EDFSEED=	specifies seed value for EDF goodness-of-fit simulation
MU=	specifies location parameter μ for fitted Gumbel curve
SIGMA=	specifies scale parameter σ for fitted Gumbel curve
IGauss-Options	
EDFNSAMPLES=	specifies number of samples for EDF goodness-of-fit simulation
EDFSEED=	specifies seed value for EDF goodness-of-fit simulation
LAMBDA=	specifies shape parameter λ for fitted inverse Gaussian curve
MU=	specifies mean μ for fitted inverse Gaussian curve
Lognormal-Options	
SIGMA=	specifies shape parameter σ for fitted lognormal curve
THETA=	specifies threshold parameter θ for fitted lognormal curve
ZETA=	specifies scale parameter ζ for fitted lognormal curve
Normal-Options	
MU=	specifies mean μ for fitted normal curve
SIGMA=	specifies standard deviation σ for fitted normal curve
Pareto-Options	
ALPHA=	specifies shape parameter α for fitted Pareto curve
EDFNSAMPLES=	specifies number of samples for EDF goodness-of-fit simulation
EDFSEED=	specifies seed value for EDF goodness-of-fit simulation
SIGMA=	specifies scale parameter σ for fitted Pareto curve
THETA=	specifies threshold parameter θ for fitted Pareto curve
Power-Options	
ALPHA=	specifies shape parameter α for fitted power function curve
SIGMA=	specifies scale parameter σ for fitted power function curve
THETA=	specifies threshold parameter θ for fitted power function curve
Rayleigh-Options	
EDFNSAMPLES=	specifies number of samples for EDF goodness-of-fit simulation
EDFSEED=	specifies seed value for EDF goodness-of-fit simulation
SIGMA=	specifies scale parameter σ for fitted Rayleigh curve
THETA=	specifies threshold parameter θ for fitted Rayleigh curve
S_B**-Options**	
DELTA=	specifies first shape parameter δ for fitted S_B curve
FITINTERVAL=	specifies z-value for method of percentiles
FITMETHOD=	specifies method of parameter estimation
FITTOLERANCE=	specifies tolerance for method of percentiles
GAMMA=	specifies second shape parameter γ for fitted S_B curve
SIGMA=	specifies scale parameter σ for fitted S_B curve

Table 6.19 (*continued*)

Option	Description
THETA=	specifies lower threshold parameter θ for fitted S_B curve
S_U-Options	
DELTA=	specifies first shape parameter δ for fitted S_U curve
FITINTERVAL=	specifies z-value for method of percentiles
FITMETHOD=	specifies method of parameter estimation
FITTOLERANCE=	specifies tolerance for method of percentiles
GAMMA=	specifies second shape parameter γ for fitted S_U curve
OPTBOUNDRANGE=	specifies the sampling range for parameter starting values in MLE optimization
OPTMAXITER=	specifies an iteration limit for MLE optimization
OPTMAXSTARTS=	specifies the maximum number of starting points to be used for MLE optimization
OPTPRINT	prints an iteration history for MLE optimization
OPTSEED=	specifies a seed value for MLE optimization
OPTTOLERANCE=	specifies the optimality tolerance for MLE optimization
SIGMA=	specifies scale parameter σ for fitted S_U curve
THETA=	specifies location parameter θ for fitted S_U curve
Weibull-Options	
C=	specifies shape parameter c for fitted Weibull curve
CDELTA=	specifies change in successive estimates of c at which the Newton-Raphson approximation of $\hat{c}$ terminates
CINITIAL=	specifies initial value for c in Newton-Raphson approximation of $\hat{c}$
MAXITER=	specifies maximum number of iterations in Newton-Raphson approximation of $\hat{c}$
SIGMA=	specifies scale parameter σ for fitted Weibull curve
THETA=	specifies threshold parameter θ for fitted Weibull curve

Nonparametric Density Estimation Options

Table 6.20 Kernel Density Estimation Options

Option	Description
KERNEL(*kernel-options*)	fits kernel density estimates

Specify the options listed in Table 6.21 in parentheses after the keyword KERNEL to control features of kernel density estimates requested with the KERNEL option.

Table 6.21 Kernel-Options

Option	Description
C=	specifies standardized bandwidth parameter c for fitted kernel density estimate
COLOR=	specifies color of the fitted kernel density curve
FILL	fills area under fitted kernel density curve
K=	specifies type of kernel function
L=	specifies line type used for fitted kernel density curve
LOWER=	specifies lower bound for fitted kernel density curve
SYMBOL=	specifies character used for fitted kernel density curve in line printer plots
UPPER=	specifies upper bound for fitted kernel density curve
W=	specifies line width for fitted kernel density curve

General Options

Table 6.22 summarizes general options for the HISTOGRAM statement, including options for enhancing charts and producing output data sets.

Table 6.22 General HISTOGRAM Statement Options

Option	Description
Options to Create Output Data Sets	
OUTFIT=	requests information about fitted curves
OUTHISTOGRAM=	requests information about histogram intervals
OUTKERNEL=	creates a data set containing kernel density estimates
General Histogram Layout Options	
CLIPCURVES	scales vertical axis without considering fitted curves
CONTENTS=	specifies table of contents entry for histogram grouping
CURVELEGEND=	specifies LEGEND statement for curves
ENDPOINTS=	lists endpoints for histogram intervals
HANGING	constructs hanging histogram
HREF=	specifies reference lines perpendicular to the horizontal axis
HREFLABELS=	specifies labels for HREF= lines
MIDPERCENTS	prints table of histogram intervals
MIDPOINTS=	lists midpoints for histogram intervals
NENDPOINTS=	specifies number of histogram interval endpoints
NMIDPOINTS=	specifies number of histogram interval midpoints
NOBARS	suppresses histogram bars
NOCURVELEGEND	suppresses legend for curves
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOPLOT	suppresses plot
NOSPECLEGEND	suppresses specifications legend
NOTABCONTENTS	suppresses table of contents entries for tables produced by HISTOGRAM statement

Table 6.22 (*continued*)

Option	Description
RTINCLUDE	includes right endpoint in interval
SPECLEGEND=	specifies LEGEND statement for specification limits
VREF=	specifies reference lines perpendicular to the vertical axis
VREFLABELS=	specifies labels for VREF= lines
VSCALE=	specifies scale for vertical axis
Options to Enhance Graphical Output	
ANNOTATE=	specifies annotate data set
BARLABEL=	produces labels above histogram bars
BARWIDTH=	specifies width for the bars
BMCFILL=	specifies fill color for box-and-whisker plot in bottom margin
BMCFRAME=	specifies fill color bottom margin plot frame
BMCOLOR=	specifies color for bottom margin plot
BMMARGIN=	specifies height of margin for bottom margin plot
BMPLOT=	requests a plot in bottom margin of histogram
CAXIS=	specifies color for axis
CBARLINE=	specifies color for outlines of histogram bars
CFILL=	specifies color for filling under curve
CFRAME=	specifies color for frame
CGRID=	specifies color for grid lines
CHREF=	specifies colors for HREF= lines
CLIPREF	draws reference lines behind histogram bars
CLIPSPEC=	clips histogram bars at specification limits
CSTATREF=	specifies colors for STATREF= lines
CTEXT=	specifies color for text
CVREF=	specifies colors for VREF= lines
DESCRIPTION=	specifies description for plot in graphics catalog
FONT=	specifies software font for text
FRONTREF	draws reference lines in front of histogram bars
GRID	creates a grid
HAXIS=	specifies AXIS statement for horizontal axis
HEIGHT=	specifies height of text used outside framed areas
HMINOR=	specifies number of horizontal minor tick marks
HOFFSET=	specifies offset for horizontal axis
HREFLABPOS=	specifies vertical position of labels for HREF= lines
INFONT=	specifies software font for text inside framed areas
INHEIGHT=	specifies height of text inside framed areas
INTERBAR=	specifies space between histogram bars
LEGEND=	identifies LEGEND statement
LGRID=	specifies a line type for grid lines
LHREF=	specifies line styles for HREF= lines
LSTATREF=	specifies line styles for STATREF= lines
LVREF=	specifies line styles for VREF= lines
MAXNBIN=	specifies maximum number of bins to display

Table 6.22 (*continued*)

Option	Description
MAXSIGMAS=	limits the number of bins that display to within a specified number of standard deviations above and below mean of data in key cell
MIDPOINTS=	specifies midpoints for histogram intervals
NAME=	specifies name for plot in graphics catalog
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
PFILL=	specifies pattern for filling under curve
STATREF=	specifies reference lines at values of summary statistics
STATREFLABELS=	specifies labels for STATREF= lines
STATREFSUBCHAR=	specifies substitution character for displaying statistic values in STATREFLABELS= labels
TURNVLABELS	turns and vertically strings out characters in labels for vertical axis
VAXIS=	specifies AXIS statement or values for vertical axis
VAXISLABEL=	specifies label for vertical axis
VMINOR=	specifies number of vertical minor tick marks
VOFFSET=	specifies length of offset at upper end of vertical axis
VREFLABPOS=	specifies horizontal position of labels for VREF= lines
WAXIS=	specifies line thickness for axes and frame
WBARLINE=	specifies line thickness for bar outlines
WGRID=	specifies line thickness for grid
Options for ODS Graphics Output	
ODSFOOTNOTE=	specifies footnote displayed on histogram
ODSFOOTNOTE2=	specifies secondary footnote displayed on histogram
ODSTITLE=	specifies title displayed on histogram
ODSTITLE2=	specifies secondary title displayed on histogram
Options for Comparative Plots	
ANNOKEY	applies annotation requested in ANNOTATE= data set to key cell only
CFRAMESIDE=	specifies color for filling frame for row labels
CFRAMETOP=	specifies color for filling frame for column labels
CPROP=	specifies color for proportion of frequency bar
CTEXTSIDE=	specifies color for row labels of comparative histograms
CTEXTTOP=	specifies color for column labels of comparative histograms
INTERTILE=	specifies distance between tiles
NCOLS=	specifies number of columns in comparative histogram
NROWS=	specifies number of rows in comparative histogram
OVERLAY	overlays plots for different class levels (ODS Graphics only)
Options to Enhance Line Printer Plots	
HREFCHAR=	specifies line character for HREF= lines
VREFCHAR=	specifies line character for VREF= lines

Dictionary of Options

The following sections provide detailed descriptions of options specific to the HISTOGRAM statement. See "Dictionary of Common Options: CAPABILITY Procedure" on page 534 for detailed descriptions of options common to all the plot statements.

General Options

ALPHA=*value-list*

specifies the shape parameter α for fitted curves requested with the BETA, GAMMA, PARETO, and POWER options. Enclose the ALPHA= option in parentheses after the distribution keyword. If you do not specify a value for α, the procedure calculates a maximum likelihood estimate. See Example 6.8. You can specify A= as an alias for ALPHA= if you use it as a *beta-option*. You can specify SHAPE= as an alias for ALPHA= if you use it as a *gamma-option*.

BARLABEL=COUNT | PERCENT | PROPORTION

displays labels above the histogram bars. If you specify BARLABEL=COUNT, the label shows the number of observations associated with a given bar. BARLABEL=PERCENT shows the percent of observations represented by that bar. If you specify BARLABEL=PROPORTION, the label displays the proportion of observations associated with the bar.

BETA<(*beta-options***)>**

displays a fitted beta density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\sigma+\theta-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} hv & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and

θ = lower threshold parameter (lower endpoint parameter)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
β = shape parameter ($\beta > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The beta distribution is bounded below by the parameter θ and above by the value $\theta+\sigma$. You can specify θ and σ by using the THETA= and SIGMA= *beta-options*. The following statements fit a beta distribution bounded between 50 and 75 by using maximum likelihood estimates for α and β:

```
proc capability;
   histogram length / beta(theta=50 sigma=25);
run;
```

In general, the default values for THETA= and SIGMA= are 0 and 1, respectively. You can specify THETA=EST and SIGMA=EST to request maximum likelihood estimates for θ and σ.

The beta distribution has two shape parameters, α and β. If these parameters are known, you can specify their values with the ALPHA= and BETA= *beta-options*. If you do not specify values, the procedure calculates maximum likelihood estimates for α and β.

The BETA option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the BETA option. See Example 6.8. Also see "Formulas for Fitted Curves" on page 336.

BETA=*value-list*

B=*value-list*

specifies the second shape parameter β for beta density curves requested with the BETA option. Enclose the BETA= option in parentheses after the BETA option. If you do not specify a value for β, the procedure calculates a maximum likelihood estimate. See Example 6.8.

BMPLOT=CARPET | DOTPLOT | SKELETAL | SCHEMATIC

produces a carpet plot, dot plot, or box-and-whisker plot along the bottom margin of a histogram. A carpet plot or dot plot shows the distribution of individual observations along the histogram's horizontal axis. A carpet plot represents each observation with a vertical line. A dot plot marks each observation with a symbol. A box-and-whisker plot gives a summary of the data distribution that a histogram alone does not provide. The left and right edges of the box are located at the first and third quartiles. A central vertical line is drawn at the median and a symbol is plotted inside the box at the mean. If you specify the SKELETAL keyword, a box-and-whisker plot is produced with whiskers extending to the minimum and maximum values. If you specify SCHEMATIC, a *schematic* box-and-whisker plot is produced. In a schematic box-and-whisker plot, the whiskers extend to the smallest value within the *lower fence* and the largest value within the *upper fence*. Fences are defined in terms of the interquartile range (IQR). The lower fence is 1.5 IQR below the first quartile and the upper fence is 1.5 IQR above the third quartile. Each observation outside the fences is plotted with a symbol.

C=*value-list*

specifies the shape parameter c for Weibull density curves requested with the WEIBULL option. Enclose the C= option in parentheses after the WEIBULL option. If you do not specify a value for c, the procedure calculates a maximum likelihood estimate. See Example 6.9. You can specify the SHAPE= option as an alias for the C= option.

C=*value-list* **| MISE**

specifies the standardized bandwidth parameter c for kernel density estimates requested with the KERNEL option. Enclose the C= option in parentheses after the KERNEL option. You can specify up to five values to request multiple estimates. You can also specify the C=MISE option, which produces the estimate with a bandwidth that minimizes the approximate mean integrated square error (MISE). For example, the following statements compute three density estimates:

```
proc capability;
   histogram length / kernel(c=0.5 1.0 mise);
run;
```

The first two estimates have standardized bandwidths of 0.5 and 1.0, respectively, and the third has a bandwidth that minimizes the approximate MISE.

You can also use the C= option with the K= option, which specifies the kernel function, to compute multiple estimates. If you specify more kernel functions than bandwidths, the last bandwidth in the list is repeated for the remaining estimates. Likewise, if you specify more bandwidths than kernel functions, the last kernel function is repeated for the remaining estimates. For example, the following statements compute three density estimates:

```
proc capability;
   histogram length / kernel(c=1 2 3 k=normal quadratic);
run;
```

The first uses a normal kernel and a bandwidth of 1, the second uses a quadratic kernel and a bandwidth of 2, and the third uses a quadratic kernel and a bandwidth of 3. See Example 6.12.

If you do not specify a value for c, the bandwidth that minimizes the approximate MISE is used for all the estimates.

CLIPCURVES

scales the vertical axis without taking fitted curves into consideration. Curves that extend above the tallest histogram bar may be clipped. You can use this option to avoid compression of the histogram bars due to extremely high fitted curve peaks.

DELTA=*value-list*

specifies the first shape parameter δ for Johnson S_B and Johnson S_U density curves requested with the SB and SU options. Enclose the DELTA= option in parentheses after the SB or SU option. If you do not specify a value for δ, the procedure calculates an estimate.

EDFNSAMPLES=*value*

specifies the number of simulation samples used to compute p-values for EDF goodness-of-fit statistics for density curves requested with the GUMBEL, IGAUSS, PARETO, and RAYLEIGH options. Enclose the EDFNSAMPLES= option in parentheses after the distribution option. The default value is 500.

EDFSEED=*value*

specifies an integer value used to start the pseudo-random number generator when creating simulation samples for computing EDF goodness-of-fit statistic p-values for density curves requested with the GUMBEL, IGAUSS, PARETO, and RAYLEIGH options. Enclose the EDFSEED= option in parentheses after the distribution option. By default, the procedure uses a random number seed generated from reading the time of day from the computer's clock.

ENDPOINTS

ENDPOINTS=*value-list*

specifies that histogram interval endpoints, rather than midpoints, are aligned with horizontal axis tick marks. If you specify ENDPOINTS, the number of histogram intervals is based on the number of observations by using the method of Terrell and Scott (1985). If you specify ENDPOINTS=*value-list*, the *values* must be listed in increasing order and must be evenly spaced. All observations in the input data set, as well as any specification limits, must lie between the first and last values specified. The same *value-list* is used for all variables.

EXPONENTIAL< (*exponential-options*) >

EXP< (*exponential-options*) >

displays a fitted exponential density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{hv}{\sigma} \exp(-(\frac{x-\theta}{\sigma})) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The parameter θ must be less than or equal to the minimum data value. You can specify θ with the THETA= *exponential-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ with the SIGMA= *exponential-option*. By default, a maximum likelihood estimate is computed for σ. For example, the following statements fit an exponential curve with $\theta = 10$ and with a maximum likelihood estimate for σ:

```
proc capability;
   histogram / exponential(theta=10 l=2 color=red);
run;
```

The curve is red and has a line type of 2. The EXPONENTIAL option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the EXPONENTIAL option. See "Formulas for Fitted Curves" on page 336.

FILL

fills areas under a parametric density curve or kernel density estimate with colors and patterns. Enclose the FILL option in parentheses after a curve option or the KERNEL option, as in the following statements:

```
proc capability;
   histogram length / normal(fill) cfill=green pfill=solid;
run;
```

Depending on the area to be filled (outside or between the specification limits), you can specify the color and pattern with options in the SPEC statement and HISTOGRAM statement, as summarized in the following table:

Area Under Curve	Statement	Option
between specification	HISTOGRAM	CFILL=
limits	HISTOGRAM	PFILL=
left of lower	SPEC	CLEFT=
specification limit	SPEC	PLEFT=
right of upper	SPEC	CRIGHT=
specification limit	SPEC	PRIGHT=

If you do not display specification limits, the CFILL= and PFILL= options specify the color and pattern for the entire area under the curve. Solid fills are used by default if patterns are not specified. You can specify the FILL option with only one fitted curve. For an example, see Output 6.8.1. Refer to *SAS/GRAPH: Reference* for a list of available patterns and colors. If you do not specify the FILL option but specify the options in the preceding table, the colors and patterns are applied to the corresponding areas under the histogram.

FITINTERVAL=*value*

specifies the value of z for the method of percentiles when this method is used to fit a Johnson S_B or Johnson S_U distribution. The FITINTERVAL= option is specified in parentheses after the SB or SU option. The default of z is 0.524.

FITMETHOD=PERCENTILE | MLE | MOMENTS

specifies the method used to estimate the parameters of a Johnson S_B or Johnson S_U distribution. The FITMETHOD= option is specified in parentheses after the SB or SU option. By default, the method of percentiles is used. You can specify the MLE keyword to request maximum likelihood estimation. The OPTBOUNDRANGE=, OPTMAXITER=, OPTMAXSTARTS=, OPTPRINT, OPTSEED=, and OPTTOLERANCE= options control the optimizer that performs the maximum likelihood calculation.

FITTOLERANCE=*value*

specifies the tolerance value for the ratio criterion when the method of percentiles is used to fit a Johnson S_B or Johnson S_U distribution. The FITTOLERANCE= option is specified in parentheses after the SB or SU option. The default value is 0.01.

GAMMA< (*gamma-options***) >**

displays a fitted gamma density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{hv}{\Gamma(\alpha)\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The parameter θ for the gamma distribution must be less than the minimum data value. You can specify θ with the THETA= *gamma-option*. The default value for θ is 0. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, the gamma distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *gamma-options*. By default, maximum likelihood estimates are computed for α and σ. For example, the following statements fit a gamma curve with $\theta = 4$ and with maximum likelihood estimates for α and σ:

```
proc capability;
   histogram length / gamma(theta=4);
run;
```

Note that the maximum likelihood estimate of α is calculated iteratively using the Newton-Raphson approximation. The ALPHADELTA=, ALPHAINITIAL=, and MAXITER= *gamma-options* control the approximation.

The GAMMA option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the GAMMA option. See Example 6.9 and "Formulas for Fitted Curves" on page 336.

GAMMA=*value-list*

specifies the second shape parameter γ for Johnson S_B and Johnson S_U density curves requested with the SB and SU options. Enclose the GAMMA= option in parentheses after the SB or SU option. If you do not specify a value for γ, the procedure calculates an estimate.

GRID

adds a grid to the histogram. Grid lines are horizontal lines positioned at major tick marks on the vertical axis.

GUMBEL<(*Gumbel-options***)>**

displays a fitted Gumbel (also known as Type 1 extreme value distribution) density curve on the histogram. The curve equation is

$$p(x) = \frac{hv}{\sigma} e^{-(x-\mu)/\sigma} \exp\left(-e^{-(x-\mu)/\sigma}\right)$$

where

μ = location parameter
σ = scale parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

You can specify values for μ and σ with the MU= and SIGMA= *Gumbel-options*. By default, maximum likelihood estimates are computed for μ and σ.

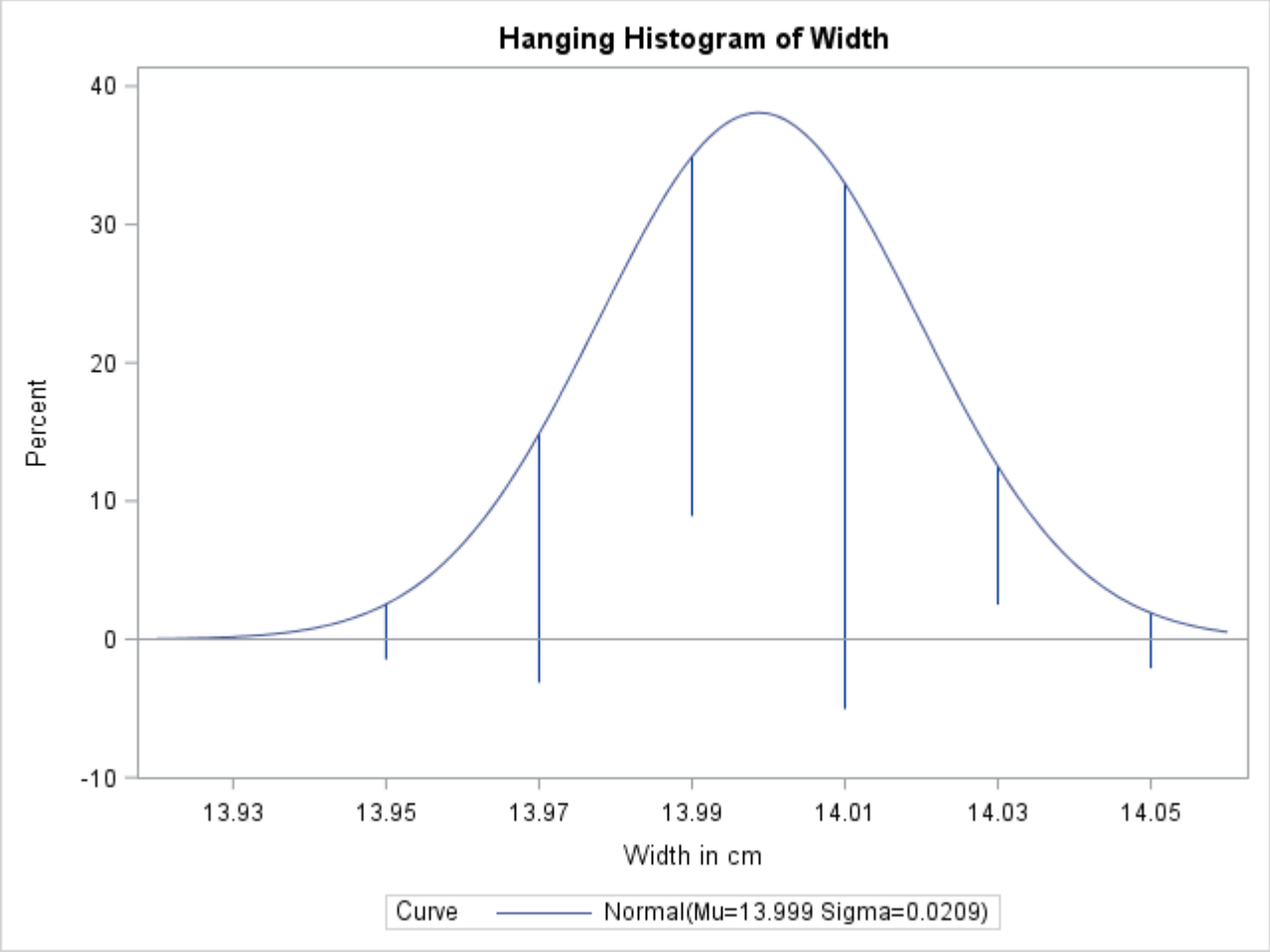
The GUMBEL option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the GUMBEL option. See "Formulas for Fitted Curves" on page 336.

HANGING

HANG

requests a hanging histogram , as illustrated in Figure 6.12.

Figure 6.12 Hanging Histogram



You can use the HANGING option with only one fitted density curve. A hanging histogram aligns the tops of the histogram bars (displayed as lines) with the fitted curve. The lines are positioned at the midpoints of the histogram bins. A hanging histogram is a goodness-of-fit diagnostic in the sense that the closer the lines are to the horizontal axis, the better the fit. Hanging histograms are discussed by Tukey (1977), Wainer (1974), and Velleman and Hoaglin (1981).

IGAUSS< (*iGauss-options*) >

displays a fitted inverse Gaussian density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} hv\left(\frac{\lambda}{2\pi x^3}\right)^{1/2} \exp(-\frac{\lambda}{2\mu^2 x}(x-\mu)^2) & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

μ = mean parameter ($\mu > 0$)
λ = shape parameter ($\lambda > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

You can specify values for μ and λ with the MU= and LAMBDA= *iGauss-options*. By default, the sample mean is used for μ and a maximum likelihood estimate is computed for λ.

The IGAUSS option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the IGAUSS option. See "Formulas for Fitted Curves" on page 336.

INDICES

requests capability indices based on the fitted distribution. Enclose the keyword INDICES in parentheses after the distribution keyword. See "Indices Using Fitted Curves" on page 354 for computational details and see Output 6.11.2.

K=NORMAL | QUADRATIC | TRIANGULAR

specifies the kernel function (normal, quadratic, or triangular) used to compute a kernel density estimate. Enclose the K= option in parentheses after the KERNEL option, as in the following statements:

```
proc capability;
   histogram length / kernel(k=quadratic);
run;
```

You can specify kernel functions for up to five estimates. You can also use the K= option together with the C= option, which specifies standardized bandwidths. If you specify more kernel functions than bandwidths, the last bandwidth in the list is repeated for the remaining estimates. Likewise, if you specify more bandwidths than kernel functions, the last kernel function is repeated for the remaining estimates. For example, the following statements compute three estimates with bandwidths of 0.5, 1.0, and 1.5:

```
proc capability;
   histogram length / kernel(c=0.5 1.0 1.5 k=normal quadratic);
run;
```

The first estimate uses a normal kernel, and the last two estimates use a quadratic kernel. By default, a normal kernel is used.

KERNEL<(*kernel-options*)>

superimposes up to five kernel density estimates on the histogram. You can specify the *kernel-options* described in the following table:

Option	Description
C=	specifies the smoothing parameter
COLOR=	specifies the color of the curve
FILL	specifies that the area under the curve is to be filled
K=	specifies the type of kernel function
L=	specifies the line style for the curve
LOWER=	specifies the lower bound for the curve
SYMBOL=	specifies the character used for the kernel density curve in line printer plots
UPPER=	specifies the upper bound for the curve
W=	specifies the width of the curve

You can request multiple kernel density estimates on the same histogram by specifying a list of values for either the C= or K= option. For more information, see the entries for these options. Also see Output 6.6.1 and "Kernel Density Estimates" on page 347. By default, kernel density estimates are computed using the AMISE method.

LAMBDA=*value*

specifies the shape parameter λ for fitted curves requested with the IGAUSS option. Enclose the LAMBDA= option in parentheses after the IGAUSS distribution keyword. If you do not specify a value for λ, the procedure calculates a maximum likelihood estimate.

LOGNORMAL< (*lognormal-options***) >**

displays a fitted lognormal density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{hv}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = threshold parameter
ζ = scale parameter
σ = shape parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

Note that the lognormal distribution is also referred to as the S_L distribution in the Johnson system of distributions.

The parameter θ for the lognormal distribution must be less than the minimum data value. You can specify θ with the THETA= *lognormal-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify the parameters σ and ζ with the SIGMA= and ZETA= *lognormal-options*. By default, maximum likelihood estimates are computed for σ and ζ. For example, the following statements fit a lognormal distribution function with a default value of $\theta = 0$ and with maximum likelihood estimates for σ and ζ:

```
proc capability;
   histogram length / lognormal;
run;
```

The LOGNORMAL option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options that you can specify with the LOGNORMAL option. See Example 6.9 and "Formulas for Fitted Curves" on page 336.

LOWER=*value-list*

specifies lower bounds for kernel density estimates requested with the KERNEL option. Enclose the LOWER= option in parentheses after the KERNEL option. You can specify up to five lower bounds for multiple kernel density estimates. If you specify more kernel estimates than lower bounds, the last lower bound is repeated for the remaining estimates.

MAXNBIN=*n*

specifies the maximum number of bins to be displayed in a comparative histogram. This option is useful in situations where the scales or ranges of the data distributions differ greatly from cell to cell. By default, the bin size and midpoints are determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. However, if the cell scales differ considerably, the resulting number of bins may be so great that each cell histogram is scaled into a narrow region. By limiting the number of bins with the MAXNBIN= option, you can narrow the window about the data distribution in the key cell. Note that the MAXNBIN= option provides an alternative to the MAXSIGMAS= option.

MAXSIGMAS=*value*

limits the number of bins to be displayed to a range of *value* standard deviations (of the data in the key cell) above and below the mean of the data in the key cell. This option is useful in situations where the scales or ranges of the data distributions differ greatly from cell to cell. By default, the bin size and midpoints are determined for the key cell, and then the midpoint list is extended to accommodate the data ranges for the remaining cells. If the cell scales differ considerably, however, the resulting number of bins may be so great that each cell histogram is scaled into a narrow region. By limiting the number of bins with the MAXSIGMAS= option, you narrow the window about the data distribution in the key cell. Note that the MAXSIGMAS= option provides an alternative to the MAXNBIN= option.

MIDPERCENTS

requests a table listing the midpoints and percent of observations in each histogram interval. For example, the following statements create the table in Figure 6.13:

```
proc capability;
   histogram Length / midpercents;
run;
```

Figure 6.13 Table of Midpoints and Observed Percentages

The CAPABILITY Procedure

Histogram Bins for Length	
Bin Midpoint	Observed Percent
10.02	12.000
10.08	32.000
10.14	28.000
10.20	18.000
10.26	6.000
10.32	4.000

If you specify the MIDPERCENTS option in parentheses after a density estimate option, a table listing the midpoints, observed percent of observations, and the estimated percent of the population in each interval (estimated from the fitted distribution) is printed.

The following statements create the table shown in Figure 6.14:

```
proc capability;
   histogram Length / gamma(theta=3 midpercents);
run;
```

Figure 6.14 Table of Observed and Expected Percentages

The CAPABILITY Procedure
Fitted Gamma Distribution for Length (Attachment Point Offset in mm)

Histogram Bin Percents for Gamma Distribution		
	Percent	
Bin Midpoint	Observed	Estimated
10.02	12.000	11.480
10.08	32.000	26.182
10.14	28.000	31.354
10.20	18.000	19.916
10.26	6.000	6.766
10.32	4.000	1.238

MIDPOINTS=*value-list* | **KEY** | **UNIFORM**

specifies how to determine the midpoints for the histogram intervals, where *values-list* determines the width of the histogram bars as the difference between consecutive midpoints. The procedure uses the same values for all variables. See Output 6.9.1.

The range of midpoints, extended at each end by half of the bar width, must cover the range of the data as well as any specification limits. For example, if you specify

```
midpoints=2 to 10 by 0.5
```

then all of the observations and specification limits should fall between 1.75 and 10.25. (Otherwise, a default list of midpoints is used.) You must use evenly spaced midpoints listed in increasing order.

KEY determines the midpoints for the data in the key cell. The initial number of midpoints is based on the number of observations in the key cell that use the method of Terrell and Scott (1985). The procedure extends the midpoint list for the key cell in either direction as necessary until it spans the data in the remaining cells.

UNIFORM determines the midpoints by using all the observations as if there were no cells. In other words, the number of midpoints is based on the total sample size by using the method of Terrell and Scott (1985).

Neither KEY nor UNIFORM apply unless you use the CLASS statement. By default, if you use a CLASS statement, MIDPOINTS=KEY. However, if the key cell is empty then MIDPOINTS=UNIFORM. Otherwise, the procedure computes the midpoints by using the algorithm described in Terrell and Scott (1985). The default midpoints are primarily applicable to continuous data that are approximately normally distributed.

If you produce traditional graphics and use the MIDPOINTS= and HAXIS= options, you can use the ORDER= option in the AXIS statement you specified with the HAXIS= option. However, for the tick mark labels to coincide with the histogram interval midpoints, the range of the ORDER= list must encompass the range of the MIDPOINTS= list, as illustrated in the following statements:

```
proc capability;
   histogram length / midpoints=20 to 80 by 10
                      haxis=axis1;
   axis1 length=6 in order=10 20 30 40 50 60 70 80 90;
run;
```

MIDPTAXIS=*name*

is an alias for the HAXIS= option.

MU=*value-list*

specifies the parameter μ for fitted curves requested with the GUMBEL, IGAUSS, and NORMAL options. Enclose the MU= option in parentheses after the distribution keyword. For the normal and inverse Gaussian distributions, the default value of μ is the sample mean. If you do not specify a value for μ for the Gumbel distribution, the procedure calculates a maximum likelihood estimate.

NENDPOINTS=*n*

specifies the number of histogram interval endpoints and causes the endpoints, rather than interval midpoints, to be aligned with horizontal axis tick marks.

NMIDPOINTS=*n*

specifies the number of histogram intervals.

NOBARS
suppresses drawing of histogram bars. This option is useful when you want to display fitted curves only.

NOCURVELEGEND
NOCURVEL
suppresses the portion of the legend for fitted curves. If you use the INSET statement to display information about the fitted curve on the histogram, you can use the NOCURVELEGEND option to prevent the information about the fitted curve from being repeated in a legend at the bottom of the histogram. See Output 6.15.1.

NOLEGEND
suppresses legends for specification limits, fitted curves, and hidden observations. See Example 6.13. Specifying the NOLEGEND option is equivalent to specifying LEGEND=NONE.

NOPLOT
suppresses the creation of a plot. Use the NOPLOT option when you want only to print summary statistics for a fitted density or create either an OUTFIT= or an OUTHISTOGRAM= data set. See Example 6.11.

NOPRINT
suppresses printed output summarizing the fitted curve. Enclose the NOPRINT option in parentheses following the distribution option. See "Customizing a Histogram" on page 304 for an example.

NORMAL< (*normal-options*) >
displays a fitted normal density curve on the histogram. The curve equation is

$$p(x) = \frac{hv}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

where

μ = mean
σ = standard deviation ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

Note that the normal distribution is also referred to as the S_N distribution in the Johnson system of distributions.

You can specify values for μ and σ with the MU= and SIGMA= *normal-options*, as shown in the following statements:

```
proc capability;
   histogram length / normal(mu=14 sigma=0.05);
run;
```

By default, the sample mean and sample standard deviation are used for μ and σ. The NORMAL option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options that you can specify with the NORMAL option. See Figure 6.10 and "Formulas for Fitted Curves" on page 336.

NOSPECLEGEND

NOSPECL

suppresses the portion of the legend for specification limit reference lines. See Figure 6.11.

NOTABCONTENTS

suppresses the table of contents entries for tables produced by the HISTOGRAM statement. See the section "ODS Tables" on page 359 for descriptions of the tables produced by the HISTOGRAM statement.

OPTBOUNDRANGE=*value*

defines the sampling range for each parameter during maximum likelihood estimation for the Johnson S_U distribution. PROC UNIVARIATE computes initial estimates for each parameter by using the method of percentiles. The *value* determines the range of parameter values around the initial estimate that can be sampled for local optimization starting values. The default is 100.

OPTMAXITER=*value*

limits the number of iterations that are used by the optimizer in maximum likelihood estimation for the Johnson S_U distribution. The default is 500.

OPTMAXSTARTS=*N*

defines the maximum number of starting points to be used for local optimization in maximum likelihood estimation for the Johnson S_U distribution. That is, no more than *N* local optimizations are used in the multistart algorithm. The default value is 100.

OPTPRINT

prints the iteration history for the Johnson S_U distribution maximum likelihood estimation.

OPTSEED=*value*

specifies a positive integer seed for generating random number sequences in Johnson S_U distribution maximum likelihood estimation. You can use this option to replicate results from different runs.

OPTTOLERANCE=*value*

specifies the tolerance for declaring optimality in maximum likelihood estimation for the Johnson S_U distribution. The default value is 1E–8.

OUTFIT=*SAS-data-set*

creates a SAS data set that contains parameter estimates for fitted curves and related goodness-of-fit information. See "Output Data Sets" on page 356.

OUTHISTOGRAM=*SAS-data-set*

OUTHIST=*SAS-data-set*

creates a SAS data set that contains information about histogram intervals. Specifically, the data set contains the midpoints of the histogram intervals, the observed percent of observations in each interval, and the estimated percent of observations in each interval (estimated from each of the specified fitted curves). See "Output Data Sets" on page 356.

OUTKERNEL=*SAS-data-set*

creates a SAS data set containing information about kernel density estimates requested with the KERNEL option. See "OUTKERNEL= Output Data Set" on page 359 for details.

PARETO< (*Pareto-options***) >**

displays a fitted generalized Pareto density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{hv}{\sigma}(1-\alpha(x-\theta)/\sigma)^{1/\alpha-1} & \text{if } \alpha \neq 0 \\ \frac{hv}{\sigma}\exp(-(x-\theta)/\sigma) & \text{if } \alpha = 0 \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Pareto-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, the generalized Pareto distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *Pareto-options*. By default, maximum likelihood estimates are computed for α and σ.

The PARETO option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the PARETO option. See "Formulas for Fitted Curves" on page 336.

PCTAXIS=*name*|*value-list*

is an alias for the VAXIS= option.

PERCENTS=*value-list*

PERCENT=*value-list*

specifies a list of percents for which quantiles calculated from the data and quantiles estimated from the fitted curve are tabulated. The percents must be between 0 and 100. Enclose the PERCENTS= option in parentheses after the curve option. The default percents are 1, 5, 10, 25, 50, 75, 90, 95, and 99.

For example, the following statements create the table shown in Figure 6.15:

```
proc capability;
   histogram Length / lognormal(percents=1 3 5 95 97 99);
run;
```

Figure 6.15 Estimated and Observed Quantiles for the Lognormal Curve

The CAPABILITY Procedure
Fitted Lognormal Distribution for Length (Attachment Point Offset in mm)

Quantiles for Lognormal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	10.0180	9.95696
3.0	10.0180	9.98937
5.0	10.0310	10.00658
95.0	10.2780	10.24963
97.0	10.2930	10.26729
99.0	10.3220	10.30071

POWER< (*power-options*) >
displays a fitted power function density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} hv\frac{\alpha}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1} & \text{for } \theta < x < \theta + \sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta + \sigma \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The parameter θ must be less than or equal to the minimum data value. You can specify θ and σ with the THETA= and the SIGMA= *power-options*. The default values for θ and σ are 0 and 1, respectively. You can specify THETA=EST and SIGMA=EST to request maximum likelihood estimates for θ and σ.

In addition, the generalized Pareto distribution has a shape parameter α. You can specify α with the ALPHA= *power-option*. By default, a maximum likelihood estimate is computed for α.

The POWER option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the POWER option. See "Formulas for Fitted Curves" on page 336.

RAYLEIGH< (*Rayleigh-options*) >
displays a fitted Rayleigh density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} hv\frac{x-\theta}{\sigma^2}e^{-(x-\theta)^2/(2\sigma^2)} & \text{for } x \ge \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The parameter θ must be less than or equal to the minimum data value. You can specify θ with the THETA= *Rayleigh-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ with the SIGMA= *Rayleigh-option*. By default, a maximum likelihood estimate is computed for σ.

The RAYLEIGH option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the RAYLEIGH option. See "Formulas for Fitted Curves" on page 336.

RTINCLUDE
includes the right endpoint of each histogram interval in that interval. By default, the left endpoint is included in the histogram interval.

SB<(*S_B-options*)>
displays a fitted Johnson S_B density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{\delta h v}{\sigma\sqrt{2\pi}} \left[\left(\frac{x-\theta}{\sigma} \right) \left(1 - \frac{x-\theta}{\sigma} \right) \right]^{-1} \times & \\ \exp\left[-\frac{1}{2} \left(\gamma + \delta \log\left(\frac{x-\theta}{\theta+\sigma-x} \right) \right)^2 \right] & \text{for } \theta < x < \theta + \sigma \\ 0 & \text{for } x \leq \theta \text{ or } x \geq \theta + \sigma \end{cases}$$

where

θ = threshold parameter ($-\infty < \theta < \infty$)
σ = scale parameter ($\sigma > 0$)
δ = shape parameter ($\delta > 0$)
γ = shape parameter ($-\infty < \gamma < \infty$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The S_B distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. The parameter θ must be less than the minimum data value. You can specify θ with the THETA= *S_B-option*, or you can request that θ be estimated with the THETA = EST *S_B-option*. The default value for θ is zero.

The sum $\theta + \sigma$ must be greater than the maximum data value. The default value for σ is one. You can specify σ with the SIGMA= S_B*-option*, or you can request that σ be estimated with the SIGMA = EST S_B*-option*. You can specify δ with the DELTA= S_B*-option*, and you can specify γ with the GAMMA= S_B*-option*. Note that the S_B*-options* are given in parentheses after the SB option.

By default, the method of percentiles is used to estimate the parameters of the S_B distribution. Alternatively, you can request the method of moments or the method of maximum likelihood with the FITMETHOD = MOMENTS or FITMETHOD = MLE options, respectively. Consider the following example:

```
proc capability;
   histogram length / sb;
   histogram length / sb( theta=est sigma=est );
   histogram length / sb( theta=0.5 sigma=8.4
                          delta=0.8 gamma=-0.6 );
run;
```

The first HISTOGRAM statement fits an S_B distribution with default values of $\theta = 0$ and $\sigma = 1$ and with percentile-based estimates for δ and γ. The second HISTOGRAM statement estimates all four parameters with the method of percentiles. The third HISTOGRAM statement displays an S_B curve with specified values for all four parameters.

The SB option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the SB option.

SIGMA=*value-list*

specifies the parameter σ for fitted curves requested with the BETA, EXPONENTIAL, GAMMA, GUMBEL, LOGNORMAL, NORMAL, PARETO, POWER, RAYLEIGH, SB, SU, and WEIBULL options. Enclose the SIGMA= option in parentheses after the distribution keyword. The following table summarizes the use of the SIGMA= option.

Distribution Keyword	SIGMA= Specifies	Default Value	Alias
BETA	scale parameter σ	1	SCALE=
EXPONENTIAL	scale parameter σ	maximum likelihood estimate	SCALE=
GAMMA	scale parameter σ	maximum likelihood estimate	SCALE=
GUMBEL	scale parameter σ	maximum likelihood estimate	
LOGNORMAL	shape parameter σ	maximum likelihood estimate	SHAPE=
NORMAL	scale parameter σ	standard deviation	
PARETO	scale parameter σ	maximum likelihood estimate	
POWER	scale parameter σ	1	SCALE=
RAYLEIGH	scale parameter σ	maximum likelihood estimate	
SB	scale parameter σ	1	SCALE=
SU	scale parameter σ	percentile-based estimate	SCALE=
WEIBULL	scale parameter σ	maximum likelihood estimate	SCALE=

If you specify SIGMA=EST, an estimate is computed for σ. For syntax examples, see the entries for the distribution options.

SU<(*S_U -options*)>

displays a fitted Johnson S_U density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{\delta h v}{\sigma\sqrt{2\pi}} \frac{1}{\sqrt{1+((x-\theta)/\sigma)^2}} \times \\ \exp\left[-\frac{1}{2}\left(\gamma + \delta \sinh^{-1}\left(\frac{x-\theta}{\sigma}\right)\right)^2\right] & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = location parameter ($-\infty < \theta < \infty$)
σ = scale parameter ($\sigma > 0$)
δ = shape parameter ($\delta > 0$)
γ = shape parameter ($-\infty < \gamma < \infty$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

You can specify the parameters with the THETA=, SIGMA=, DELTA=, and GAMMA= *S_U -options*, which are enclosed in parentheses after the SU option. If you do not specify these parameters, they are estimated.

By default, the method of percentiles is used to estimate the parameters of the S_U distribution. Alternatively, you can request the method of moments or the method of maximum likelihood with the FITMETHOD = MOMENTS or FITMETHOD = MLE options, respectively. Consider the following example:

```
proc capability;
   histogram length / su;
   histogram length / su( theta=0.5 sigma=8.4
                          delta=0.8 gamma=-0.6 );
run;
```

The first HISTOGRAM statement estimates all four parameters with the method of percentiles. The second HISTOGRAM statement displays an S_U curve with specified values for all four parameters.

The SU option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options you can specify with the SU option.

THETA=*value-list*

THRESHOLD=*value-list*

specifies the lower threshold parameter θ for curves requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, PARETO, POWER, RAYLEIGH, SB, and WEIBULL options, and the location parameter θ for curves requested with the SU option. Enclose the THETA= option in parentheses after the curve option. See Example 6.8. The default *value* is zero. If you specify THETA=EST, an estimate is computed for θ.

UPPER=*value-list*

specifies upper bounds for kernel density estimates requested with the KERNEL option. Enclose the UPPER= option in parentheses after the KERNEL option. You can specify up to five upper bounds for multiple kernel density estimates. If you specify more kernel estimates than upper bounds, the last upper bound is repeated for the remaining estimates.

VSCALE=COUNT | PERCENT | PROPORTION

specifies the scale of the vertical axis. The value COUNT scales the data in units of the number of observations per data unit. The value PERCENT scales the data in units of percent of observations per data unit. The value PROPORTION scales the data in units of proportion of observations per data unit. See Figure 6.11 for an illustration of VSCALE=COUNT. The default is PERCENT.

WEIBULL< (*Weibull-options***) >**

displays a fitted Weibull density curve on the histogram. The curve equation is

$$p(x) = \begin{cases} \frac{chv}{\sigma}(\frac{x-\theta}{\sigma})^{c-1}\exp(-(\frac{x-\theta}{\sigma})^c) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)
h = width of histogram interval
v = vertical scaling factor
and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Weibull-option*. The default value for θ is zero. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ and c with the SIGMA= and C= *Weibull-options*. By default, maximum likelihood estimates are computed for c and σ. For example, the following statements fit a Weibull distribution with $\theta = 15$ and with maximum likelihood estimates for σ and c:

```
proc capability;
   histogram length / weibull(theta=15);
run;
```

Note that the maximum likelihood estimate of c is calculated iteratively using the Newton-Raphson approximation. The CDELTA=, CINITIAL=, and MAXITER= *Weibull-options* control the approximation.

The WEIBULL option can appear only once in a HISTOGRAM statement. Table 6.19 lists secondary options that you can specify with the WEIBULL option. See Example 6.9 and "Formulas for Fitted Curves" on page 336.

ZETA=*value-list*
specifies a value for the scale parameter ζ for lognormal density curves requested with the LOGNORMAL option. Enclose the ZETA= option in parentheses after the LOGNORMAL option. By default, the procedure calculates a maximum likelihood estimate for ζ. You can specify the SCALE= option as an alias for the ZETA= option.

Options for Traditional Graphics

BARWIDTH=*value*
specifies the width of the histogram bars in screen percent units.

BMCFILL=*color*
specifies the fill color for a box-and-whisker plot in a bottom margin requested with the BMPLOT= option. By default, the box-and-whisker plot is not filled.

BMCFRAME=*color*
specifies the color for filling the frame of a bottom margin plot requested with the BMPLOT= option. By default, this area is not filled.

BMCOLOR=*color*
specifies the color of a carpet plot, or the outline color of a box-and-whisker plot, in a bottom margin plot requested with the BMPLOT= option.

BMMARGIN=*height*
specifies the height in screen percentage units of a bottom margin plot requested with the BMPLOT= option. By default, a bottom margin plot occupies 15 percent of the vertical display space.

CBARLINE=*color*
specifies the color of the outline of histogram bars. This option overrides the C= option in the SYMBOL1 statement.

CFILL=*color*
specifies a color used to fill the bars of the histogram (or the area under a fitted curve if you also specify the FILL option). See the entries for the FILL and PFILL= options for additional details. See Figure 6.11 and Output 6.8.1. Refer to *SAS/GRAPH: Reference* for a list of colors. By default, bars are filled with an appropriate color from the ODS style.

CGRID=*color*
specifies the color for grid lines requested with the GRID option. By default, grid lines are the same color as the axes. If you use CGRID=, you do not need to specify the GRID option.

CLIPREF
draws reference lines requested with the HREF= and VREF= options behind the histogram bars. By default, reference lines are drawn in front of the histogram bars.

CLIPSPEC=CLIP | NOFILL
specifies that histogram bars are clipped at the upper and lower specification limit lines when there are no observations outside the specification limits. The bar intersecting the lower specification limit is clipped if there are no observations less than the lower limit; the bar intersecting the upper specification limit is clipped if there are no observations greater than the upper limit. If you specify CLIPSPEC=CLIP, the histogram bar is truncated at the specification limit. If you specify CLIPSPEC=NOFILL, the portion of a filled histogram bar outside the specification limit is left unfilled. Specifying CLIPSPEC=NOFILL when histogram bars are not filled has no effect.

CURVELEGEND=*name* | **NONE**
specifies the name of a LEGEND statement describing the legend for specification limits and fitted curves. Specifying CURVELEGEND=NONE suppresses the legend for fitted curves; this is equivalent to specifying the NOCURVELEGEND option.

FRONTREF
draws reference lines requested with the HREF= and VREF= options in front of the histogram bars. When the NOGSTYLE system option is specified, reference lines are drawn behind the histogram bars by default, and can be obscured by them.

HOFFSET=*value*
specifies the offset in percent screen units at both ends of the horizontal axis. Specify HOFFSET=0 to eliminate the default offset.

INTERBAR=*value*
specifies the horizontal space in percent screen units between histogram bars. By default, the bars are contiguous.

LEGEND=*name* | **NONE**
specifies the name of a LEGEND statement describing the legend for specification limit reference lines and fitted curves. Specifying LEGEND=NONE suppresses all legend information and is equivalent to specifying the NOLEGEND option.

LGRID=*n*
specifies the line type for the grid requested with the GRID option. If you use the LGRID= option, you do not need to specify the GRID option. The default is 1, which produces a solid line.

PFILL=*pattern*
specifies a pattern used to fill the bars of the histograms (or the areas under a fitted curve if you also specify the FILL option). See the entries for the CFILL= and FILL options for additional details. Refer to *SAS/GRAPH: Reference* for a list of pattern values. By default, the bars and curve areas are not filled.

SPECLEGEND=*name* | **NONE**
specifies the name of a LEGEND statement describing the legend for specification limits and fitted curves. Specifying SPECLEGEND=NONE, which suppresses the portion of the legend for specification limit references lines, is equivalent to specifying the NOSPECLEGEND option.

VOFFSET=*value*
specifies the offset in percent screen units at the upper end of the vertical axis.

WBARLINE=*n*
specifies the width of bar outlines. By default, $n = 1$.

WGRID=*n*
specifies the width of the grid lines requested with the GRID option. By default, grid lines are the same width as the axes. If you use the WGRID= option, you do not need to specify the GRID option.

Options for Legacy Line Printer Charts

SYMBOL='*character*'

specifies the *character* used for the density curve or kernel density curve in line printer plots. Enclose the SYMBOL= option in parentheses after the distribution option or the KERNEL option. The default character is the first letter of the distribution keyword or '1' for the first kernel density estimate, '2' for the second kernel density estimate, and so on. If you use the SYMBOL= option with the KERNEL option, you can specify a list of up to five characters in parentheses for multiple kernel density estimates. If there are more estimates than characters, the last character specified is used for the remaining estimates.

Details: HISTOGRAM Statement

This section provides details on the following topics:

- formulas for fitted distributions
- formulas for kernel density estimates
- printed output
- OUTFIT=, OUTHISTOGRAM=, and OUTKERNEL= data sets
- graphical enhancements to histograms

Formulas for Fitted Curves

The following sections provide information about the families of parametric distributions that you can fit with the HISTOGRAM statement. Properties of these distributions are discussed by Johnson, Kotz, and Balakrishnan (1994) and Johnson, Kotz, and Balakrishnan (1995).

Beta Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\sigma+\theta-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} hv & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and

θ = lower threshold parameter (lower endpoint parameter)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
β = shape parameter ($\beta > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

NOTE: This notation is consistent with that of other distributions that you can fit with the HISTOGRAM statement. However, many texts, including Johnson, Kotz, and Balakrishnan (1995), write the beta density function as

$$p(x) = \begin{cases} \frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-a)^{p+q-1}} & \text{for } a < x < b \\ 0 & \text{for } x \leq a \text{ or } x \geq b \end{cases}$$

The two notations are related as follows:

$\sigma = b - a$
$\theta = a$
$\alpha = p$
$\beta = q$

The range of the beta distribution is bounded below by a threshold parameter $\theta = a$ and above by $\theta + \sigma = b$. If you specify a fitted beta curve by using the BETA option, θ must be less than the minimum data value, and $\theta + \sigma$ must be greater than the maximum data value. You can specify θ and σ with the THETA= and SIGMA= *beta-options* in parentheses after the keyword BETA. By default, $\sigma = 1$ and $\theta = 0$. If you specify THETA=EST and SIGMA=EST, maximum likelihood estimates are computed for θ and σ.

In addition, you can specify α and β with the ALPHA= and BETA= *beta-options*, respectively. By default, the procedure calculates maximum likelihood estimates for α and β. For example, to fit a beta density curve to a set of data bounded below by 32 and above by 212 with maximum likelihood estimates for α and β, use the following statement:

```
histogram length / beta(theta=32 sigma=180);
```

The beta distributions are also referred to as Pearson Type I or II distributions. These include the *power-function* distribution ($\beta = 1$), the *arc-sine* distribution ($\alpha = \beta = \frac{1}{2}$), and the *generalized arc-sine* distributions ($\alpha + \beta = 1, \beta \neq \frac{1}{2}$).

You can use the DATA step function BETAINV to compute beta quantiles and the DATA step function PROBBETA to compute beta probabilities.

Exponential Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{hv}{\sigma} \exp(-(\frac{x-\theta}{\sigma})) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The threshold parameter θ must be less than or equal to the minimum data value. You can specify θ with the THRESHOLD= *exponential-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, you can specify σ with the SCALE= *exponential-option*. By default, the procedure calculates a maximum likelihood estimate for σ. Note that some authors define the scale parameter as $\frac{1}{\sigma}$.

The exponential distribution is a special case of both the gamma distribution (with $\alpha = 1$) and the Weibull distribution (with $c = 1$). A related distribution is the *extreme value* distribution. If $Y = \exp(-X)$ has an exponential distribution, then X has an extreme value distribution.

Gamma Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{hv}{\Gamma(\alpha)\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THRESHOLD= *gamma-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. In addition, you can specify σ and α with the SCALE= and ALPHA= *gamma-options*. By default, the procedure calculates maximum likelihood estimates for σ and α.

The gamma distributions are also referred to as Pearson Type III distributions, and they include the chi-square, exponential, and Erlang distributions. The probability density function for the chi-square distribution is

$$p(x) = \begin{cases} \frac{1}{2\Gamma\left(\frac{\nu}{2}\right)}\left(\frac{x}{2}\right)^{\frac{\nu}{2}-1}\exp\left(-\frac{x}{2}\right) & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

Notice that this is a gamma distribution with $\alpha = \frac{\nu}{2}$, $\sigma = 2$, and $\theta = 0$. The exponential distribution is a gamma distribution with $\alpha = 1$, and the Erlang distribution is a gamma distribution with α being a positive integer. A related distribution is the Rayleigh distribution. If $R = \frac{\max(X_1,\ldots,X_n)}{\min(X_1,\ldots,X_n)}$ where the X_i's

are independent χ^2_ν variables, then $\log R$ is distributed with a χ_ν distribution having a probability density function of

$$p(x) = \begin{cases} \left[2^{\frac{\nu}{2}-1}\Gamma(\frac{\nu}{2})\right]^{-1} x^{\nu-1}\exp(-\frac{x^2}{2}) & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

If $\nu = 2$, the preceding distribution is referred to as the Rayleigh distribution.

You can use the DATA step function GAMINV to compute gamma quantiles and the DATA step function PROBGAM to compute gamma probabilities.

Gumbel Distribution

The fitted density function is

$$p(x) = \frac{hv}{\sigma} e^{-(x-\mu)/\sigma} \exp\left(-e^{-(x-\mu)/\sigma}\right)$$

where

μ = location parameter
σ = scale parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

You can specify μ and σ with the MU= and SIGMA= *Gumbel-options*, respectively. By default, the procedure calculates maximum likelihood estimates for these parameters.

NOTE: The Gumbel distribution is also referred to as Type 1 extreme value distribution.

NOTE: The random variable X has Gumbel (Type 1 extreme value) distribution if and only if e^X has Weibull distribution and $\exp((X-\mu)/\sigma)$ has standard exponential distribution.

Inverse Gaussian Distribution

The fitted density function is

$$p(x) = \begin{cases} hv\left(\frac{\lambda}{2\pi x^3}\right)^{1/2} \exp\left(-\frac{\lambda}{2\mu^2 x}(x-\mu)^2\right) & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

where

μ = location parameter ($\mu > 0$)
λ = shape parameter ($\lambda > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The location parameter μ has to be greater then zero. You can specify μ with the MU= *iGauss-option*. In addition, you can specify shape parameter λ with the LAMBDA= *iGauss-option*. By default, the procedure uses the sample mean for μ and calculates a maximum likelihood estimate for λ.

NOTE: The special case where $\mu = 1$ and $\lambda = \phi$ corresponds to the Wald distribution.

Lognormal Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{hv}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
ζ = scale parameter $(-\infty < \zeta < \infty)$
σ = shape parameter $(\sigma > 0)$
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THRESHOLD= *lognormal-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify ζ and σ with the SCALE= and SHAPE= *lognormal-options*, respectively. By default, the procedure calculates maximum likelihood estimates for these parameters.

NOTE: The lognormal distribution is also referred to as the S_L distribution in the Johnson system of distributions.

NOTE: This book uses σ to denote the shape parameter of the lognormal distribution, whereas σ is used to denote the scale parameter of the beta, exponential, gamma, Gumbel, inverse Gaussian, normal, generalized Pareto, power function, Rayleigh, and Weibull distributions. The use of σ to denote the lognormal shape parameter is based on the fact that $\frac{1}{\sigma}(\log(X-\theta)-\zeta)$ has a standard normal distribution if X is lognormally distributed.

Normal Distribution

The fitted density function is

$$p(x) = \frac{hv}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad \text{for } -\infty < x < \infty$$

where
μ = mean
σ = standard deviation ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

You can specify μ and σ with the MU= and SIGMA= *normal-options*, respectively. By default, the procedure estimates μ with the sample mean and σ with the sample standard deviation.

You can use the DATA step function PROBIT to compute normal quantiles and the DATA step function PROBNORM to compute probabilities.

NOTE: The normal distribution is also referred to as the S_N distribution in the Johnson system of distributions.

Generalized Pareto Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{hv}{\sigma}(1 - \alpha(x-\theta)/\sigma)^{1/\alpha - 1} & \text{if } \alpha \neq 0 \\ \frac{hv}{\sigma}\exp(-(x-\theta)/\sigma) & \text{if } \alpha = 0 \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The support of the distribution is $x > \theta$ for $\alpha \leq 0$ and $\theta < x < \sigma/\alpha$ for $\alpha > 0$.

NOTE: Special cases of the generalized Pareto distribution with $\alpha = 0$ and $\alpha = 1$ correspond respectively to the exponential distribution with mean σ and uniform distribution on the interval (θ, σ).

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Pareto-option*. By default, $\theta = 0$. You can also specify α and σ with the ALPHA= and SIGMA= *Pareto-options*,respectively. By default, the procedure calculates maximum likelihood estimates for these parameters.

NOTE: Maximum likelihood estimation of the parameters works well if $\alpha < \frac{1}{2}$, but not otherwise. In this case the estimators are asymptotically normal and asymptotically efficient. The asymptotic normal

distribution of the maximum likelihood estimates has mean (α, σ) and variance-covariance matrix

$$\frac{1}{n}\begin{pmatrix} (1-\alpha)^2 & \sigma(1-\alpha) \\ \sigma(1-\alpha) & 2\sigma^2(1-\alpha) \end{pmatrix}.$$

NOTE: If no local minimum is found in the region

$$\{\alpha < 0, \sigma > 0\} \cup \{0 < \alpha \leq 1, \sigma/\alpha > \max(X_i)\},$$

there is no maximum likelihood estimator. More details on how to find maximum likelihood estimators and suggested algorithm can be found in Grimshaw(1993).

Power Function Distribution

The fitted density function is

$$p(x) = \begin{cases} h v \frac{\alpha}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1} & \text{for } \theta < x < \theta + \sigma \\ 0 & \text{for } x \leq \theta \text{ or } x \geq \theta + \sigma \end{cases}$$

where

θ = lower threshold parameter (lower endpoint parameter)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

NOTE: This notation is consistent with that of other distributions that you can fit with the HISTOGRAM statement. However, many texts, including Johnson, Kotz, and Balakrishnan (1995), write the density function of power function distribution as

$$p(x) = \begin{cases} \frac{p}{b-a}\left(\frac{x-a}{b-a}\right)^{p-1} & \text{for } a < x < b \\ 0 & \text{for } x \leq a \text{ or } x \geq b \end{cases}$$

The two parameterizations are related as follows:

$$\sigma = b - a$$

$$\theta = a$$

$$\alpha = p$$

NOTE: The family of power function distributions is a subclass of beta distribution with density function

$$p(x) = \begin{cases} hv\frac{(x-\theta)^{\alpha-1}(\sigma+\theta-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ with parameter $\beta = 1$. Therefore, all properties and estimation procedures of beta distribution apply.

The range of the power function distribution is bounded below by a threshold parameter $\theta = a$ and above by $\theta + \sigma = b$. If you specify a fitted power function curve by using the POWER option, θ must be less than the minimum data value and $\theta + \sigma$ must be greater than the maximum data value. You can specify θ and σ with the THETA= and SIGMA= *power-options* in parentheses after the keyword POWER. By default, $\sigma = 1$ and $\theta = 0$. If you specify THETA=EST and SIGMA=EST, maximum likelihood estimates are computed for θ and σ. However, three-parameter maximum likelihood estimation does not always converge.

In addition, you can specify α with the ALPHA= *power-option*. By default, the procedure calculates a maximum likelihood estimate for α. For example, to fit a power function density curve to a set of data bounded below by 32 and above by 212 with maximum likelihood estimate for α, use the following statement:

```
histogram Length / power(theta=32 sigma=180);
```

Rayleigh Distribution

The fitted density function is

$$p(x) = \begin{cases} hv\frac{x-\theta}{\sigma^2}e^{-(x-\theta)^2/(2\sigma^2)} & \text{for } x \ge \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where

θ = lower threshold parameter (lower endpoint parameter)
σ = scale parameter ($\sigma > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

NOTE: The Rayleigh distribution is a Weibull distribution with density function

$$p(x) = \begin{cases} hv\frac{k}{\lambda}\left(\frac{x-\theta}{\lambda}\right)^{k-1}\exp(-(\frac{x-\theta}{\lambda})^k) & \text{for } x \ge \theta \\ 0 & \text{for } x < \theta \end{cases}$$

and with shape parameter $k = 2$ and scale parameter $\lambda = \sqrt{2}\sigma$.

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THETA= *Rayleigh-option*. By default, $\theta = 0$. In addition you can specify σ with the SIGMA= *Rayleigh-option*. By default, the procedure calculates maximum likelihood estimate for σ.

For example, to fit a Rayleigh density curve to a set of data bounded below by 32 with maximum likelihood estimate for σ, use the following statement:

```
histogram Length / rayleigh(theta=32);
```

Johnson S_B Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{\delta h v}{\sigma\sqrt{2\pi}} \left[\left(\frac{x-\theta}{\sigma} \right) \left(1 - \frac{x-\theta}{\sigma} \right) \right]^{-1} \times & \\ \exp\left[-\frac{1}{2} \left(\gamma + \delta \log\left(\frac{x-\theta}{\theta+\sigma-x} \right) \right)^2 \right] & \text{for } \theta < x < \theta + \sigma \\ 0 & \text{for } x \leq \theta \text{ or } x \geq \theta + \sigma \end{cases}$$

where

θ = threshold parameter ($-\infty < \theta < \infty$)
σ = scale parameter ($\sigma > 0$)
δ = shape parameter ($\delta > 0$)
γ = shape parameter ($-\infty < \gamma < \infty$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The S_B distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. The parameter θ must be less than the minimum data value. You can specify θ with the THETA= S_B-*option*, or you can request that θ be estimated with the THETA = EST S_B-*option*. The default value for θ is zero. The sum $\theta + \sigma$ must be greater than the maximum data value. The default value for σ is one. You can specify σ with the SIGMA= S_B-*option*, or you can request that σ be estimated with the SIGMA = EST S_B-*option*.

By default, the method of percentiles given by Slifker and Shapiro (1980) is used to estimate the parameters. This method is based on four data percentiles, denoted by x_{-3z}, x_{-z}, x_z, and x_{3z}, which correspond to the four equally spaced percentiles of a standard normal distribution, denoted by $-3z$, $-z$, z, and $3z$, under the transformation

$$z = \gamma + \delta \log\left(\frac{x-\theta}{\theta+\sigma-x} \right)$$

The default value of z is 0.524. The results of the fit are dependent on the choice of z, and you can specify other values with the FITINTERVAL= option (specified in parentheses after the SB option). If you use the method of percentiles, you should select a value of z that corresponds to percentiles which are critical to your application.

The following values are computed from the data percentiles:

$$\begin{aligned} m &= x_{3z} - x_z \\ n &= x_{-z} - x_{-3z} \\ p &= x_z - x_{-z} \end{aligned}$$

It was demonstrated by Slifker and Shapiro (1980) that

$$\begin{array}{ll} \frac{mn}{p^2} > 1 & \text{for any } S_U \text{ distribution} \\ \frac{mn}{p^2} < 1 & \text{for any } S_B \text{ distribution} \\ \frac{mn}{p^2} = 1 & \text{for any } S_L \text{ (lognormal) distribution} \end{array}$$

A tolerance interval around one is used to discriminate among the three families with this ratio criterion. You can specify the tolerance with the FITTOLERANCE= option (specified in parentheses after the SB option). The default tolerance is 0.01. Assuming that the criterion satisfies the inequality

$$\frac{mn}{p^2} < 1 - \text{tolerance}$$

the parameters of the S_B distribution are computed using the explicit formulas derived by Slifker and Shapiro (1980).

If you specify FITMETHOD = MOMENTS (in parentheses after the SB option) the method of moments is used to estimate the parameters. If you specify FITMETHOD = MLE (in parentheses after the SB option) the method of maximum likelihood is used to estimate the parameters. Note that maximum likelihood estimates may not always exist. Refer to Bowman and Shenton (1983) for discussion of methods for fitting Johnson distributions.

Johnson S_U Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{\delta h v}{\sigma\sqrt{2\pi}} \frac{1}{\sqrt{1+((x-\theta)/\sigma)^2}} \times \\ \exp\left[-\frac{1}{2}\left(\gamma + \delta \sinh^{-1}\left(\frac{x-\theta}{\sigma}\right)\right)^2\right] & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = location parameter ($-\infty < \theta < \infty$)
σ = scale parameter ($\sigma > 0$)
δ = shape parameter ($\delta > 0$)
γ = shape parameter ($-\infty < \gamma < \infty$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

You can specify the parameters with the THETA=, SIGMA=, DELTA=, and GAMMA= S_U*-options*, which are enclosed in parentheses after the SU option. If you do not specify these parameters, they are estimated.

By default, the method of percentiles given by Slifker and Shapiro (1980) is used to estimate the parameters. This method is based on four data percentiles, denoted by x_{-3z}, x_{-z}, x_z, and x_{3z}, which correspond to the

four equally spaced percentiles of a standard normal distribution, denoted by $-3z$, $-z$, z, and $3z$, under the transformation

$$z = \gamma + \delta \sinh^{-1}\left(\frac{x-\theta}{\sigma}\right)$$

The default value of z is 0.524. The results of the fit are dependent on the choice of z, and you can specify other values with the FITINTERVAL= option (specified in parentheses after the SU option). If you use the method of percentiles, you should select a value of z that corresponds to percentiles which are critical to your application.

The following values are computed from the data percentiles:

$$\begin{aligned} m &= x_{3z} - x_z \\ n &= x_{-z} - x_{-3z} \\ p &= x_z - x_{-z} \end{aligned}$$

It was demonstrated by Slifker and Shapiro (1980) that

$$\begin{aligned} \frac{mn}{p^2} &> 1 \quad \text{for any } S_U \text{ distribution} \\ \frac{mn}{p^2} &< 1 \quad \text{for any } S_B \text{ distribution} \\ \frac{mn}{p^2} &= 1 \quad \text{for any } S_L \text{ (lognormal) distribution} \end{aligned}$$

A tolerance interval around one is used to discriminate among the three families with this ratio criterion. You can specify the tolerance with the FITTOLERANCE= option (specified in parentheses after the SU option). The default tolerance is 0.01. Assuming that the criterion satisfies the inequality

$$\frac{mn}{p^2} > 1 + \text{tolerance}$$

the parameters of the S_U distribution are computed using the explicit formulas derived by Slifker and Shapiro (1980).

If you specify FITMETHOD = MOMENTS (in parentheses after the SU option) the method of moments is used to estimate the parameters. If you specify FITMETHOD = MLE (in parentheses after the SU option) the method of maximum likelihood is used to estimate the parameters. Note that maximum likelihood estimates may not always exist. Refer to Bowman and Shenton (1983) for discussion of methods for fitting Johnson distributions.

Weibull Distribution

The fitted density function is

$$p(x) = \begin{cases} \frac{chv}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where
θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)
h = width of histogram interval
v = vertical scaling factor, and

$$v = \begin{cases} n & \text{the sample size, for VSCALE=COUNT} \\ 100 & \text{for VSCALE=PERCENT} \\ 1 & \text{for VSCALE=PROPORTION} \end{cases}$$

The threshold parameter θ must be less than the minimum data value. You can specify θ with the THRESHOLD= *Weibull-option*. By default, $\theta = 0$. If you specify THETA=EST, a maximum likelihood estimate is computed for θ. You can specify σ and c with the SCALE= and SHAPE= *Weibull-options*, respectively. By default, the procedure calculates maximum likelihood estimates for σ and c.

The exponential distribution is a special case of the Weibull distribution where $c = 1$.

Kernel Density Estimates

You can use the KERNEL option to superimpose kernel density estimates on histograms. Smoothing the data distribution with a kernel density estimate can be more effective than using a histogram to examine features that might be obscured by the choice of histogram bins or sampling variation. A kernel density estimate can also be more effective than a parametric curve fit when the process distribution is multimodal. See Example 6.12.

The general form of the kernel density estimator is

$$\hat{f}_\lambda(x) = \frac{1}{n\lambda} \sum_{i=1}^{n} K_0\left(\frac{x - x_i}{\lambda}\right)$$

where $K_0(\cdot)$ is a kernel function, λ is the bandwidth, n is the sample size, and x_i is the ith observation.

The KERNEL option provides three kernel functions (K_0): normal, quadratic, and triangular. You can specify the function with the K= *kernel-option* in parentheses after the KERNEL option. Values for the K= option are NORMAL, QUADRATIC, and TRIANGULAR (with aliases of N, Q, and T, respectively). By default, a normal kernel is used. The formulas for the kernel functions are

Normal	$K_0(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2)$	for $-\infty < t < \infty$
Quadratic	$K_0(t) = \frac{3}{4}(1 - t^2)$	for $\lvert t \rvert \le 1$
Triangular	$K_0(t) = 1 - \lvert t \rvert$	for $\lvert t \rvert \le 1$

The value of λ, referred to as the bandwidth parameter, determines the degree of smoothness in the estimated density function. You specify λ indirectly by specifying a standardized bandwidth c with the C= *kernel-option*. If Q is the interquartile range, and n is the sample size, then c is related to λ by the formula

$$\lambda = cQn^{-\frac{1}{5}}$$

For a specific kernel function, the discrepancy between the density estimator $\hat{f}_\lambda(x)$ and the true density $f(x)$ is measured by the mean integrated square error (MISE):

$$\text{MISE}(\lambda) = \int_x \{E(\hat{f}_\lambda(x)) - f(x)\}^2 dx + \int_x var(\hat{f}_\lambda(x)) dx$$

The MISE is the sum of the integrated squared bias and the variance. An approximate mean integrated square error (AMISE) is

$$\text{AMISE}(\lambda) = \frac{1}{4}\lambda^4 \left(\int_t t^2 K(t) dt \right)^2 \int_x \left(f''(x) \right)^2 dx + \frac{1}{n\lambda} \int_t K(t)^2 dt$$

A bandwidth that minimizes AMISE can be derived by treating $f(x)$ as the normal density having parameters μ and σ estimated by the sample mean and standard deviation. If you do not specify a bandwidth parameter or if you specify C=MISE, the bandwidth that minimizes AMISE is used. The value of AMISE can be used to compare different density estimates. For each estimate, the bandwidth parameter c, the kernel function type, and the value of AMISE are reported in the SAS log.

The general kernel density estimates assume that the domain of the density to estimate can take on all values on a real line. However, sometimes the domain of a density is an interval bounded on one or both sides. For example, if a variable Y is a measurement of only positive values, then the kernel density curve should be bounded so that it is zero for negative Y values.

The CAPABILITY procedure uses a reflection technique to create the bounded kernel density curve, as described in Silverman (1986, pp 30-31). It adds the reflections of kernel density that are outside the boundary to the bounded kernel estimates. The general form of the bounded kernel density estimator is computed by replacing $K_0\left(\frac{x-x_i}{\lambda}\right)$ in the original equation with

$$\left\{ K_0\left(\frac{x-x_i}{\lambda}\right) + K_0\left(\frac{(x-x_l)+(x_i-x_l)}{\lambda}\right) + K_0\left(\frac{(x_u-x)+(x_u-x_i)}{\lambda}\right) \right\}$$

where x_l is the lower bound and x_u is the upper bound.

Without a lower bound, $x_l = \infty$ and $K_0\left(\frac{(x-x_l)+(x_i-x_l)}{\lambda}\right)$ equals zero. Similarly, without an upper bound, $x_u = \infty$ and $K_0\left(\frac{(x_u-x)+(x_u-x_i)}{\lambda}\right)$ equals zero.

When C=MISE is used with a bounded kernel density, the CAPABILITY procedure uses a bandwidth that minimizes the AMISE for its corresponding unbounded kernel.

Printed Output

If you request a fitted parametric distribution, printed output summarizing the fit is produced in addition to the graphical display. Figure 6.16 shows the printed output for a fitted lognormal distribution requested by the following statements:

```
proc capability data=Hang;
   spec target=14 lsl=13.95 usl=14.05;
   hist / lognormal(indices midpercents);
run;
```

Figure 6.16 Sample Summary of Fitted Distribution

The CAPABILITY Procedure
Fitted Lognormal Distribution for Width (Width in cm)

Parameters for Lognormal Distribution		
Parameter	**Symbol**	**Estimate**
Threshold	Theta	0
Scale	Zeta	2.638966
Shape	Sigma	0.001497
Mean		13.99873
Std Dev		0.020952

Goodness-of-Fit Tests for Lognormal Distribution					
Test		**Statistic**	**DF**	**p Value**	
Kolmogorov-Smirnov	**D**	0.09148348		**Pr > D**	>0.150
Cramer-von Mises	**W-Sq**	0.05040427		**Pr > W-Sq**	>0.500
Anderson-Darling	**A-Sq**	0.33476355		**Pr > A-Sq**	>0.500
Chi-Square	**Chi-Sq**	2.87938822	3	**Pr > Chi-Sq**	0.411

Percent Outside Specifications for Lognormal Distribution			
Lower Limit		**Upper Limit**	
LSL	13.950000	**USL**	14.050000
Obs Pct < LSL	2.000000	**Obs Pct > USL**	0
Est Pct < LSL	0.992170	**Est Pct > USL**	0.728125

Capability Indices Based on Lognormal Distribution	
Cp	0.795463
CPL	0.776822
CPU	0.814021
Cpk	0.776822
Cpm	0.792237

Histogram Bin Percents for Lognormal Distribution		
	Percent	
Bin Midpoint	**Observed**	**Estimated**
13.95	4.000	2.963
13.97	18.000	15.354
13.99	26.000	33.872
14.01	38.000	32.055
14.03	10.000	13.050
14.05	4.000	2.281

Figure 6.16 *continued*

Quantiles for Lognormal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	13.9440	13.9501
5.0	13.9656	13.9643
10.0	13.9710	13.9719
25.0	13.9860	13.9846
50.0	14.0018	13.9987
75.0	14.0129	14.0129
90.0	14.0218	14.0256
95.0	14.0241	14.0332
99.0	14.0470	14.0475

The summary is organized into the following parts:

- Parameters
- Chi-Square Goodness-of-Fit Test
- EDF Goodness-of-Fit Tests
- Specifications
- Indices Using the Fitted Curve
- Histogram Intervals
- Quantiles

These parts are described in the sections that follow.

Parameters

This section lists the parameters for the fitted curve as well as the estimated mean and estimated standard deviation. See "Formulas for Fitted Curves" on page 336.

Chi-Square Goodness-of-Fit Test

The chi-square goodness-of-fit statistic for a fitted parametric distribution is computed as follows:

$$\chi^2 = \sum_{i=1}^{m} \frac{(O_i - E_i)^2}{E_i}$$

where

O_i = observed value in ith histogram interval
E_i = expected value in ith histogram interval

m = number of histogram intervals
p = number of estimated parameters

The degrees of freedom for the chi-square test is equal to $m - p - 1$. You can save the observed and expected interval values in the OUTFIT= data set discussed in "Output Data Sets" on page 356.

Note that empty intervals are not combined, and the range of intervals used to compute χ^2 begins with the first interval containing observations and ends with the final interval containing observations.

EDF Goodness-of-Fit Tests

When you fit a parametric distribution, the HISTOGRAM statement provides a series of goodness-of-fit tests based on the empirical distribution function (EDF). The EDF tests offer advantages over the chi-square goodness-of-fit test, including improved power and invariance with respect to the histogram midpoints. For a thorough discussion, refer to D'Agostino and Stephens (1986).

The empirical distribution function is defined for a set of n independent observations $X_1, \ldots, X_n$ with a common distribution function $F(x)$. Denote the observations ordered from smallest to largest as $X_{(1)}, \ldots, X_{(n)}$. The empirical distribution function, $F_n(x)$, is defined as

$$
\begin{aligned}
&F_n(x) = 0, \quad x < X_{(1)} \\
&F_n(x) = \tfrac{i}{n}, \quad X_{(i)} \le x < X_{(i+1)} \quad i = 1, \ldots, n-1 \\
&F_n(x) = 1, \quad X_{(n)} \le x
\end{aligned}
$$

Note that $F_n(x)$ is a step function that takes a step of height $\frac{1}{n}$ at each observation. This function estimates the distribution function $F(x)$. At any value x, $F_n(x)$ is the proportion of observations less than or equal to x, while $F(x)$ is the probability of an observation less than or equal to x. EDF statistics measure the discrepancy between $F_n(x)$ and $F(x)$.

The computational formulas for the EDF statistics make use of the probability integral transformation $U = F(X)$. If $F(X)$ is the distribution function of X, the random variable U is uniformly distributed between 0 and 1.

Given n observations $X_{(1)}, \ldots, X_{(n)}$, the values $U_{(i)} = F(X_{(i)})$ are computed by applying the transformation, as shown in the following sections.

The HISTOGRAM statement provides three EDF tests:

- Kolmogorov-Smirnov
- Anderson-Darling
- Cramér-von Mises

These tests are based on various measures of the discrepancy between the empirical distribution function $F_n(x)$ and the proposed parametric cumulative distribution function $F(x)$.

The following sections provide formal definitions of the EDF statistics.

Kolmogorov-Smirnov Statistic The Kolmogorov-Smirnov statistic (D) is defined as

$$D = \sup_x |F_n(x) - F(x)|$$

The Kolmogorov-Smirnov statistic belongs to the supremum class of EDF statistics. This class of statistics is based on the largest vertical difference between $F(x)$ and $F_n(x)$.

The Kolmogorov-Smirnov statistic is computed as the maximum of D^+ and D^-, where D^+ is the largest vertical distance between the EDF and the distribution function when the EDF is greater than the distribution function, and D^- is the largest vertical distance when the EDF is less than the distribution function.

$$\begin{aligned} D^+ &= \max_i \left(\frac{i}{n} - U_{(i)}\right) \\ D^- &= \max_i \left(U_{(i)} - \frac{i-1}{n}\right) \\ D &= \max\left(D^+, D^-\right) \end{aligned}$$

Anderson-Darling Statistic The Anderson-Darling statistic and the Cramér-von Mises statistic belong to the quadratic class of EDF statistics. This class of statistics is based on the squared difference $(F_n(x) - F(x))^2$. Quadratic statistics have the following general form:

$$Q = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, \psi(x) dF(x)$$

The function $\psi(x)$ weights the squared difference $(F_n(x) - F(x))^2$.

The Anderson-Darling statistic (A^2) is defined as

$$A^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \left[F(x)\left(1 - F(x)\right)\right]^{-1} dF(x)$$

Here the weight function is $\psi(x) = \left[F(x)\left(1 - F(x)\right)\right]^{-1}$.

The Anderson-Darling statistic is computed as

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^{n} \left[(2i-1) \log U_{(i)} + (2n + 1 - 2i) \log\left(\{1 - U_{(i)}\right)\right]$$

Cramér-von Mises Statistic The Cramér-von Mises statistic (W^2) is defined as

$$W^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, dF(x)$$

Here the weight function is $\psi(x) = 1$.

The Cramér-von Mises statistic is computed as

$$W^2 = \sum_{i=1}^{n} \left(U_{(i)} - \frac{2i-1}{2n}\right)^2 + \frac{1}{12n}$$

Probability Values for EDF Tests Once the EDF test statistics are computed, the associated probability values (p-values) must be calculated.

For the Gumbel, inverse Gaussian, generalized Pareto, and Rayleigh distributions, the procedure computes associated probability values (p-values) by resampling from the estimated distribution. It generates k random samples of size n, where k is specified by the EDFNSAMPLES= option and n is the number of observations in the original data. EDF test statistics are computed for each sample, and the p-value is the proportion of samples whose EDF statistic is greater than or equal to the statistic computed for the original data. You can use the EDFSEED= option to specify a seed value for generating the sample values.

For the beta, exponential, gamma, lognormal, normal, power function, and Weibull distributions, the CAPABILITY procedure uses internal tables of probability levels similar to those given by D'Agostino and Stephens (1986). If the value is between two probability levels, then linear interpolation is used to estimate the probability value. The probability value depends upon the parameters that are known and the parameters that are estimated for the distribution you are fitting. Table 6.23 summarizes different combinations of estimated parameters for which EDF tests are available.

Table 6.23 Availability of EDF Tests

Distribution	Parameters			Tests Available
	Threshold	Scale	Shape	
Beta	θ known	σ known	α, β known	all
	θ known	σ known	$\alpha, \beta < 5$ unknown	all
Exponential	θ known,	σ known		all
	θ known	σ unknown		all
	θ unknown	σ known		all
	θ unknown	σ unknown		all
Gamma	θ known	σ known	α known	all
	θ known	σ unknown	α known	all
	θ known	σ known	α unknown	all
	θ known	σ unknown	$\alpha > 1$ unknown	all
	θ unknown	σ known	$\alpha > 1$ known	all
	θ unknown	σ unknown	$\alpha > 1$ known	all
	θ unknown	σ known	$\alpha > 1$ unknown	all
	θ unknown	σ unknown	$\alpha > 1$ unknown	all
Lognormal	θ known	ζ known	σ known	all
	θ known	ζ known	σ unknown	A^2 and W^2
	θ known	ζ unknown	σ known	A^2 and W^2
	θ known	ζ unknown	σ unknown	all
	θ unknown	ζ known	$\sigma < 3$ known	all
	θ unknown	ζ known	$\sigma < 3$ unknown	all
	θ unknown	ζ unknown	$\sigma < 3$ known	all
	θ unknown	ζ unknown	$\sigma < 3$ unknown	all
Normal	θ known	σ known		all
	θ known	σ unknown		A^2 and W^2
	θ unknown	σ known		A^2 and W^2
	θ unknown	σ unknown		all

Table 6.23 (*continued*)

Distribution	Parameters			Tests Available
	Threshold	Scale	Shape	
Weibull	θ known	σ known	c known	all
	θ known	σ unknown	c known	A^2 and W^2
	θ known	σ known	c unknown	A^2 and W^2
	θ known	σ unknown	c unknown	A^2 and W^2
	θ unknown	σ known	$c > 2$ known	all
	θ unknown	σ unknown	$c > 2$ known	all
	θ unknown	σ known	$c > 2$ unknown	all
	θ unknown	σ unknown	$c > 2$ unknown	all

Specifications

This section is included in the summary only if you provide specification limits, and it tabulates the limits as well as the observed percentages and estimated percentages outside the limits.

The estimated percentages are computed only if fitted distributions are requested and are based on the probability that an observed value exceeds the specification limits, assuming the fitted distribution. The observed percentages are the percents of observations outside the specification limits.

Indices Using Fitted Curves

This section is included in the summary only if you specify the INDICES option in parentheses after a distribution option, as in the statements that produce Figure 6.16. Standard process capability indices, such as C_p and C_{pk}, are not appropriate if the data are not normally distributed. The INDICES option computes generalizations of the standard indices by using the fact that for the normal distribution, 3σ is both the distance from the lower 0.135 percentile to the median (or mean) and the distance from the median (or mean) to the upper 99.865 percentile. These percentiles are estimated from the fitted distribution, and the appropriate percentile-to-median distances are substituted for 3σ in the standard formulas.

Writing T for the target, LSL and USL for the lower and upper specification limits, and P_α for the 100αth percentile, the generalized capability indices are as follows:

$$CPL = \frac{P_{0.5} - LSL}{P_{0.5} - P_{0.00135}}$$

$$CPU = \frac{USL - P_{0.5}}{P_{0.99865} - P_{0.5}}$$

$$C_p = \frac{USL - LSL}{P_{0.99865} - P_{0.00135}}$$

$$C_{pk} = \min\left(\frac{P_{0.5} - \mathrm{LSL}}{P_{0.5} - P_{0.00135}}, \frac{\mathrm{USL} - P_{0.5}}{P_{0.99865} - P_{0.5}}\right)$$

$$K = 2 \times \frac{\left|\frac{1}{2}(\text{USL} + \text{LSL}) - P_{0.5}\right|}{\text{USL} - \text{LSL}}$$

$$C_{pm} = \frac{\min\left(\frac{T-LSL}{P_{0.5}-P_{0.00135}}, \frac{USL-T}{P_{0.99865}-P_{0.5}}\right)}{\sqrt{1+\left(\frac{\mu-T}{\sigma}\right)^2}}$$

If the data are normally distributed, these formulas reduce to the formulas for the standard capability indices, which are given in the section "Standard Capability Indices" on page 235.

The following guidelines apply to the use of generalized capability indices requested with the INDICES option:

- When you choose the family of parametric distributions for the fitted curve, consider whether an appropriate family can be derived from assumptions about the process.
- Whenever possible, examine the data distribution with a histogram, probability plot, or quantile-quantile plot.
- Apply goodness-of-fit tests to assess how well the parametric distribution models the data.
- Consider whether a generalized index has a meaningful practical interpretation in your application.

At the time of this writing, there is ongoing research concerning the application of generalized capability indices, and it is important to note that other approaches can be used with nonnormal data:

- Transform the data to normality, then compute and report standard capability indices on the transformed scale.
- Report the proportion of nonconforming output estimated from the fitted distribution.
- If it is not possible to adequately model the data distribution with a parametric density, smooth the data distribution with a kernel density estimate and simply report the proportion of nonconforming output.

Refer to Rodriguez and Bynum (1992) for additional discussion.

Histogram Intervals

This section is included in the summary only if you specify the MIDPERCENTS option in parentheses after the distribution option, as in the statements that produce Figure 6.16. This table lists the interval midpoints along with the observed and estimated percentages of the observations that lie in the interval. The estimated percentages are based on the fitted distribution.

In addition, you can specify the MIDPERCENTS option to request a table of interval midpoints with the observed percent of observations that lie in the interval. See the entry for the MIDPERCENTS option on page 323.

Quantiles

This table lists observed and estimated quantiles. You can use the PERCENTS= option to specify the list of quantiles to appear in this list. The list in Figure 6.16 is the default list. See the entry for the PERCENTS= option on page 328.

Output Data Sets

You can create two output data sets with the HISTOGRAM statement: the OUTFIT= data set and the OUTHISTOGRAM= data set. These data sets are described in the following sections.

OUTFIT= Data Set

The OUTFIT= data set contains the parameters of fitted density curves, information about chi-square and EDF goodness-of-fit tests, specification limit information, and capability indices based on the fitted distribution. Because you can specify multiple HISTOGRAM statements with the CAPABILITY procedure, you can create several OUTFIT= data sets. For each variable plotted with the HISTOGRAM statement, the OUTFIT= data set contains one observation for each fitted distribution requested in the HISTOGRAM statement. If you use a BY statement, the OUTFIT= data set contains several observations for each BY group (one observation for each variable and fitted density combination). ID variables are not saved in the OUTFIT= data set.

The OUTFIT= data set contains the variables listed in Table 6.24. By default, an OUTFIT= data set contains _MIDPT1_ and _MIDPTN_ variables, whose values identify histogram intervals by their midpoints. When the ENDPOINTS= or NENDPOINTS option is specified, intervals are identified by endpoint values instead. If the RTINCLUDE option is specified, the variables _MAXPT1_ and _MAXPTN_ contain upper endpoint values. Otherwise, the variables _MINPT1_ and _MINPTN_ contain lower endpoint values.

Table 6.24 Variables in the OUTFIT= Data Set

Variable	Description
ADASQ	Anderson-Darling EDF goodness-of-fit statistic
ADP	p-value for Anderson-Darling EDF goodness-of-fit test
CHISQ	chi-square goodness-of-fit statistic
_CP_	generalized capability index C_p based on the fitted curve
CPK	generalized capability index C_{pk} based on the fitted curve
CPL	generalized capability index CPL based on the fitted curve
CPM	generalized capability index C_{pm} based on the fitted curve
CPU	generalized capability index CPU based on the fitted curve
CURVE	name of fitted distribution (abbreviated to 8 characters)
CVMWSQ	Cramer-von Mises EDF goodness-of-fit statistic
CVMP	p-value for Cramer-von Mises EDF goodness-of-fit test
_DF_	degrees of freedom for chi-square goodness-of-fit test
ESTGTR	estimated percent of population greater than upper specification limit
ESTLSS	estimated percent of population less than lower specification limit
ESTSTD	estimated standard deviation
EXPECT	estimated mean
_K_	generalized capability index K based on the fitted curve
KSD	Kolmogorov-Smirnov EDF goodness-of-fit statistic
KSP	p-value for Kolmogorov-Smirnov EDF goodness-of-fit test

Table 6.24 *(continued)*

Variable	Description
LOCATN	location parameter for fitted distribution. For the Gumbel, inverse Gaussian, and normal distributions, this is either the value of μ specified with the MU= option or the value estimated by the procedure. For all other distributions, this is either the value specified or estimated according to the THETA= option, or zero.
LSL	lower specification limit
MAXPT1	upper endpoint of first interval used to calculate the value of the chi-square statistic.
MAXPTN	upper endpoint of last interval used to calculate the value of the chi-square statistic.
MIDPT1	midpoint of first interval used to calculate the value of the chi-square statistic. This is the leftmost interval that contains at least one value of the variable.
MIDPTN	midpoint of last interval used to calculate the value of the chi-square statistic. This is the rightmost interval that contains at least one value of the variable.
MINPT1	lower endpoint of first interval used to calculate the value of the chi-square statistic.
MINPTN	lower endpoint of last interval used to calculate the value of the chi-square statistic.
OBSGTR	observed percent of data greater than upper specification limit
OBSLSS	observed percent of data less than the lower specification limit
PCHISQ	p-value for chi-square goodness-of-fit test
SCALE	value of scale parameter for fitted distribution. For the lognormal distribution, this is the value of ζ specified or estimated according to the ZETA= option. For all other distributions, this is the value specified or estimated according to the SIGMA= option.
SHAPE1	value of shape parameter for fitted distribution. For the beta, gamma, generalized Pareto, and power function distributions, this is the value of α, either specified with the ALPHA= option or estimated by the procedure. For the lognormal distribution, this is the value of σ, either specified with the SIGMA= option or estimated by the procedure. For the Weibull distribution, this is the value of c, either specified with the C= option or estimated by the procedure. For the Johnson S_B and S_U distributions, this is the value of δ, either specified with the DELTA= option or estimated by the procedure. For distributions without a shape parameter (Gumbel, normal, exponential, and Rayleigh distributions), _SHAPE1_ is set to missing.

Table 6.24 *(continued)*

Variable	Description
SHAPE2	value of shape parameter for fitted distribution. For the beta distribution, this is the value of β, either specified with the BETA= option or estimated by the procedure. For the Johnson S_B and S_U distributions, this is the value of γ, either specified with the GAMMA= option or estimated by the procedure. For all other distributions, _SHAPE2_ is set to missing.
TARGET	target value
USL	upper specification limit
VAR	variable name
WIDTH	width of histogram interval

OUTHISTOGRAM= Data Set

The OUTHISTOGRAM= data set contains information about histogram intervals. Because you can specify multiple HISTOGRAM statements with the CAPABILITY procedure, you can create multiple OUTHISTOGRAM= data sets.

The data set contains a group of observations for each variable plotted with the HISTOGRAM statement. The group contains an observation for each interval of the histogram, beginning with the leftmost interval that contains a value of the variable and ending with the rightmost interval that contains a value of the variable. These intervals will not necessarily coincide with the intervals displayed in the histogram because the histogram may be padded with empty intervals at either end. If you superimpose one or more fitted curves on the histogram, the OUTHISTOGRAM= data set contains multiple groups of observations for each variable (one group for each curve). If you use a BY statement, the OUTHISTOGRAM= data set contains groups of observations for each BY group. ID variables are not saved in the OUTHISTOGRAM= data set.

The OUTHISTOGRAM= data set contains the variables listed in Table 6.25. By default, an OUTHISTOGRAM= data set contains the _MIDPT_ variable, whose values identify histogram intervals by their midpoints. When the ENDPOINTS= or NENDPOINTS option is specified, intervals are identified by endpoint values instead. If the RTINCLUDE option is specified, the _MAXPT_ variable contains an interval's upper endpoint value. Otherwise, the _MINPT_ variable contains the interval's lower endpoint value.

Table 6.25 Variables in the OUTHISTOGRAM= Data Set

Variable	Description
COUNT	number of variable values in histogram interval
CURVE	name of fitted distribution (if requested in HISTOGRAM statement)
EXPPCT	estimated percent of population in histogram interval determined from optional fitted distribution
MAXPT	upper endpoint of histogram interval
MIDPT	midpoint of histogram interval
MINPT	lower endpoint of histogram interval
OBSPCT	percent of variable values in histogram interval
VAR	variable name

OUTKERNEL= Output Data Set

An OUTKERNEL= data set contains information about kernel density estimates requested with the KERNEL option. Because you can specify multiple HISTOGRAM statements with the CAPABILITY procedure, you can create multiple OUTKERNEL= data sets.

An OUTKERNEL= data set contains a group of observations for each kernel density estimate requested with the HISTOGRAM statement. These observations span a range of analysis variable values recorded in the _VALUE_ variable. The procedure determines the increment between values, and therefore the number of observations in the group. The variable _DENSITY_ contains the kernel density calculated for the corresponding analysis variable value.

When a density curve is overlaid on a histogram, the curve is scaled so that the area under the curve equals the total area of the histogram bars. The scaled density values are saved in the variable _COUNT_, _PERCENT_, or _PROPORTION_, depending on the histogram's vertical axis scale, determined by the VSCALE= option. Only one of these variables appears in a given OUTKERNEL= data set.

Table 6.26 lists the variables in an OUTKERNEL= data set.

Table 6.26 Variables in the OUTKERNEL= Data Set

Variable	Description
_C_	standardized bandwidth parameter
COUNT	kernel density scaled for VSCALE=COUNT
DENSITY	kernel density
PERCENT	kernel density scaled for VSCALE=PERCENT (default)
PROPORTION	kernel density scaled for VSCALE=PROPORTION
TYPE	kernel function
VALUE	variable value at which kernel function is calculated
VAR	variable name

ODS Tables

The following table summarizes the ODS tables related to fitted distributions that you can request with the HISTOGRAM statement.

Table 6.27 ODS Tables Produced with the HISTOGRAM Statement

Table Name	Description	Option
Bins	histogram bins	MIDPERCENTS suboption with any distribution option, such as NORMAL(MIDPERCENTS)
FitIndices	capability indices computed from fitted distribution	INDICES suboption with any distribution option, such as LOGNORMAL(INDICES)
FitQuantiles	quantiles of fitted distribution	any distribution option such as NORMAL
GoodnessOfFit	goodness-of-fit tests for fitted distribution	any distribution option such as NORMAL
ParameterEstimates	parameter estimates for fitted distribution	any distribution option such as NORMAL
Specifications	percents outside specification limits based on empirical and fitted distributions	any distribution option such as NORMAL

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. HISTOGRAM options used to control the appearance of traditional graphics are ignored for ODS Graphics output.

When ODS Graphics is in effect, the HISTOGRAM statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 6.28.

Table 6.28 ODS Graphics Produced by the HISTOGRAM Statement

ODS Graph Name	Plot Description
Histogram	histogram

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

SYMBOL and PATTERN Statement Options

In earlier releases of SAS/QC software, graphical features (such as colors and line types) of specification lines, histogram bars, and fitted curves were controlled with options in SYMBOL and PATTERN statements when producing traditional graphics. These options are still supported, although they have been superseded by options in the HISTOGRAM and SPEC statements. The following tables summarize the two sets of options. **NOTE:** These statements have no effect on ODS Graphics output.

Table 6.29 Graphical Enhancement of Histogram Outlines and Specification Lines

Feature	Statement and Options	Alternative Statement and Options
Outline of Histogram Bars	HISTOGRAM Statement	SYMBOL1 Statement
color	CBARLINE=*color*	C=*color*
width		W=*value*
Target Reference Line	SPEC Statement	SYMBOL1 Statement
position	TARGET=*value*	
color	CTARGET=*color*	C=*color*
line type	LTARGET=*linetype*	L=*linetype*
width	WTARGET=*value*	W=*value*
Lower Specification Line	SPEC Statement	SYMBOL2 Statement
position	LSL=*value*	
color	CLSL=*color*	C=*color*
line type	LLSL=*linetype*	L=*linetype*
width	WLSL=*value*	W=*value*
Upper Specification Line	SPEC Statement	SYMBOL3 Statement
position	USL=*value*	
color	CUSL=*color*	C=*color*
line type	LUSL=*linetype*	L=*linetype*
width	WUSL=*value*	W=*value*

Table 6.30 Graphical Enhancement of Areas Under Histograms and Curves

Area Under Histogram or Curve	Statement and Options	Alternative Statement and Options
Histogram or Curve	HISTOGRAM Statement	PATTERN1 Statement
pattern	PFILL=*pattern*	V=*pattern*
color	CFILL=*color*	C=*color*
Left of Lower Specification Limit	SPEC Statement	PATTERN2 Statement
pattern	PLEFT=*pattern*	V=*pattern*
color	CLEFT=*color*	C=*color*
Right of Upper Specification Limit	SPEC Statement	PATTERN3 Statement
pattern	PRIGHT=*pattern*	V=*pattern*
color	CRIGHT=*color*	C=*color*

Table 6.31 Graphical Enhancement of Fitted Curves

Feature	Statement and Options	Alternative Statement and Options
Normal Curve	Normal-options	SYMBOL4 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Lognormal Curve	Lognormal-options	SYMBOL5 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Exponential Curve	Exponential-options	SYMBOL6 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Weibull Curve	Weibull-options	SYMBOL7 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Gamma Curve	Gamma-options	SYMBOL8 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Beta Curve	Beta-options	SYMBOL9 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Johnson S_B Curve	S_B-options	SYMBOL10 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Johnson S_U Curve	S_U-options	SYMBOL11 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Rayleigh Curve	Rayleigh-options	SYMBOL12 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Generalized Pareto Curve	Pareto-options	SYMBOL13 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*

Table 6.31 (*continued*)

Feature	Statement and Options	Alternative Statement and Options
Gumbel Curve	Gumbel-options	SYMBOL14 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Power Function Curve	Power-options	SYMBOL15 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*
Inverse Gaussian Curve	IGauss-options	SYMBOL16 Statement
color	COLOR=*color*	C=*color*
line type	L=*linetype*	L=*linetype*
width	W=*value*	W=*value*

Examples: HISTOGRAM Statement

This section provides advanced examples of the HISTOGRAM statement.

Example 6.8: Fitting a Beta Curve

NOTE: See *Fitting a Beta Curve on a Histogram* in the SAS/QC Sample Library.

You can use a beta distribution to model the distribution of a quantity that is known to vary between lower and upper bounds. In this example, a manufacturing company uses a robotic arm to attach hinges on metal sheets. The attachment point should be offset 10.1 mm from the left edge of the sheet. The actual offset varies between 10.0 and 10.5 mm due to variation in the arm. Offsets for 50 attachment points are saved in the following data set:

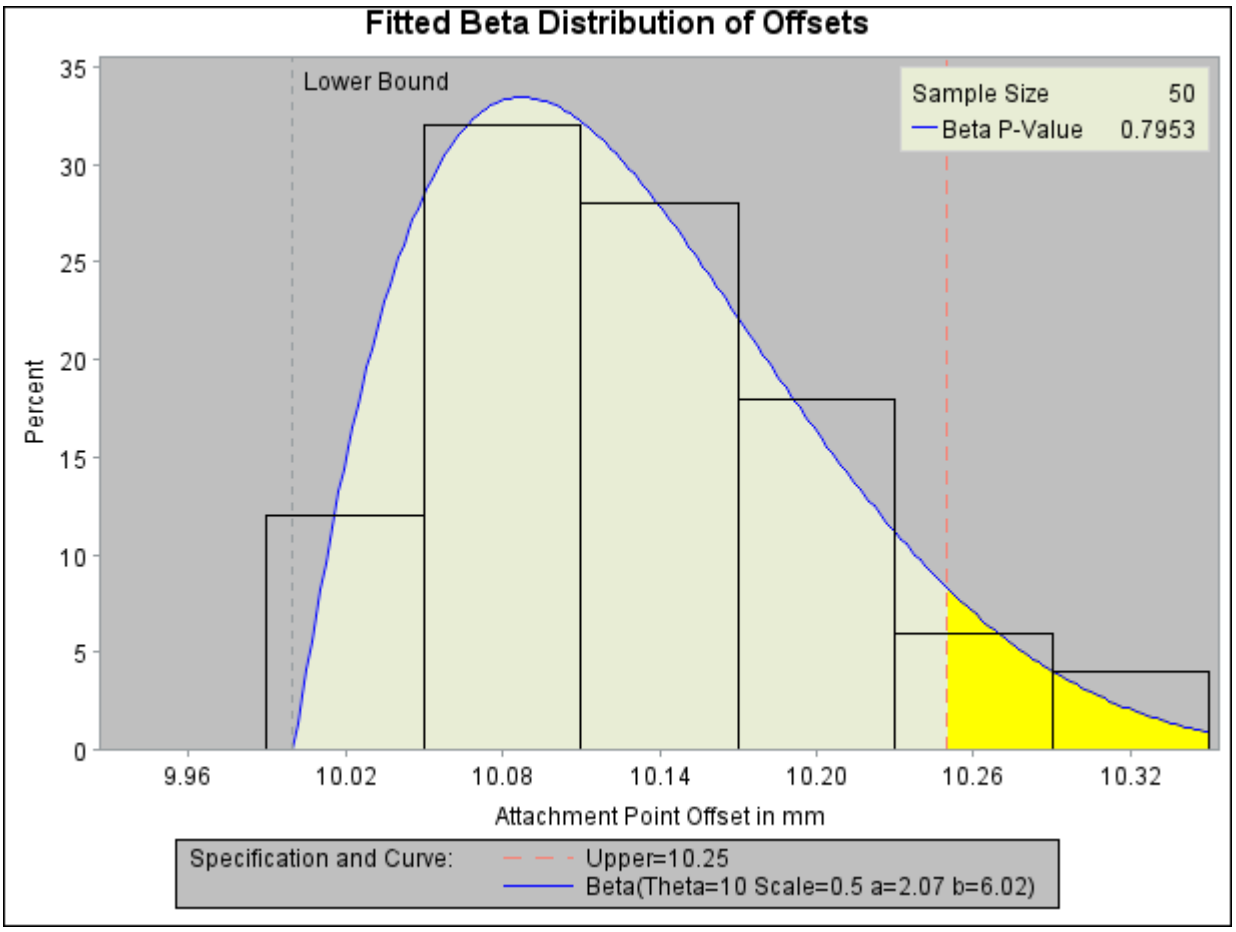
```
data Measures;
   input Length @@;
   label Length = 'Attachment Point Offset in mm';
   datalines;
10.147 10.070 10.032 10.042 10.102
10.034 10.143 10.278 10.114 10.127
10.122 10.018 10.271 10.293 10.136
10.240 10.205 10.186 10.186 10.080
10.158 10.114 10.018 10.201 10.065
10.061 10.133 10.153 10.201 10.109
10.122 10.139 10.090 10.136 10.066
10.074 10.175 10.052 10.059 10.077
10.211 10.122 10.031 10.322 10.187
10.094 10.067 10.094 10.051 10.174
;
```

The following statements create a histogram with a fitted beta density curve:

```
ods graphics off;
legend2 frame cframe=ligr cborder=black position=center;
title1 'Fitted Beta Distribution of Offsets';
proc capability data=Measures;
   specs usl=10.25 lusl=20 cusl=salmon cright=yellow pright=solid;
   histogram Length /
      beta(theta=10 scale=0.5 color=blue fill)
      cfill     = ywh
      cframe    = ligr
      href      = 10
      hreflabel = 'Lower Bound'
      lhref     = 2
      legend    = legend2
      vaxis     = axis1;
   axis1 label=(a=90 r=0);
   inset n = 'Sample Size'
         beta(pchisq = 'P-Value') / pos=ne  cfill=ywh;
run;
```

The histogram is shown in Output 6.8.1. The THETA= *beta-option* specifies the lower threshold. The SCALE= *beta-option* specifies the range between the lower threshold and the upper threshold (in this case, 0.5 mm). Note that in general, the default THETA= and SCALE= values are zero and one, respectively.

Output 6.8.1 Superimposing a Histogram with a Fitted Beta Curve



The FILL *beta-option* specifies that the area under the curve is to be filled with the CFILL= color. (If FILL were omitted, the CFILL= color would be used to fill the histogram bars instead.) The CRIGHT= option in the SPEC statement specifies the color under the curve to the right of the upper specification limit. If the CRIGHT= option were not specified, the entire area under the curve would be filled with the CFILL= color. When a lower specification limit is available, you can use the CLEFT= option in the SPEC statement to specify the color under the curve to the left of this limit.

The HREF= option draws a reference line at the lower bound, and the HREFLABEL= option adds the label *Lower Bound*. The option LHREF=2 specifies a dashed line type. The INSET statement adds an inset with the sample size and the *p*-value for a chi-square goodness-of-fit test.

In addition to displaying the beta curve, the BETA option summarizes the curve fit, as shown in Output 6.8.2. The output tabulates the parameters for the curve, the chi-square goodness-of-fit test whose *p*-value is shown in Output 6.8.1, the observed and estimated percents above the upper specification limit, and the observed and estimated quantiles. For instance, based on the beta model, the percent of offsets greater than the upper specification limit is 6.6%. For computational details, see the section "Formulas for Fitted Curves" on page 336.

Output 6.8.2 Summary of Fitted Beta Distribution

Fitted Beta Distribution of Offsets

The CAPABILITY Procedure
Fitted Beta Distribution for Length (Attachment Point Offset in mm)

Parameters for Beta Distribution

Parameter	Symbol	Estimate
Threshold	Theta	10
Scale	Sigma	0.5
Shape	Alpha	2.06832
Shape	Beta	6.022479
Mean		10.12782
Std Dev		0.072339

Goodness-of-Fit Tests for Beta Distribution

Test	Statistic		DF	p Value	
Chi-Square	**Chi-Sq**	1.02463588	3	**Pr > Chi-Sq**	0.795

Percent Outside Specifications for Beta Distribution

Upper Limit	
USL	10.250000
Obs Pct > USL	8.000000
Est Pct > USL	6.618103

Quantiles for Beta Distribution

	Quantile	
Percent	**Observed**	**Estimated**
1.0	10.0180	10.0124
5.0	10.0310	10.0285
10.0	10.0380	10.0416
25.0	10.0670	10.0718
50.0	10.1220	10.1174
75.0	10.1750	10.1735
90.0	10.2255	10.2292
95.0	10.2780	10.2630
99.0	10.3220	10.3237

Example 6.9: Fitting Lognormal, Weibull, and Gamma Curves

NOTE: See *Superimposing Fitted Curves on a Histogram* in the SAS/QC Sample Library.

To find an appropriate model for a process distribution, you should consider curves from several distribution families. As shown in this example, you can use the HISTOGRAM statement to fit more than one type of distribution and display the density curves on the same histogram.

The gap between two plates is measured (in cm) for each of 50 welded assemblies selected at random from the output of a welding process assumed to be in statistical control. The lower and upper specification limits for the gap are 0.3 cm and 0.8 cm, respectively. The measurements are saved in a data set named Plates.

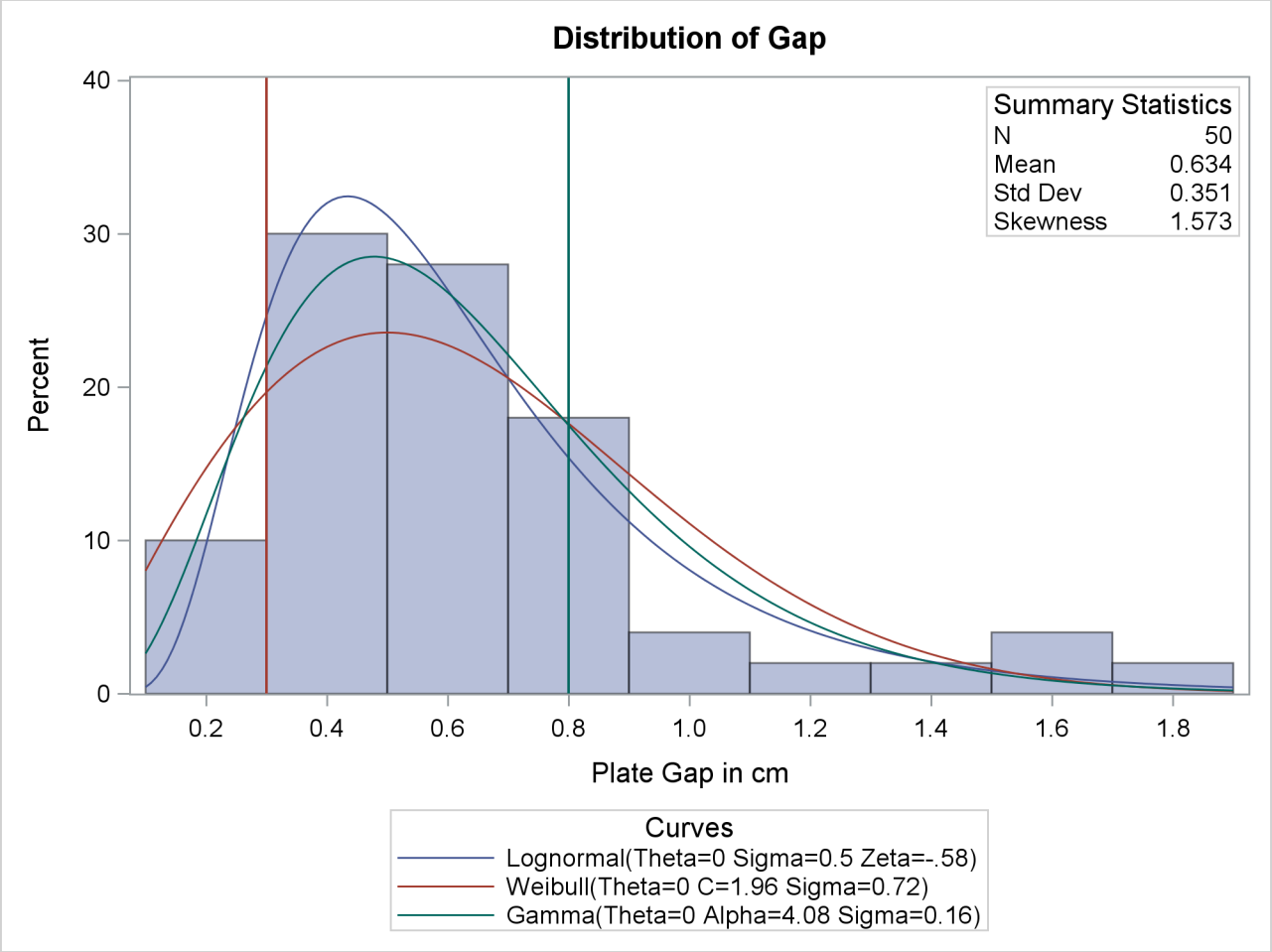
```
data Plates;
   label Gap='Plate Gap in cm';
   input Gap @@;
   datalines;
0.746  0.357  0.376  0.327  0.485 1.741  0.241  0.777  0.768  0.409
0.252  0.512  0.534  1.656  0.742 0.378  0.714  1.121  0.597  0.231
0.541  0.805  0.682  0.418  0.506 0.501  0.247  0.922  0.880  0.344
0.519  1.302  0.275  0.601  0.388 0.450  0.845  0.319  0.486  0.529
1.547  0.690  0.676  0.314  0.736 0.643  0.483  0.352  0.636  1.080
;
```

The following statements fit three distributions (lognormal, Weibull, and gamma) and display their density curves on a single histogram:

```
ods graphics on;
proc capability data=Plates;
   var Gap;
   specs  lsl = 0.3 usl = 0.8;
   histogram /
       midpoints=0.2 to 1.8 by 0.2
       lognormal
       weibull
       gamma
       nospeclegend;
   inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
         / pos = ne  header = 'Summary Statistics';
run;
```

The LOGNORMAL, WEIBULL, and GAMMA options superimpose fitted curves on the histogram in Output 6.9.1. Note that a threshold parameter $\theta = 0$ is assumed for each curve. In applications where the threshold is not zero, you can specify θ with the THETA= option.

Output 6.9.1 Superimposing a Histogram with Fitted Curves



The LOGNORMAL, WEIBULL, and GAMMA options also produce the summaries for the fitted distributions shown in Output 6.9.2, Output 6.9.3, and Output 6.9.4.

Output 6.9.2 Summary of Fitted Lognormal Distribution

The CAPABILITY Procedure
Fitted Lognormal Distribution for Gap (Plate Gap in cm)

Parameters for Lognormal Distribution		
Parameter	**Symbol**	**Estimate**
Threshold	Theta	0
Scale	Zeta	-0.58375
Shape	Sigma	0.499546
Mean		0.631932
Std Dev		0.336436

Output 6.9.2 *continued*

Goodness-of-Fit Tests for Lognormal Distribution					
Test		**Statistic**	**DF**	**p Value**	
Kolmogorov-Smirnov	**D**	0.06441431		**Pr > D**	>0.150
Cramer-von Mises	**W-Sq**	0.02823022		**Pr > W-Sq**	>0.500
Anderson-Darling	**A-Sq**	0.24308402		**Pr > A-Sq**	>0.500
Chi-Square	**Chi-Sq**	7.51762213	6	**Pr > Chi-Sq**	0.276

Percent Outside Specifications for Lognormal Distribution			
Lower Limit		**Upper Limit**	
LSL	0.300000	**USL**	0.800000
Obs Pct < LSL	10.000000	**Obs Pct > USL**	20.000000
Est Pct < LSL	10.719540	**Est Pct > USL**	23.519008

Quantiles for Lognormal Distribution		
	Quantile	
Percent	**Observed**	**Estimated**
1.0	0.23100	0.17449
5.0	0.24700	0.24526
10.0	0.29450	0.29407
25.0	0.37800	0.39825
50.0	0.53150	0.55780
75.0	0.74600	0.78129
90.0	1.10050	1.05807
95.0	1.54700	1.26862
99.0	1.74100	1.78313

Output 6.9.2 provides four goodness-of-fit tests for the lognormal distribution: the chi-square test and three tests based on the EDF (Anderson-Darling, Cramer-von Mises, and Kolmogorov-Smirnov). See "Chi-Square Goodness-of-Fit Test" on page 350 and "EDF Goodness-of-Fit Tests" on page 351 for more information. The EDF tests are superior to the chi-square test because they are not dependent on the set of midpoints used for the histogram.

At the $\alpha = 0.10$ significance level, all four tests support the conclusion that the two-parameter lognormal distribution with scale parameter $\hat{\zeta} = -0.58$, and shape parameter $\hat{\sigma} = 0.50$ provides a good model for the distribution of plate gaps.

Output 6.9.3 Summary of Fitted Weibull Distribution

The CAPABILITY Procedure
Fitted Weibull Distribution for Gap (Plate Gap in cm)

Parameters for Weibull Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	0.719208
Shape	C	1.961159
Mean		0.637641
Std Dev		0.339248

Goodness-of-Fit Tests for Weibull Distribution

Test		Statistic	DF		p Value
Cramer-von Mises	**W-Sq**	0.1593728		**Pr > W-Sq**	0.016
Anderson-Darling	**A-Sq**	1.1569354		**Pr > A-Sq**	<0.010
Chi-Square	**Chi-Sq**	15.0252997	6	**Pr > Chi-Sq**	0.020

Percent Outside Specifications for Weibull Distribution

Lower Limit		Upper Limit	
LSL	0.300000	**USL**	0.800000
Obs Pct < LSL	10.000000	**Obs Pct > USL**	20.000000
Est Pct < LSL	16.473319	**Est Pct > USL**	29.165543

Quantiles for Weibull Distribution

	Quantile	
Percent	Observed	Estimated
1.0	0.23100	0.06889
5.0	0.24700	0.15817
10.0	0.29450	0.22831
25.0	0.37800	0.38102
50.0	0.53150	0.59661
75.0	0.74600	0.84955
90.0	1.10050	1.10040
95.0	1.54700	1.25842
99.0	1.74100	1.56691

Output 6.9.3 provides two EDF goodness-of-fit tests for the Weibull distribution: the Anderson-Darling and the Cramer-von Mises tests. (See Table 6.23 for a complete list of the EDF tests available in the HISTOGRAM statement.) The probability values for the chi-square and EDF tests are all less than 0.10, indicating that the data do not support a Weibull model.

Output 6.9.4 Summary of Fitted Gamma Distribution

The CAPABILITY Procedure
Fitted Gamma Distribution for Gap (Plate Gap in cm)

Parameters for Gamma Distribution

Parameter	Symbol	Estimate
Threshold	Theta	0
Scale	Sigma	0.155198
Shape	Alpha	4.082646
Mean		0.63362
Std Dev		0.313587

Goodness-of-Fit Tests for Gamma Distribution

Test		Statistic	DF	p Value	
Kolmogorov-Smirnov	**D**	0.0969533		**Pr > D**	>0.250
Cramer-von Mises	**W-Sq**	0.0739847		**Pr > W-Sq**	>0.250
Anderson-Darling	**A-Sq**	0.5810661		**Pr > A-Sq**	0.137
Chi-Square	**Chi-Sq**	12.3075959	6	**Pr > Chi-Sq**	0.055

Percent Outside Specifications for Gamma Distribution

Lower Limit		Upper Limit	
LSL	0.300000	**USL**	0.800000
Obs Pct < LSL	10.000000	**Obs Pct > USL**	20.000000
Est Pct < LSL	12.111039	**Est Pct > USL**	25.696522

Quantiles for Gamma Distribution

	Quantile	
Percent	**Observed**	**Estimated**
1.0	0.23100	0.13326
5.0	0.24700	0.21951
10.0	0.29450	0.27938
25.0	0.37800	0.40404
50.0	0.53150	0.58271
75.0	0.74600	0.80804
90.0	1.10050	1.05392
95.0	1.54700	1.22160
99.0	1.74100	1.57939

Output 6.9.4 provides four goodness-of-fit tests for the gamma distribution. The probability value for the chi-square test is less than 0.10, indicating that the data do not support a gamma model.

Based on this analysis, the fitted lognormal distribution is the best model for the distribution of plate gaps. You can use this distribution to calculate useful quantities. For instance, you can compute the probability that the gap of a randomly sampled plate exceeds the upper specification limit, as follows:

$$\begin{aligned} \Pr[\text{gap} > \text{USL}] &= \Pr\left[Z > \tfrac{1}{\sigma}(\log(\text{USL} - \theta) - \zeta)\right] \\ &= 1 - \Phi\left[\tfrac{1}{\sigma}(\log(\text{USL} - \theta) - \zeta)\right] \end{aligned}$$

where Z has a standard normal distribution, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. Note that $\Phi(\cdot)$ can be computed with the DATA step function PROBNORM. In this example, USL = 0.8 and $\Pr[\text{gap} > 0.8] = 0.2352$. This value is expressed as a percent (*Est Pct* > *USL*) in Output 6.9.2.

Example 6.10: Comparing Goodness-of-Fit Tests

NOTE: See *Comparing Goodness-of-Fit Tests* in the SAS/QC Sample Library.

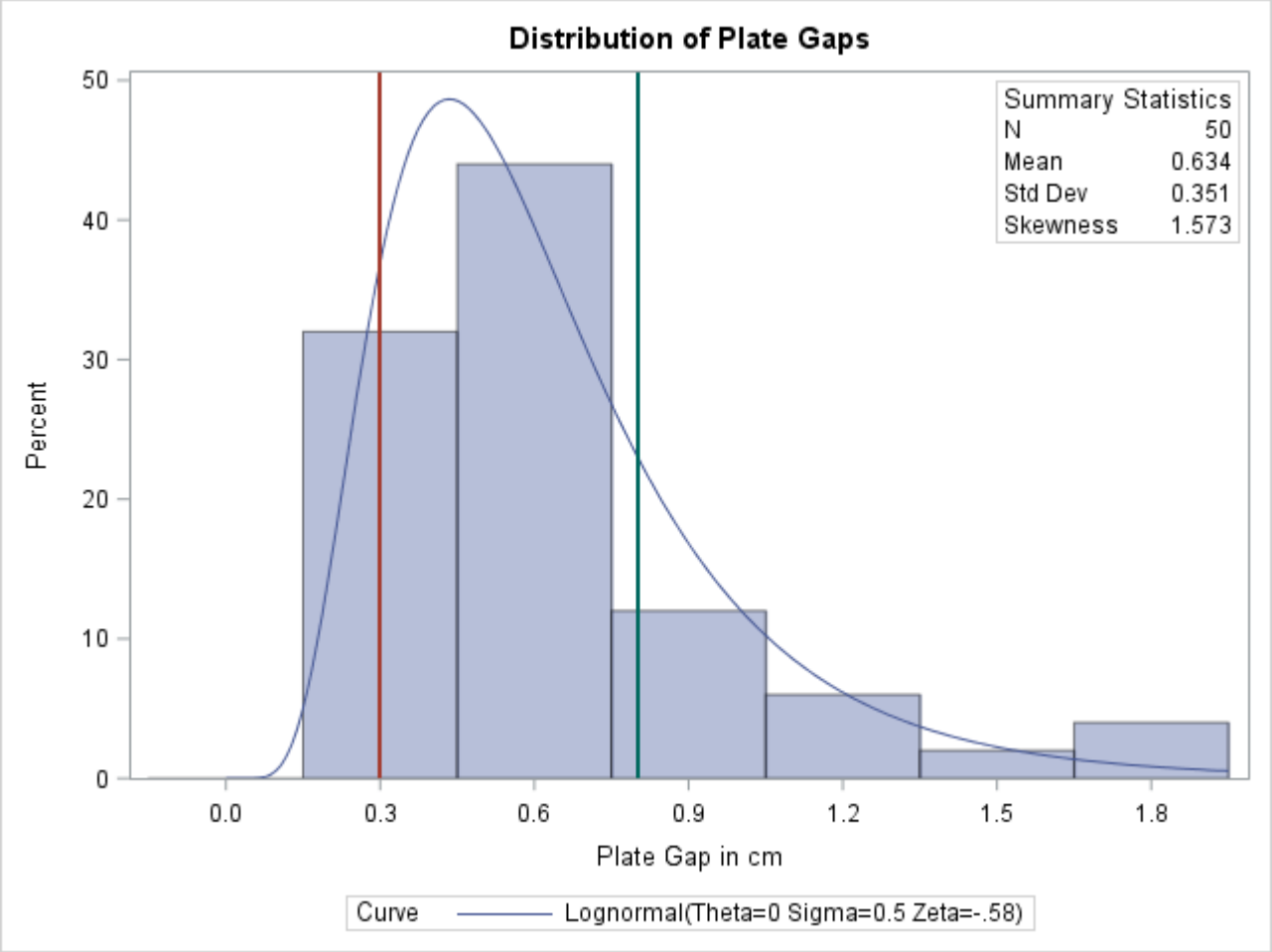
A weakness of the chi-square goodness-of-fit test is its dependence on the choice of histogram midpoints. An advantage of the EDF tests is that they give the same results regardless of the midpoints, as illustrated in this example.

In Example 6.9, the option MIDPOINTS=0.2 TO 1.8 BY 0.2 was used to specify the histogram midpoints for Gap. The following statements refit the lognormal distribution by using default midpoints (0.3 to 1.8 by 0.3).

```
data Plates;
   label Gap='Plate Gap in cm';
   input Gap @@;
   datalines;
0.746  0.357  0.376  0.327  0.485 1.741  0.241  0.777  0.768  0.409
0.252  0.512  0.534  1.656  0.742 0.378  0.714  1.121  0.597  0.231
0.541  0.805  0.682  0.418  0.506 0.501  0.247  0.922  0.880  0.344
0.519  1.302  0.275  0.601  0.388 0.450  0.845  0.319  0.486  0.529
1.547  0.690  0.676  0.314  0.736 0.643  0.483  0.352  0.636  1.080
;


title1 'Distribution of Plate Gaps';
proc capability data=Plates noprint;
   var Gap;
   specs  lsl = 0.3 usl = 0.8;
   histogram / lognormal
               nospeclegend
               odstitle  = title;
   inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3) /
      pos    = ne
      header = 'Summary Statistics';
run;
```

The histogram is shown in Output 6.10.1.

Output 6.10.1 Lognormal Curve Fit with Default Midpoints



A summary of the lognormal fit is shown in Output 6.10.2. The *p*-value for the chi-square goodness-of-fit test is 0.082. Because this value is less than 0.10 (a typical cutoff level), the conclusion is that the lognormal distribution is not an appropriate model for the data. This is the *opposite* conclusion drawn from the chi-square test in Example 6.9, which is based on a different set of midpoints and has a *p*-value of 0.2756 (see Output 6.9.2). Moreover, the results of the EDF goodness-of-fit tests are the same because these tests do not depend on the midpoints. When available, the EDF tests provide more powerful alternatives to the chi-square test. For a thorough discussion of EDF tests, refer to D'Agostino and Stephens (1986).

Output 6.10.2 Printed Output for the Lognormal Curve

Distribution of Plate Gaps

The CAPABILITY Procedure
Fitted Lognormal Distribution for Gap (Plate Gap in cm)

Parameters for Lognormal Distribution		
Parameter	**Symbol**	**Estimate**
Threshold	Theta	0
Scale	Zeta	-0.58375
Shape	Sigma	0.499546
Mean		0.631932
Std Dev		0.336436

Goodness-of-Fit Tests for Lognormal Distribution					
Test		**Statistic**	**DF**	**p Value**	
Kolmogorov-Smirnov	**D**	0.06441431		**Pr > D**	>0.150
Cramer-von Mises	**W-Sq**	0.02823022		**Pr > W-Sq**	>0.500
Anderson-Darling	**A-Sq**	0.24308402		**Pr > A-Sq**	>0.500
Chi-Square	**Chi-Sq**	6.69789360	3	**Pr > Chi-Sq**	0.082

Example 6.11: Computing Capability Indices for Nonnormal Distributions

NOTE: See *Nonnormal Distribution Capability Indices* in the SAS/QC Sample Library.

Standard capability indices such as C_{pk} are generally considered meaningful only if the process output has a normal (or reasonably normal) distribution. In practice, however, many processes have nonnormal distributions. This example, which is a continuation of Example 6.9 and Example 6.10, shows how you can use the HISTOGRAM statement to compute generalized capability indices based on fitted nonnormal distributions.

The following statements produce printed output that is partially listed in Output 6.11.1 and Output 6.11.2:

```
data Plates;
   label Gap='Plate Gap in cm';
   input Gap @@;
   datalines;
0.746  0.357  0.376  0.327  0.485 1.741  0.241  0.777  0.768  0.409
0.252  0.512  0.534  1.656  0.742 0.378  0.714  1.121  0.597  0.231
0.541  0.805  0.682  0.418  0.506 0.501  0.247  0.922  0.880  0.344
0.519  1.302  0.275  0.601  0.388 0.450  0.845  0.319  0.486  0.529
1.547  0.690  0.676  0.314  0.736 0.643  0.483  0.352  0.636  1.080
;


proc capability data=Plates checkindices(alpha=0.05);
   specs lsl=0.3 usl= 0.8;
   histogram Gap / lognormal(indices) noplot;
run;
```

The PROC CAPABILITY statement computes the standard capability indices that are shown in Output 6.11.1.

Output 6.11.1 Standard Capability Indices for Variable Gap

Distribution of Plate Gaps

The CAPABILITY Procedure
Variable: Gap (Plate Gap in cm)

Process Capability Indices			
Index	**Value**	**95% Confidence Limits**	
Cp	0.237112	0.190279	0.283853
CPL	0.316422	0.203760	0.426833
CPU	0.157803	0.059572	0.254586
Cpk	0.157803	0.060270	0.255336

Warning: Normality is rejected for alpha = 0.05 using the Shapiro-Wilk test

The CHECKINDICES option in the PROC statement requests a goodness-of-fit test for normality in conjunction with the indices and displays the warning that normality is rejected at the significance level $\alpha = 0.05$.

Example 6.9 concluded that the fitted lognormal distribution summarized in Output 6.9.2 is a good model, so one might consider computing generalized capability indices based on this distribution. These indices are requested with the INDICES option and are shown in Output 6.11.2. Formulas and recommendations for these indices are given in "Indices Using Fitted Curves" on page 354.

Output 6.11.2 Fitted Lognormal Distribution Information

Capability Indices Based on Lognormal Distribution	
Cp	0.210804
CPL	0.595156
CPU	0.124927
Cpk	0.124927

Example 6.12: Computing Kernel Density Estimates

NOTE: See *Superimposing Kernel Density Estimates* in the SAS/QC Sample Library.

This example illustrates the use of kernel density estimates to visualize a nonnormal data distribution.

The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable Length in a SAS data set named Channel:

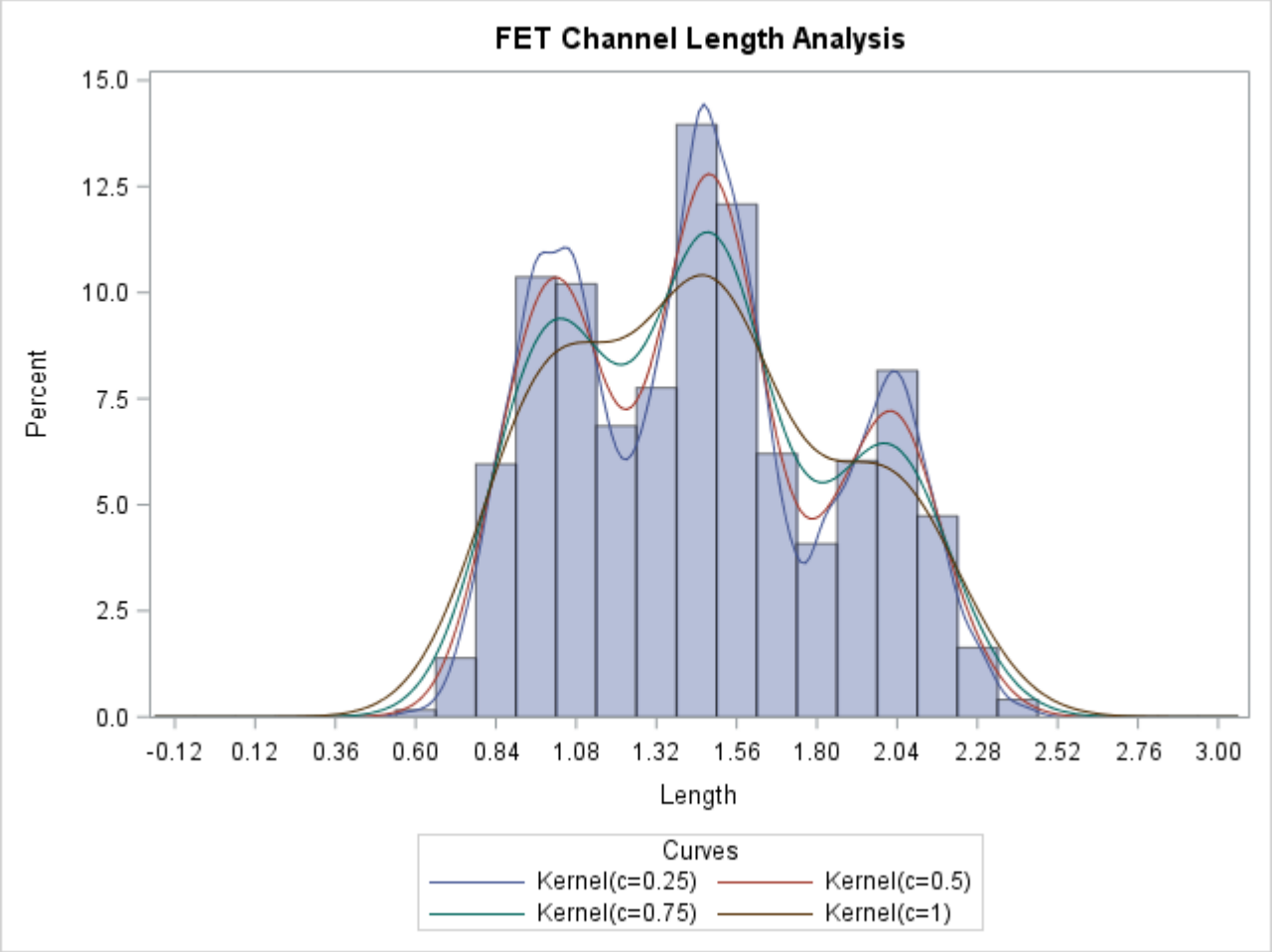
```
data Channel;
   length Lot $ 16;
   input Length @@;
   select;
      when (_n_ <= 425) Lot='Lot 1';
      when (_n_ >= 926) Lot='Lot 3';
      otherwise Lot='Lot 2';
   end;
   datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15
1.11 0.98 0.78 1.09 0.94 1.05 0.89 1.16 0.88 1.19
1.01 1.08 1.19 0.94 0.92 1.27 0.90 0.88 1.38 1.02

   ... more lines ...

2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

When you use kernel density estimates to explore a data distribution, you should try several choices for the bandwidth parameter c because this determines the smoothness and closeness of the fit. You can specify a list of C= values with the KERNEL option to request multiple density estimates, as shown in the following statements:

```
title "FET Channel Length Analysis";
proc capability data=Channel noprint;
   histogram Length / kernel(c = 0.25 0.50 0.75 1.00)
                      odstitle = title;
run;
```

The display, shown in Output 6.12.1, demonstrates the effect of c. In general, larger values of c yield smoother density estimates, and smaller values yield estimates that more closely fit the data distribution.

Output 6.12.1 Multiple Kernel Density Estimates



Output 6.12.1 reveals strong trimodality in the data, which are explored further in "Creating a One-Way Comparative Histogram" on page 276.

Example 6.13: Fitting a Three-Parameter Lognormal Curve

NOTE: See *Three-Parameter Lognormal Distribution* in the SAS/QC Sample Library.

If you request a lognormal fit with the LOGNORMAL option, a *two-parameter* lognormal distribution is assumed. This means that the shape parameter σ and the scale parameter ζ are unknown (unless specified) and that the threshold θ is known (it is either specified with the THETA= option or assumed to be zero).

If it is necessary to estimate θ in addition to ζ and σ, the distribution is referred to as a *three-parameter* lognormal distribution. The equation for this distribution is the same as the equation given in section "Lognormal Distribution" on page 340 but the method of maximum likelihood must be modified. This example shows how you can request a three-parameter lognormal distribution.

A manufacturing process (assumed to be in statistical control) produces a plastic laminate whose strength must exceed a minimum of 25 psi. Samples are tested, and a lognormal distribution is observed for the strengths. It is important to estimate θ to determine whether the process is capable of meeting the strength requirement. The strengths for 49 samples are saved in the following data set:

```
data Plastic;
   label Strength='Strength in psi';
   input Strength @@;
   datalines;
30.26 31.23 71.96 47.39 33.93 76.15 42.21
81.37 78.48 72.65 61.63 34.90 24.83 68.93
43.27 41.76 57.24 23.80 34.03 33.38 21.87
31.29 32.48 51.54 44.06 42.66 47.98 33.73
25.80 29.95 60.89 55.33 39.44 34.50 73.51
43.41 54.67 99.43 50.76 48.81 31.86 33.88
35.57 60.41 54.92 35.66 59.30 41.96 45.32
;
```

The following statements use the LOGNORMAL option in the HISTOGRAM statement to display the fitted three-parameter lognormal curve shown in Output 6.13.1:

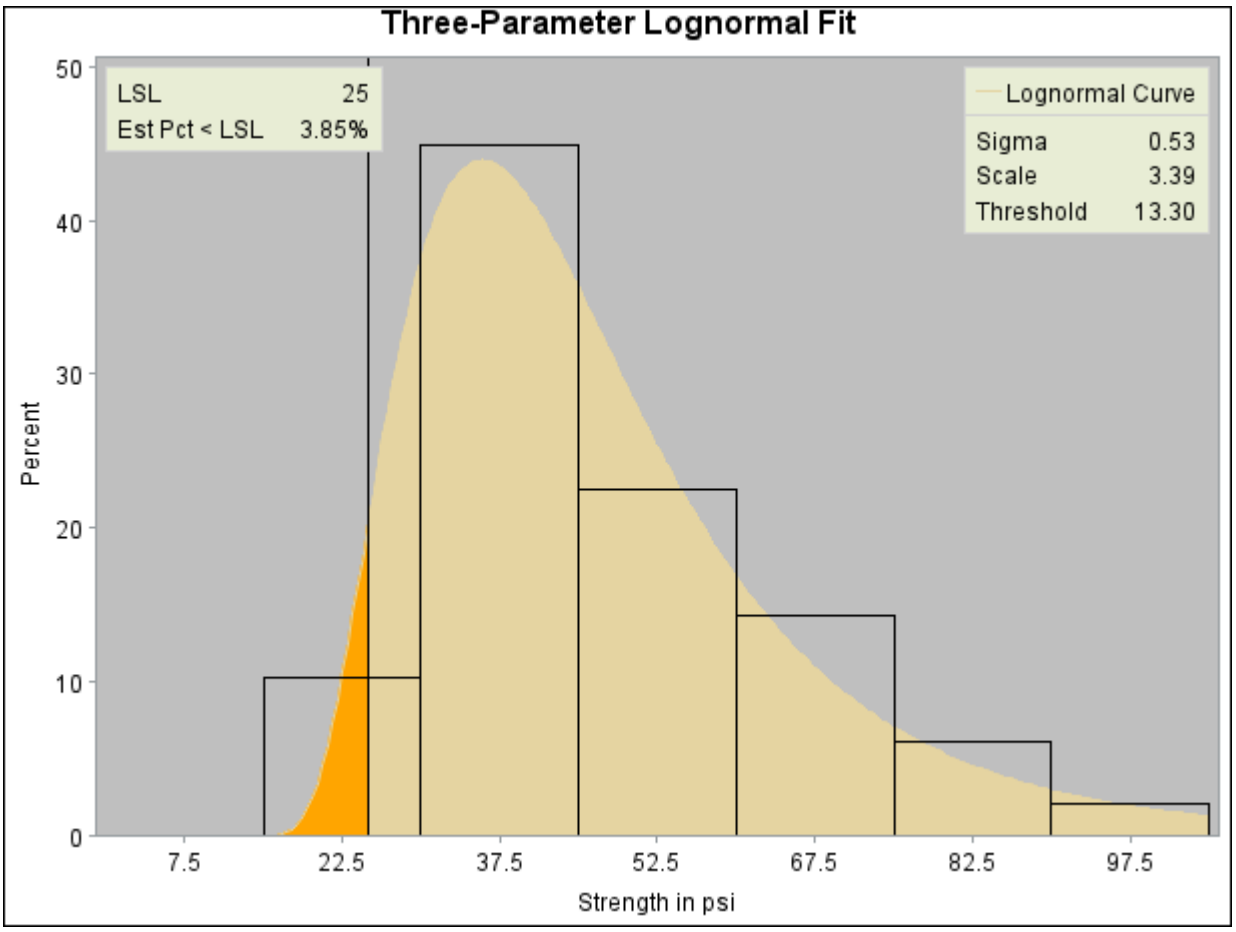
```
ods graphics off;
title 'Three-Parameter Lognormal Fit';
proc capability data=Plastic noprint;
   spec lsl=25 cleft=orange clsl=black;
   histogram Strength / lognormal(fill color = paoy
                                  theta = est)
                        cfill  = paoy
                        cframe = ligr
                        nolegend;
   inset lsl='LSL' lslpct / cfill = ywh pos=nw;
   inset lognormal        / format=6.2 pos=ne cfill = ywh;
run;
```

Specifying THETA=EST requests a *local* maximum likelihood estimate (LMLE) for θ, as described by Cohen (1951). This estimate is then used to compute maximum likelihood estimates for σ and ζ. The sample program CAPL3A illustrates a similar computational method implemented as a SAS/IML program.

NOTE: See *Three-Parameter Weibull Distribution* in the SAS/QC Sample Library.

Note that you can specify THETA=EST as a *Weibull-option* to fit a three-parameter Weibull distribution.

Output 6.13.1 Three-Parameter Lognormal Fit



Example 6.14: Annotating a Folded Normal Curve

NOTE: See *Cpk for Folded Normal Distribution* in the SAS/QC Sample Library.

This example shows how to display a fitted curve that is not supported by the HISTOGRAM statement.

The offset of an attachment point is measured (in mm) for a number of manufactured assemblies, and the measurements are saved in a data set named Assembly.

```
data Assembly;
   label Offset = 'Offset (in mm)';
   input Offset @@;
   datalines;
11.11 13.07 11.42  3.92 11.08  5.40 11.22 14.69  6.27  9.76
 9.18  5.07  3.51 16.65 14.10  9.69 16.61  5.67  2.89  8.13
 9.97  3.28 13.03 13.78  3.13  9.53  4.58  7.94 13.51 11.43
11.98  3.90  7.67  4.32 12.69  6.17 11.48  2.82 20.42  1.01
 3.18  6.02  6.63  1.72  2.42 11.32 16.49  1.22  9.13  3.34
 1.29  1.70  0.65  2.62  2.04 11.08 18.85 11.94  8.34  2.07
 0.31  8.91 13.62 14.94  4.83 16.84  7.09  3.37  0.49 15.19
```

```
 5.16  4.14  1.92 12.70  1.97  2.10  9.38  3.18  4.18  7.22
15.84 10.85  2.35  1.93  9.19  1.39 11.40 12.20 16.07  9.23
 0.05  2.15  1.95  4.39  0.48 10.16  4.81  8.28  5.68 22.81
 0.23  0.38 12.71  0.06 10.11 18.38  5.53  9.36  9.32  3.63
12.93 10.39  2.05 15.49  8.12  9.52  7.77 10.70  6.37  1.91
 8.60 22.22  1.74  5.84 12.90 13.06  5.08  2.09  6.41  1.40
15.60  2.36  3.97  6.17  0.62  8.56  9.36 10.19  7.16  2.37
12.91  0.95  0.89  3.82  7.86  5.33 12.92  2.64  7.92 14.06
;
```

The assembly process is in statistical control, and it is decided to fit a *folded normal distribution* to the offset measurements. A variable X has a folded normal distribution if $X = |Y|$, where Y is distributed as $N(\mu, \sigma)$. The fitted density is

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma}\left[\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x+\mu)^2}{2\sigma^2}\right)\right] \quad , \; x \geq 0$$

You can use SAS/IML software to compute preliminary estimates of μ and σ based on a method of moments given by Elandt (1961). These estimates are computed by solving equation (19) of Elandt (1961), which is given by

$$f(\theta) = \frac{\left(\frac{2}{\sqrt{2\pi}}e^{-\theta^2/2} - \theta\left[1 - 2\Phi(\theta)\right]\right)^2}{1+\theta^2} = A$$

where $\Phi(\cdot)$ is the standard normal distribution function, and

$$A = \frac{\bar{x}^2}{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

Then the estimates of σ and μ are given by

$$\begin{aligned} \hat{\sigma}_0 &= \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n} x_i^2}{1+\hat{\theta}^2}} \\ \hat{\mu}_0 &= \hat{\theta} \cdot \hat{\sigma}_0 \end{aligned}$$

Begin by using the MEANS procedure to compute the first and second moments and using the DATA step to compute the constant A.

```
proc means data = Assembly noprint;
   var Offset;
   output out=Stat mean=m1 var=var n=n min = min;
run;

* Compute constant A from equation (19) of Elandt (1961) ;
data Stat;
   keep m2 a min;
   set Stat;
   a  = (m1*m1);
   m2 = ((n-1)/n)*var + a;
   a  = a/m2;
run;
```

Next, use the SAS/IML subroutine NLPDD to solve equation (19) by minimizing $(f(\theta) - A)^2$, and compute $\hat{\mu}_0$ and $\hat{\sigma}_0$.

```
proc iml;
   use Stat;
   read all var {m2}  into m2;
   read all var {a}   into a;
   read all var {min} into min;

   * f(t) is the function in equation (19) of Elandt (1961) ;
   start f(t) global(a);
      y = .39894*exp(-0.5*t*t);
      y = (2*y-(t*(1-2*probnorm(t))))**2/(1+t*t);
      y = (y-a)**2;
      return(y);
   finish;

   * Minimize (f(t)-A)**2 and estimate mu and sigma ;
   if ( min < 0 ) then do;
      print "Warning: Observations are not all nonnegative.";
      print "     The folded normal is inappropriate.";
      stop;
   end;
   if ( a < 0.637 ) then do;
      print "Warning: the folded normal may be inappropriate";
   end;
   opt = { 0 0 };
   con = { 1e-6 };
   x0  = { 2.0 };
   tc  = { . . . . . 1e-12 . . . . . .};
   call nlpdd(rc,etheta0,"f",x0,opt,con,tc);
   esig0 = sqrt(m2/(1+etheta0*etheta0));
   emu0  = etheta0*esig0;

   create Prelim var {emu0 esig0 etheta0};
   append;
   close Prelim;
      * Define the log likelihood of the folded normal ;
      start g(p) global(x);
         y = 0.0;
         do i = 1 to nrow(x);
            z = exp( (-0.5/p[2])*(x[i]-p[1])*(x[i]-p[1]) );
            z = z + exp( (-0.5/p[2])*(x[i]+p[1])*(x[i]+p[1]) );
            y = y + log(z);
         end;
         y = y - nrow(x)*log( sqrt( p[2] ) );
         return(y);
      finish;
      * Maximize the log likelihood with subroutine NLPDD ;
      use Assembly;
      read all var {Offset} into x;
      esig0sq = esig0*esig0;
      x0      = emu0||esig0sq;
      opt     = { 1 0 };
      con     = { . 0.0, .  .  };
      call nlpdd(rc,xr,"g",x0,opt,con);
```

```
      emu     = xr[1];
      esig    = sqrt(xr[2]);
      etheta  = emu/esig;
      create Parmest var{emu esig etheta};
      append;
      close Parmest;
   quit;

title 'The Data Set Prelim';
proc print data=Prelim noobs;
run;
```

The preliminary estimates are saved in the data set Prelim, as shown in Output 6.14.1.

Output 6.14.1 Preliminary Estimates of μ, σ, and θ

The Data Set Prelim

EMU0	ESIG0	ETHETA0
6.51735	6.54953	0.99509

Now, using $\hat{\mu}_0$ and $\hat{\sigma}_0$ as initial estimates, call the NLPDD subroutine to maximize the log likelihood, $l(\mu, \sigma)$, of the folded normal distribution, where, up to a constant,

$$l(\mu,\sigma) = -n\log\sigma + \sum_{i=1}^{n}\log\left[\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x_i+\mu)^2}{2\sigma^2}\right)\right]$$

```
* Define the log likelihood of the folded normal ;
start g(p) global(x);
   y = 0.0;
   do i = 1 to nrow(x);
      z = exp( (-0.5/p[2])*(x[i]-p[1])*(x[i]-p[1]) );
      z = z + exp( (-0.5/p[2])*(x[i]+p[1])*(x[i]+p[1]) );
      y = y + log(z);
   end;
   y = y - nrow(x)*log( sqrt( p[2] ) );
   return(y);
finish;

* Maximize the log likelihood with subroutine NLPDD ;
use assembly;
read all var {offset} into x;
esig0sq = esig0*esig0;
x0      = emu0||esig0sq;
opt     = { 1 0 };
con     = { . 0.0, .  .  };
call nlpdd(rc,xr,"g",x0,opt,con);
emu     = xr[1];
esig    = sqrt(xr[2]);
etheta  = emu/esig;

create parmest var{emu esig etheta};
append;
```

```
   close parmest;
quit;

title 'The Data Set PARMEST';
proc print data=Parmest noobs;
   var emu esig etheta;
run;
```

The data set Parmest saves the maximum likelihood estimates $\hat{\mu}$ and $\hat{\sigma}$ (as well as $\hat{\mu}/\hat{\sigma}$), as shown in Output 6.14.2.

Output 6.14.2 Final Estimates of μ, σ, and θ

The Data Set PARMEST

EMU	ESIG	ETHETA
6.66761	6.39650	1.04239

To annotate the curve on a histogram, begin by computing the width and endpoints of the histogram intervals. The following statements save these values in an OUTFIT= data set called OUT. Note that a plot is not produced at this point.

```
ods graphics off;
proc capability data = Assembly noprint;
   histogram Offset / outfit = Out normal(noprint) noplot;
run;

title 'OUTFIT= Data Set Out';
proc print data=Out noobs round;
   var _var_ _curve_ _locatn_ _scale_ _chisq_ _df_ _pchisq_
       _midpt1_ _width_ _midptn_ _expect_ _eststd_ _adasq_
       _adp_ _cvmwsq_ _cvmp_ _ksd_ _ksp_;
run;
```

Output 6.14.3 provides a partial listing of the data set Out. The width and endpoints of the histogram bars are saved as values of the variables _WIDTH_, _MIDPT1_, and _MIDPTN_. See "Output Data Sets" on page 356.

Output 6.14.3 The OUTFIT= Data Set Out

OUTFIT= Data Set Out

VAR	_CURVE_	_LOCATN_	_SCALE_	_CHISQ_	_DF_	_PCHISQ_	_MIDPT1_	_WIDTH_	_MIDPTN_
Offset	NORMAL	7.62	5.24	31.17	5	0	1.5	3	22.5

EXPECT	_ESTSTD_	_ADASQ_	_ADP_	_CVMWSQ_	_CVMP_	_KSD_	_KSP_
7.62	5.24	1.9	0.01	0.28	0.01	0.09	0.01

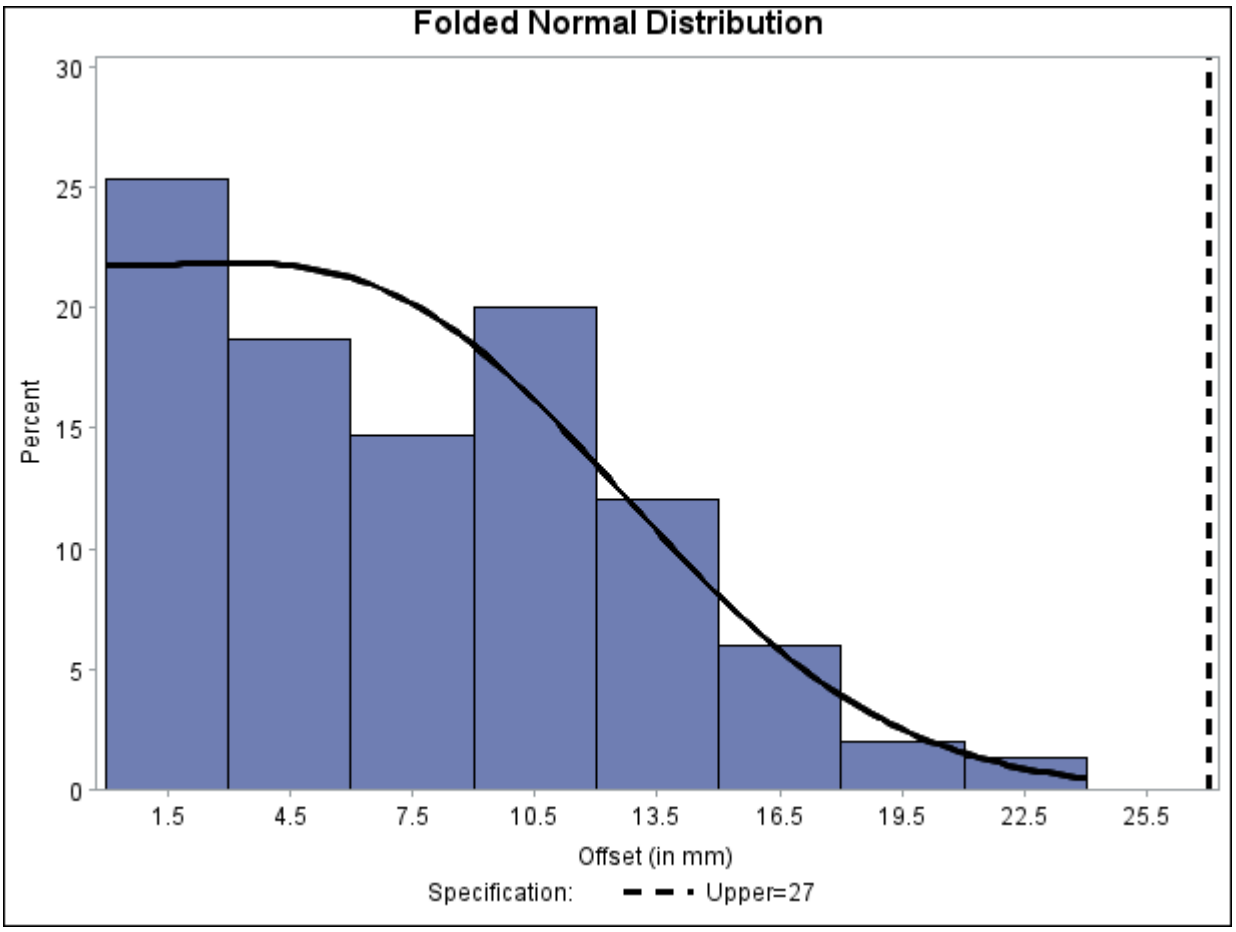
The following statements create an annotate data set named Anno, which contains the coordinates of the fitted curve:

```
data Anno;
   merge Parmest Out;
   length function color $ 8;

   function = 'point';
   color    = 'black';
   size     =  2;
   xsys     = '2';
   ysys     = '2';
   when     = 'a';
   constant = 39.894*_width_;;
   left     =  _midpt1_ - .5*_width_;
   right    =  _midptn_ + .5*_width_;
   inc      = (right-left)/100;
   do x = left to right by inc;
      z1 = (x-emu)/esig;
      z2 = (x+emu)/esig;
      y  = (constant/esig)*(exp(-0.5*z1*z1)+exp(-0.5*z2*z2));
      output;
      function = 'draw';
   end;
run;
```

The following statements read the ANNOTATE= data set and display the histogram and fitted curve, as shown in Output 6.14.4:

```
ods graphics off;
title "Folded Normal Distribution";
proc capability data=Assembly noprint;
   spec usl=27 cusl=black lusl=2 wusl=2;
   histogram Offset / annotate = Anno;
run;
```

Output 6.14.4 Histogram with Annotated Folded Normal Curve



INSET Statement: CAPABILITY Procedure

Overview: INSET Statement

Graphical displays such as histograms and probability plots are commonly used for process capability analysis. You can use the INSET statement to enhance these plots by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. An inset typically displays statistics calculated by the CAPABILITY procedure but can also display values provided in a SAS data set. A typical application of the INSET statement is to augment a histogram with the sample size, mean, standard deviation, and process capability index C_{pk}.

Note that the INSET statement by itself does not produce a display and must be used with the CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, or QQPLOT statement.

You can use options in the INSET statement to

- specify the position of the inset

- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

The INSET statement is not applicable when you produce line printer plots by specifying the LINEPRINTER option in the PROC CAPABILITY statement.

Getting Started: INSET Statement

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the section "Syntax: INSET Statement" on page 390, and advanced examples are given in the section "Examples: INSET Statement" on page 410.

Displaying Summary Statistics on a Histogram

NOTE: See *Histograms with INSET Statement Features* in the SAS/QC Sample Library.

In a plant producing copper wire, an important quality characteristic is the torsion strength, measured as the twisting force in pounds per inch necessary to break the wire. The following statements create the SAS data set Wire, which contains the torsion strengths (Strength) for 50 different wire samples:

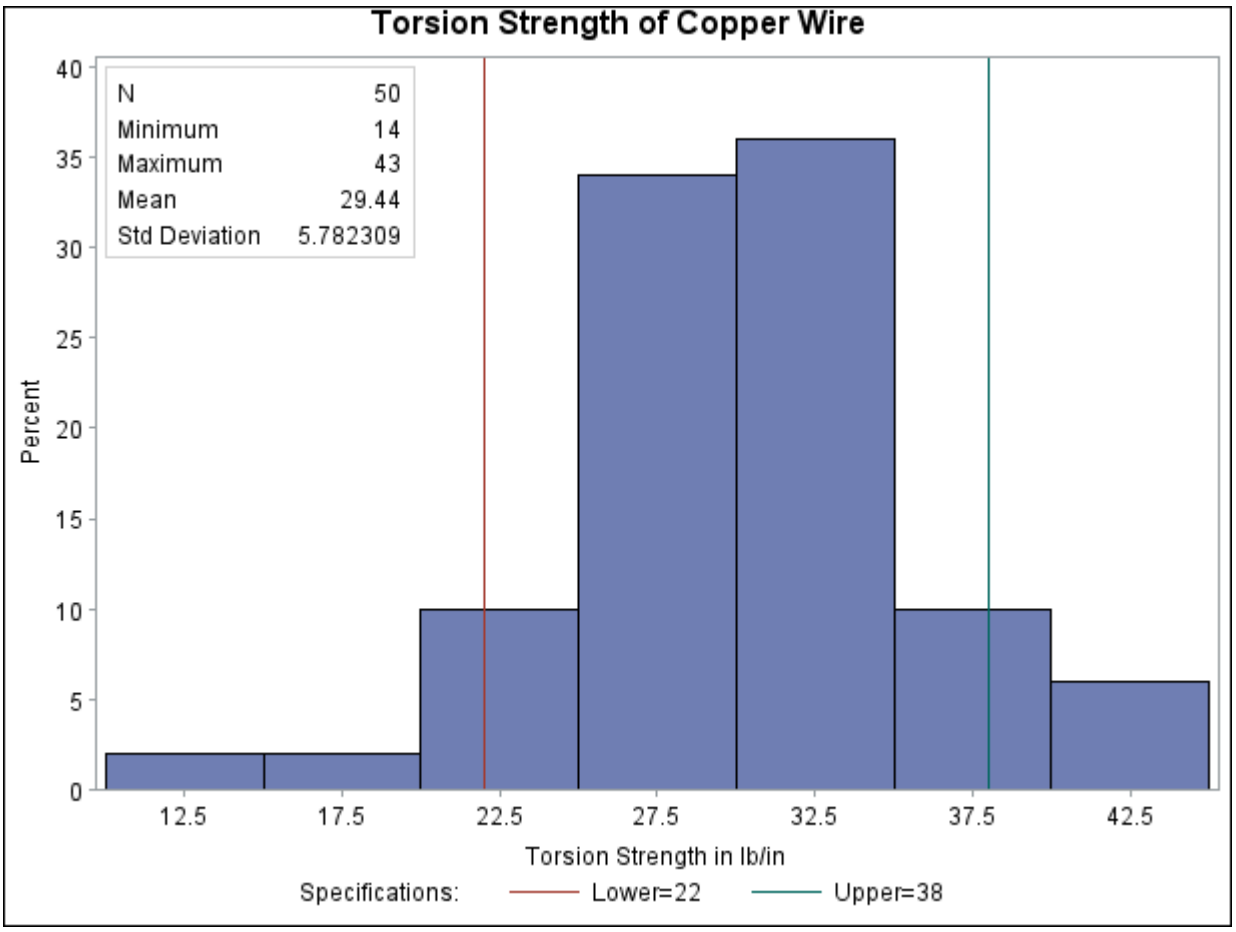
```
data Wire;
   label Strength='Torsion Strength in lb/in';
   input Strength @@;
   datalines;
25 25 36 31 26 36 29 37 37 20
34 27 21 35 30 41 33 21 26 26
19 25 14 32 30 29 31 26 22 24
34 33 28 26 43 30 40 32 32 31
25 26 27 34 33 27 33 29 30 31
;
```

A histogram is used to examine the data distribution. For a more complete report, the sample size, minimum value, maximum value, mean, and standard deviation are displayed on the histogram. The following statements illustrate how to inset these statistics:

```
ods graphics off;
title 'Torsion Strength of Copper Wire';
proc capability data=Wire noprint;
   spec lsl=22 usl=38;
   histogram Strength;
   inset n min max mean std;
run;
```

The resulting histogram is displayed in Figure 6.17. The INSET statement immediately follows the plot statement that creates the graphical display (in this case, the HISTOGRAM statement). Specify the keywords for inset statistics (such as N, MIN, MAX, MEAN, and STD) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords.

Figure 6.17 A Histogram with an Inset



A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 392. Note that the set of keywords available for a particular display depends on both the plot statement that precedes the INSET statement and the options that you specify in the plot statement.

The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

NOTE: See *Histograms with INSET Statement Features* in the SAS/QC Sample Library.

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 6.17 the default format for the standard deviation prints an excessive number of decimal places. The following statements correct this problem, as well as customizing some of the labels displayed in the inset:

```
ods graphics on;
proc capability data=Wire noprint;
   spec lsl=22 usl=38;
   histogram Strength;
   inset n='Sample Size' min max mean std='Std Dev' (5.2);
run;
```

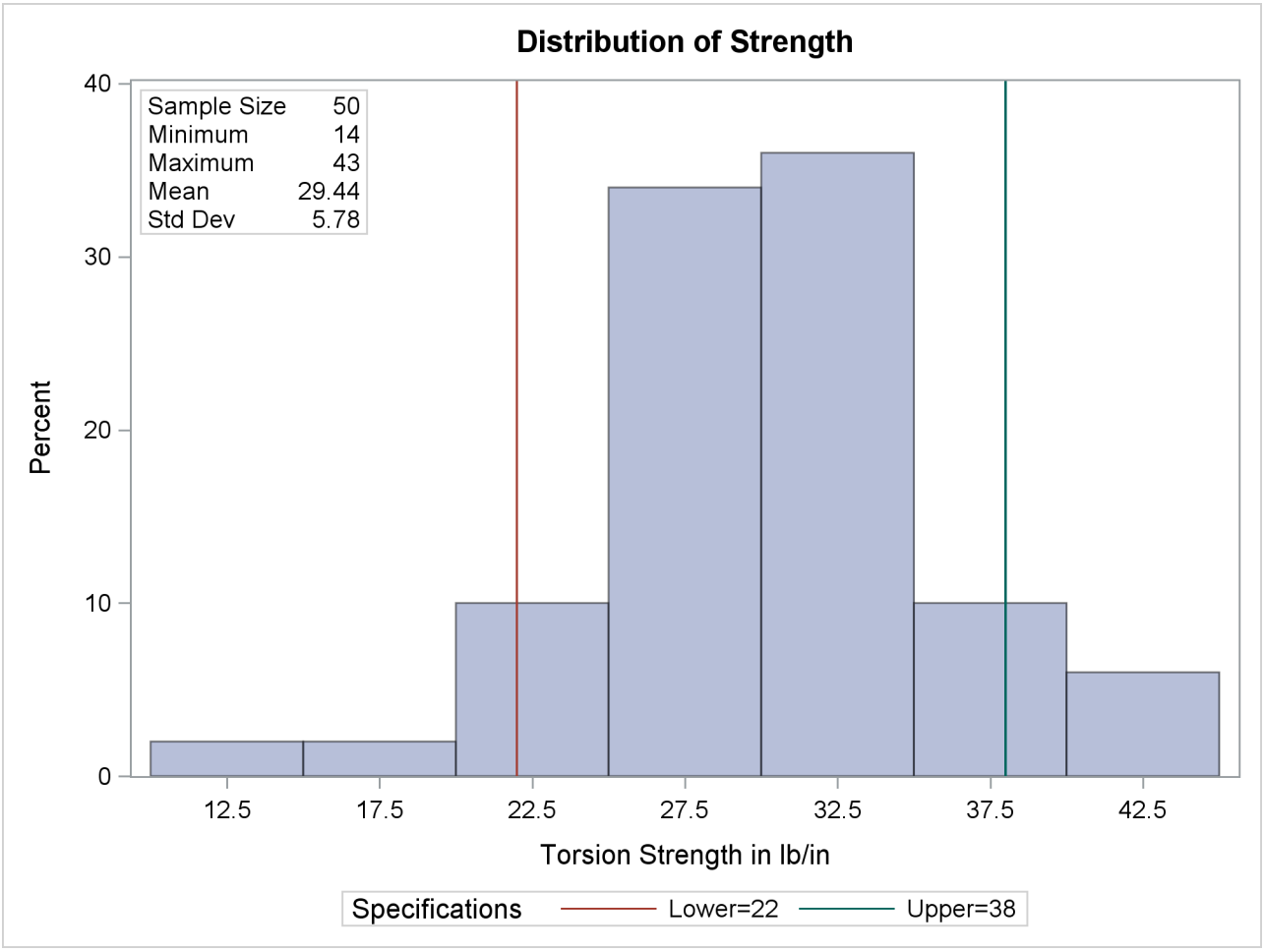
The ODS GRAPHICS ON statement specified before the PROC CAPABILITY statement enables ODS Graphics, so the histogram is created using ODS Graphics instead of traditional graphics.

The resulting histogram is displayed in Figure 6.18. You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 5.2 specified in parentheses after the keyword STD displays the standard deviation with a field width of five and two decimal places. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option (see the next section, "Adding a Header and Positioning the Inset" on page 389). For more information about SAS formats, refer to *SAS Formats and Informats: Reference*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format, as with the keyword STD in the previous statements.

Figure 6.18 Formatting Values and Customizing Labels in an Inset



Adding a Header and Positioning the Inset

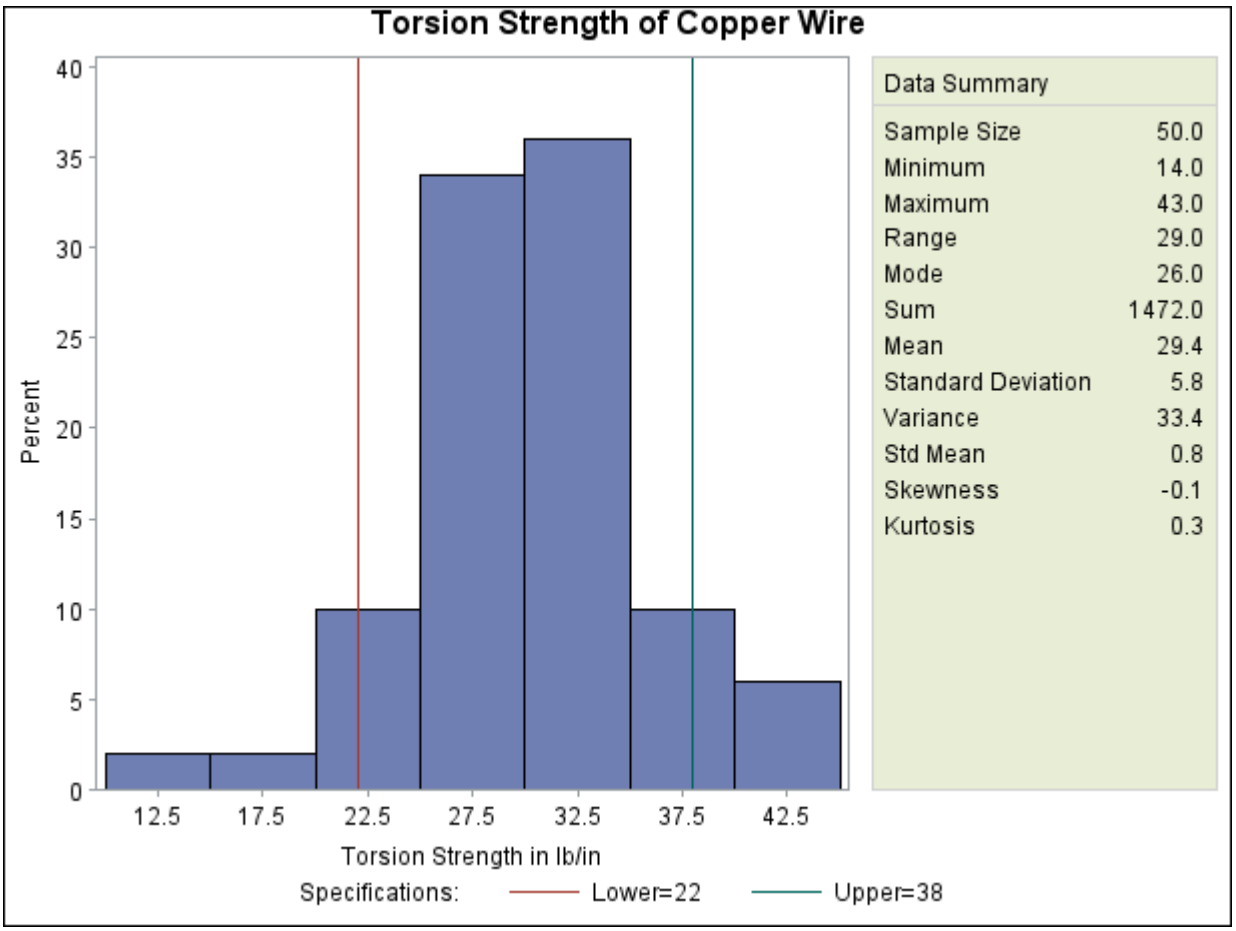
NOTE: See *Histograms with INSET Statement Features* in the SAS/QC Sample Library.

In the previous examples, the inset is displayed in the upper left corner of the plot, the default position for insets added to histograms. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create the chart shown in Figure 6.19:

```
ods graphics off;
title 'Torsion Strength of Copper Wire';
proc capability data=Wire noprint;
   spec lsl=22 usl=38;
   histogram Strength;
   inset n='Sample Size' min max range mode sum mean
      std='Standard Deviation' var stdmean skewness
      kurtosis / format = 6.1
                 pos    = rm
                 header = 'Data Summary' cfill = ywh;
run;
```

The header (in this case, *Data Summary*) can be up to 40 characters. Note that a long list of inset statistics is requested. Consequently, POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details: INSET Statement" on page 405. Also note that the FORMAT= option is used to format all inset statistics. The options, such as HEADER=, POSITION=, and FORMAT=, are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 402.

Figure 6.19 Adding a Header and Repositioning the Inset



Syntax: INSET Statement

The syntax for the INSET statement is as follows:

INSET *keyword-list* < / *options* > **;**

You can use any number of INSET statements in the CAPABILITY procedure. Each INSET statement produces an inset and must follow one of the plot statements: CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, or QQPLOT. The inset appears in all displays produced by the plot statement that immediately precedes it. The statistics are displayed in the order in which they are specified. For example, the following statements produce a cumulative distribution plot with two insets and a histogram with one inset:

```
proc capability data=Wire;
   cdfplot Strength;
      inset mean std min max n;
      inset p1 p5 p10;
   histogram Strength;
      inset var skewness kurtosis;
run;
```

The statistics displayed in an inset are computed for a specific process variable from observations for the current BY group. For example, in the following statements, there are two process variables (Strength and Diameter) and a BY variable (Batch). If there are three different batches (levels of Batch), then a total of six histograms are produced. The statistics in each inset are computed for a particular variable and batch. The labels in the inset are the same for each histogram.

```
proc capability data=Wire2;
   by Batch;
   histogram Strength Diameter / normal;
   inset mean std min max normal(mu sigma);
run;
```

The components of the INSET statement are described as follows.

keyword-list

can include any of the keywords listed in "Summary of INSET Keywords" on page 392. Some keywords allow *secondary keywords* to be specified in parentheses immediately after the *primary keyword*. Also, some inset statistics are available only if you request plot statements and options for which those statistics are calculated. For example, consider the following statements:

```
proc capability data=Wire;
   histogram Strength / normal;
   inset mean std normal(ad adpval);
run;
```

The keywords MEAN and STD display the sample mean and standard deviation of Strength. The primary keyword NORMAL with the secondary keywords AD and ADPVAL display the Anderson-Darling goodness-of-fit test statistic and p-value in the inset as well. The statistics specified with the NORMAL keyword are available only because a normal distribution has been fit to the data by using the NORMAL option in the HISTOGRAM statement. See the section "Summary of INSET Keywords" for a list of available keywords.

Typically, you specify keywords, to display statistics computed by the CAPABILITY procedure. However, you can also specify the keyword DATA= followed by the name of a SAS data set to display customized statistics. This data set must contain two variables:

- a character variable named _LABEL_ whose values provide labels for inset entries.
- a variable named _VALUE_, which can be either character or numeric, and whose values provide values for inset entries.

The label and value from each observation in the DATA= data set occupy one line in the inset. The position of the DATA= keyword in the keyword list determines the position of its lines in the inset.

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the keyword. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 387.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance options (POSITION= and CTEXT=):

```
inset mean std min max / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 402 for a list of all available options, and "Dictionary of Options" on page 402 for detailed descriptions. Note the difference between keywords and options; keywords specify the information to be displayed in an inset, whereas options control the appearance of the inset.

Summary of INSET Keywords

Summary Statistics and Process Capability Indices

Table 6.32 Summary Statistics

Keyword	Description
CSS	Corrected sum of squares
CV	Coefficient of variation
GEOMEAN	Geometric mean
KURTOSIS | KURT	Kurtosis
MAX	Largest value
MEAN	Sample mean
MIN	Smallest value
MODE	Most frequent value
N	Sample size
NEXCL	Number of observations excluded by MAXNBIN= or MAXSIGMAS= option
NMISS	Number of missing values
NOBS	Number of observations
RANGE	Range
SKEWNESS | SKEW	Skewness
STD | STDDEV	Standard deviation
STDMEAN | STDERR	Standard error of the mean
SUM	Sum of the observations
SUMWGT	Sum of the weights
USS	Uncorrected sum of squares
VAR	Variance

Table 6.33 Percentile Statistics

Keyword	Description
P1	1st percentile
P5	5th percentile
P10	10th percentile
Q1 | P25	Lower quartile (25th percentile)
MEDIAN | Q2 | P50	Median (50th percentile)
Q3 | P75	Upper quartile (75th percentile)
P90	90th percentile
P95	95th percentile
P99	99th percentile
QRANGE	Interquartile range (Q3 - Q1)

Table 6.34 lists keywords for distribution-free confidence limits for percentiles requested with the CIPCTLDF option.

Table 6.34 Keywords for Distribution-Free Confidence Limits for Percentiles

Keyword	Description
P1_LCL_DF	1st percentile lower confidence limit
P1_UCL_DF	1st percentile upper confidence limit
P5_LCL_DF	5th percentile lower confidence limit
P5_UCL_DF	5th percentile upper confidence limit
P10_LCL_DF	10th percentile lower confidence limit
P10_UCL_DF	10th percentile upper confidence limit
Q1_LCL_DF | P25_LCL_DF	Lower quartile (25th percentile) lower confidence limit
Q1_UCL_DF | P25_UCL_DF	Lower quartile (25th percentile) upper confidence limit
MEDIAN_LCL_DF | Q2_LCL_DF | P50_LCL_DF	Median (50th percentile) lower confidence limit
MEDIAN_UCL_DF | Q2_UCL_DF | P50_UCL_DF	Median (50th percentile) upper confidence limit
Q3_LCL_DF | P75_LCL_DF	Upper quartile (75th percentile) lower confidence limit
Q3_UCL_DF | P75_UCL_DF	Upper quartile (75th percentile) upper confidence limit
P90_LCL_DF	90th percentile lower confidence limit
P90_UCL_DF	90th percentile upper confidence limit
P95_LCL_DF	95th percentile lower confidence limit
P95_UCL_DF	95th percentile upper confidence limit
P99_LCL_DF	99th percentile lower confidence limit
P99_UCL_DF	99th percentile upper confidence limit

Table 6.35 lists keywords for percentile confidence limits computed assuming normality requested with the CIPCTLNORMAL option.

Table 6.35 Keywords Percentile Confidence Limits Assuming Normality

Keyword	Description
P1_LCL	1st percentile lower confidence limit
P1_UCL	1st percentile upper confidence limit
P5_LCL	5th percentile lower confidence limit
P5_UCL	5th percentile upper confidence limit
P10_LCL	10th percentile lower confidence limit
P10_UCL	10th percentile upper confidence limit
Q1_LCL | P25_LCL	Lower quartile (25th percentile) lower confidence limit
Q1_UCL | P25_UCL	Lower quartile (25th percentile) upper confidence limit
MEDIAN_LCL | Q2_LCL | P50_LCL	Median (50th percentile) lower confidence limit
MEDIAN_UCL | Q2_UCL | P50_UCL	Median (50th percentile) upper confidence limit
Q3_LCL | P75_LCL	Upper quartile (75th percentile) lower confidence limit
Q3_UCL | P75_UCL	Upper quartile (75th percentile) upper confidence limit
P90_LCL	90th percentile lower confidence limit
P90_UCL	90th percentile upper confidence limit
P95_LCL	95th percentile lower confidence limit
P95_UCL	95th percentile upper confidence limit
P99_LCL	99th percentile lower confidence limit
P99_UCL	99th percentile upper confidence limit

Table 6.36 Robust Statistics

Keyword	Description
GINI	Gini's mean difference
MAD	Median absolute difference about the median
QN	Q_n, alternative to MAD
SN	S_n, alternative to MAD
STD_GINI	Gini's standard deviation
STD_MAD	MAD standard deviation
STD_QN	Q_n standard deviation
STD_QRANGE	Interquartile range standard deviation
STD_SN	S_n standard deviation

Table 6.37 Hypothesis Testing

Keyword	Description
MSIGN	Sign statistic
NORMALTEST	Test statistic for normality
PNORMAL	Probability value for the test of normality
SIGNRANK	Signed rank statistic
PROBM	Probability of greater absolute value for the sign statistic
PROBN	Probability value for the test of normality
PROBS	Probability value for the signed rank test
PROBT	Probability value for the Student's *t* test
T	Statistics for Student's *t* test

Table 6.38 Input Data Set

Keyword	Description
DATA=	(label, value) pairs from input data set

Table 6.39 Capability Indices and Confidence Limits

Keyword	Description
CP	Capability index C_p
CPLCL	Lower confidence limit for C_p
CPUCL	Upper confidence limit for C_p
CPK	Capability index C_{pk}
CPKLCL	Lower confidence limit for C_{pk}
CPKUCL	Upper confidence limit for C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPMLCL	Lower confidence limit for C_{pm}
CPMUCL	Upper confidence interval for C_{pm}
CPU	Capability index CPU
K	Capability index K

Table 6.40 Specification Limits and Related Information

Keyword	Description
LSL	Lower specification limit
USL	Upper specification limit
TARGET	Target value
PCTGTR	Percent of nonmissing observations that exceed the upper specification limit
PCTLSS	Percent of nonmissing observations that are less than the lower specification limit
PCTBET	Percent of nonmissing observations between the upper and lower specification limits (inclusive)

Statistics Available with Parametric Density Estimates

You can request parametric density estimates with all plot statements in the CAPABILITY procedure (CDFPLOT, COMPHISTOGRAM, HISTOGRAM, PPPLOT, PROBPLOT, and QQPLOT). You can display parameters and statistics associated with these estimates in an inset by specifying a distribution keyword followed by secondary keywords in parentheses. For example, the following statements create a histogram for Strength with a fitted exponential density curve:

```
proc capability data=Wire;
   histogram Strength / exp;
   inset exp(sigma theta);
run;
```

The secondary keywords SIGMA and THETA for the EXP distribution keyword request an inset displaying the values of the exponential scale parameter σ and threshold parameter θ. You must request the distribution option in the plot statement to display the corresponding distribution statistics in an inset. Specifying a distribution keyword with no secondary keywords produces an inset displaying the full set of parameters for that distribution. See Output 6.15.1 for an example of an inset with statistics from a fitted normal curve.

The following table describes the available distribution keywords. Note that some keywords are not available with all plot statements.

Table 6.41 Density Estimation Primary Keywords

Keyword	Distribution	Plot Statement Availability
BETA	beta	all but COMPHISTOGRAM
EXPONENTIAL	exponential	all but COMPHISTOGRAM
GAMMA	gamma	all but COMPHISTOGRAM
GUMBEL	Gumbel	all but COMPHISTOGRAM
IGAUSS	inverse Gaussian	CDFPLOT, HISTOGRAM, PPPLOT
LOGNORMAL	lognormal	all but COMPHISTOGRAM
NORMAL	normal	all
PARETO	generalized Pareto	all but COMPHISTOGRAM
POWER	power function	all but COMPHISTOGRAM
RAYLEIGH	Rayleigh	all but COMPHISTOGRAM
SB	Johnson S_B	HISTOGRAM
SU	Johnson S_U	HISTOGRAM
WEIBULL	Weibull	all but COMPHISTOGRAM
WEIBULL2	2-parameter Weibull	PROBPLOT, QQPLOT

Table 6.42 lists the secondary keywords available with each distribution keyword listed in Table 6.41. In many cases, aliases can be used (for example, ALPHA in place of SHAPE1).

Table 6.42 Density Estimation Secondary Keywords

Secondary Keyword	Alias	Description
Secondary Keywords Available with the BETA Keyword		
ALPHA	SHAPE1	First shape parameter α
BETA	SHAPE2	Second shape parameter β
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Lower threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the EXPONENTIAL Keyword		
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the GAMMA Keyword		
ALPHA	SHAPE	Shape parameter α
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the GUMBEL Keyword		
MU		Location parameter μ
SIGMA	SCALE	Scale parameter σ

Table 6.42 (*continued*)

Secondary Keyword	Alias	Description
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the IGAUSS Keyword		
MU		Mean parameter μ
LAMBDA		Shape parameter λ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the LOGNORMAL Keyword		
SIGMA	SHAPE	Shape parameter σ
THETA	THRESHOLD	Threshold parameter θ
ZETA	SCALE	Scale parameter ζ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the NORMAL Keyword		
MU	MEAN	Mean parameter μ
SIGMA	STD	Scale parameter σ
Secondary Keywords Available with the PARETO Keyword		
ALPHA		Shape parameter α
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the POWER Keyword		
ALPHA		Shape parameter α
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the RAYLEIGH Keyword		
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the SB Keyword		
DELTA	SHAPE1	Shape parameter δ
GAMMA	SHAPE2	Shape parameter γ
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the SU Keyword		
DELTA	SHAPE1	Shape parameter δ
GAMMA	SHAPE2	Shape parameter γ

Table 6.42 (*continued*)

Secondary Keyword	Alias	Description
SIGMA	SCALE	Scale parameter σ
THETA		Location parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the WEIBULL Keyword		
C	SHAPE	Shape parameter c
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Threshold parameter θ
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution
Secondary Keywords Available with the WEIBULL2 Keyword		
C	SHAPE	Shape parameter c
SIGMA	SCALE	Scale parameter σ
THETA	THRESHOLD	Known lower threshold θ_0
MEAN		Mean of the fitted distribution
STD		Standard deviation of the fitted distribution

The secondary keywords listed in Table 6.43 can be used with any distribution keyword but *only* with the HISTOGRAM and COMPHISTOGRAM plot statements.

Table 6.43 Statistics Computed from Any Parametric Density Estimate

Secondary Keyword	Description
CP	Capability index C_p
CPK	Capability index C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
ESTPCTLSS	Estimated percentage less than the lower specification limit
ESTPCTGTR	Estimated percentage greater than the upper specification limit
K	Capability index K

The secondary keywords listed in Table 6.44 can be used with any distribution keyword but *only* with the HISTOGRAM plot statement (see Example 6.15).

Table 6.44 Goodness-of-Fit Statistics for Fitted Curves

Secondary Keyword	Description
CHISQ	Chi-square statistic
DF	Degrees of freedom for the chi-square test
PCHISQ	Probability value for the chi-square test
AD	Anderson-Darling EDF test statistic
ADPVAL	Anderson-Darling EDF test *p*-value
CVM	Cramér-von Mises EDF test statistic
CVMPVAL	Cramér-von Mises EDF test *p*-value
KSD	Kolmogorov-Smirnov EDF test statistic
KSDPVAL	Kolmogorov-Smirnov EDF test *p*-value

Table 6.45 lists primary keywords available only with the HISTOGRAM and COMPHISTOGRAM plot statements. These keywords display fill areas on a histogram. If you fit a parametric density on a histogram and request that the area under the curve be filled, these keywords display the percentage of the distribution area that lies below the lower specification limit, between the specification limits, or above the upper specification limit. If you do not fill the area beneath a parametric density estimate, these keywords display the observed proportion of observations (that is, the area in the bars of the histogram).

You should use these options with the FILL, CFILL=, and PFILL= options in the HISTOGRAM and COMPHISTOGRAM statements and with the CLEFT=, CRIGHT=, PLEFT=, and PRIGHT= options in the SPEC statements. See Output 6.16.1 for an example.

Table 6.45 Curve Area Keywords

Keyword	Alias	Description
BETWEENPCT	BETPCT	Area between the specification limits
LSLPCT		Area below the lower specification limit
USLPCT		Area above the upper specification limit

Statistics Available with Nonparametric Kernel Density Estimates

You can request nonparametric kernel density estimates with the HISTOGRAM and COMPHISTOGRAM plot statements. You can display statistics associated with these estimates by specifying a kernel density keyword followed by secondary keywords in parentheses. For example, the following statements create a histogram for Strength with a fitted kernel density estimate:

```
proc capability data=Wire;
   histogram Strength / kernel;
   inset kernel(c amise);
run;
```

The secondary keywords C and AMISE for the KERNEL keyword display the values of the standardized bandwidth c and the approximate mean integrated square error.

Note that you can specify more than one kernel density estimate on a single histogram. If you specify multiple kernel density estimates, you can request inset statistics for all of the estimates with the KERNEL keyword, or you can display inset statistics for up to five individual curves with KERNEL*n* keywords, as in the following example:

```
proc capability data=Wire;
   histogram Strength / kernel(c = 1 2 3);
   inset kernel2(c) kernel3(c);
run;
```

Three kernel density estimates are displayed on the histogram, but the inset displays the value of c only for the second and third estimates.

Table 6.46 lists the kernel density keywords. Table 6.47 lists the available secondary keywords.

Table 6.46 Kernel Density Estimate Primary Keywords

Keyword	Description
KERNEL	displays statistics for all kernel estimates
KERNEL*n*	displays statistics for only the *n*th kernel density estimate $n = 1, 2, 3, 4,$ or 5

Table 6.47 Secondary Keywords Available with the KERNEL Keyword

Secondary Keyword	Description
TYPE	kernel type: normal, quadratic, or triangular
BANDWIDTH	bandwidth λ for the density estimate
BWIDTH	alias for BANDWIDTH
C	standardized bandwidth c for the density estimate: $c = \frac{\lambda}{Q} n^{\frac{1}{5}}$ where n = sample size, λ = bandwidth, and Q = interquartile range
AMISE	approximate mean integrated square error (MISE) for the kernel density

Summary of Options

The following table lists the INSET statement options. See the section "Dictionary of Options" for complete descriptions of the options.

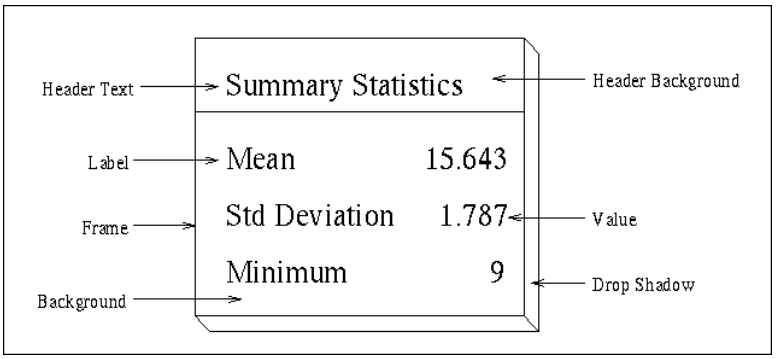
Table 6.48 INSET Options

Option	Description
CFILL=	specifies color of inset background
CFILLH=	specifies color of header background
CFRAME=	specifies color of frame
CHEADER=	specifies color of header text
CSHADOW=	specifies color of drop shadow
CTEXT=	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=	specifies font of text
FORMAT=	specifies format of values in inset
GUTTER=	specifies gutter width for inset in top or bottom margin
HEADER=	specifies header text
HEIGHT=	specifies height of inset text
NCOLS=	specifies number of columns for inset in top or bottom margin
NOFRAME	suppresses frame around inset
POSITION=	specifies position of inset
REFPOINT=	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following sections provide detailed descriptions of options for the INSET statement. Terms used in this section are illustrated in Figure 6.20.

Figure 6.20 The Inset



General Options

You can specify the following general options:

DATA

specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 407. See Figure 6.23 for an example.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option. See Figure 6.19 or Output 6.15.1 for an example.

GUTTER=*value*

specifies the gutter width in percent screen units for an inset located in the top or bottom margin of ODS Graphics output. The gutter is the space between columns of (label, value) pairs in an inset. The default value is four. This option is ignored if ODS Graphics is disabled.

HEADER= '*string*'

specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset. If all the keywords listed in the INSET statement are secondary keywords corresponding to a fitted curve on a histogram, a default header is displayed that indicates the distribution and identifies the curve. See Figure 6.19 for an example of a specified header and Output 6.15.1 for an example of the default header for a fitted normal curve.

NCOLS=*n*

specifies the number of columns of (label, value) pairs displayed in an inset located in the top or bottom margin of ODS Graphics output. The default value is three. This option is ignored if ODS Graphics is disabled.

NOFRAME

suppresses the frame drawn around the text.

POSITION=*position*

POS=*position*

determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details: INSET Statement" on page 405. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

NOTE: In this release of the CAPABILITY procedure, you cannot specify coordinates with the POSITION= option when producing ODS Graphics output.

Options for Traditional Graphics

You can specify the following options if you are producing traditional graphics:

CFILL=*color* | BLANK

specifies the color of the background (including the header background if you do not specify the CFILLH= option). See Output 6.15.1 for an example.

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as curves, histogram bars, or specification limits) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*

CS=*color*
specifies the color of the drop shadow. See Output 6.16.1 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*

CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

FONT=*font*
specifies the font of the text. By default, the font is SIMPLEX if the inset is located in the interior of the plot, and the font is the same as the other text displayed on the plot if the inset is located in the exterior of the plot.

HEIGHT=*value*
specifies the height of the text.

REFPOINT=BR | BL | TR | TL

RP=BR | BL | TR | TL
specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 6.24 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 407.

Details: INSET Statement

This section provides details on three different methods of positioning the inset with the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

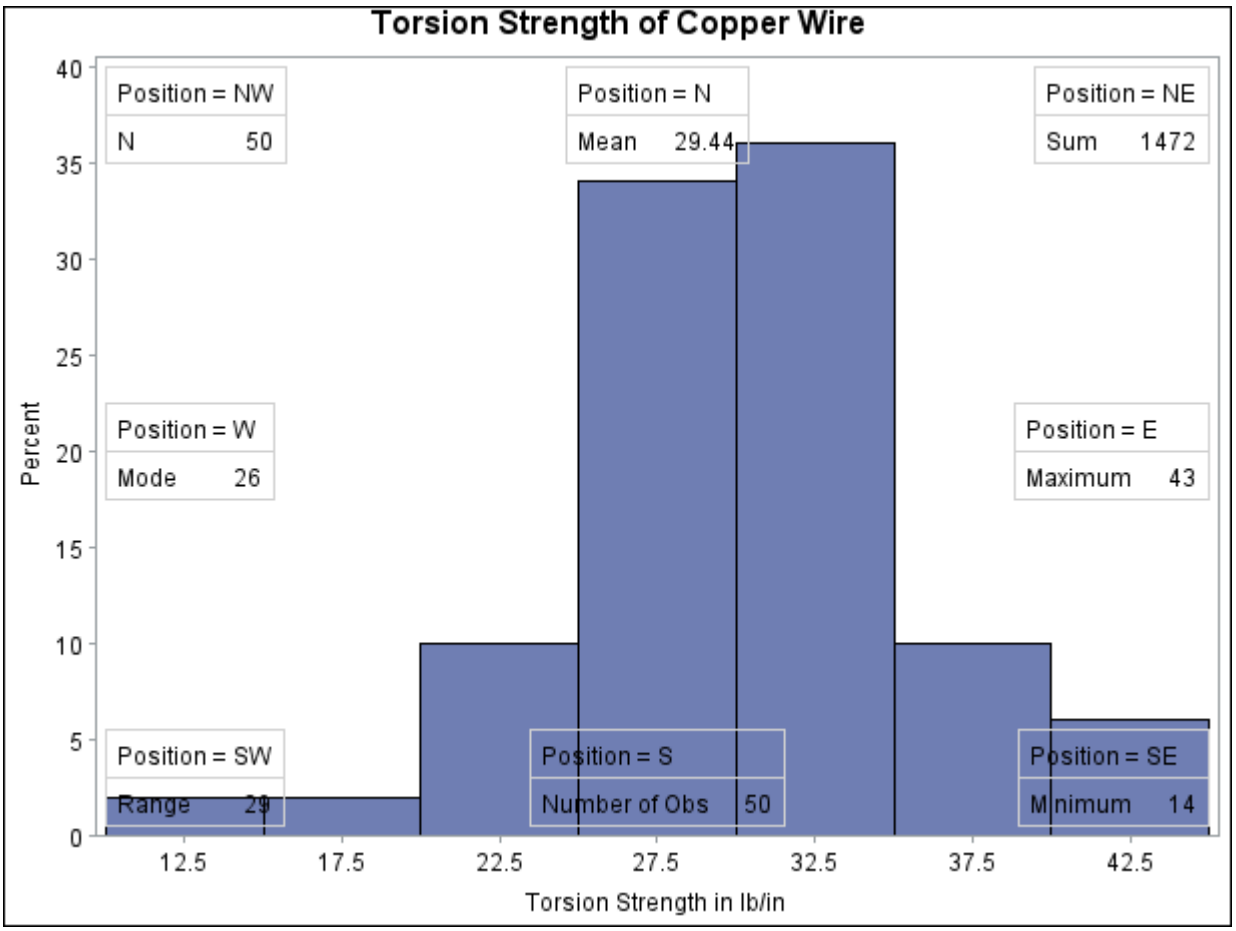
Positioning the Inset Using Compass Points

NOTE: See *Positioning the Inset* in the SAS/QC Sample Library.

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 6.21, which demonstrates all eight compass positions. The default is NW.

```
ods graphics off;
title 'Torsion Strength of Copper Wire';
proc capability data=Wire;
   histogram Strength / odstitle = title;
   inset n     / header='Position = NW' pos=nw;
   inset mean  / header='Position = N ' pos=n ;
   inset sum   / header='Position = NE' pos=ne;
   inset max   / header='Position = E ' pos=e ;
   inset min   / header='Position = SE' pos=se;
   inset nobs  / header='Position = S ' pos=s ;
   inset range / header='Position = SW' pos=sw;
   inset mode  / header='Position = W ' pos=w ;
run;
```

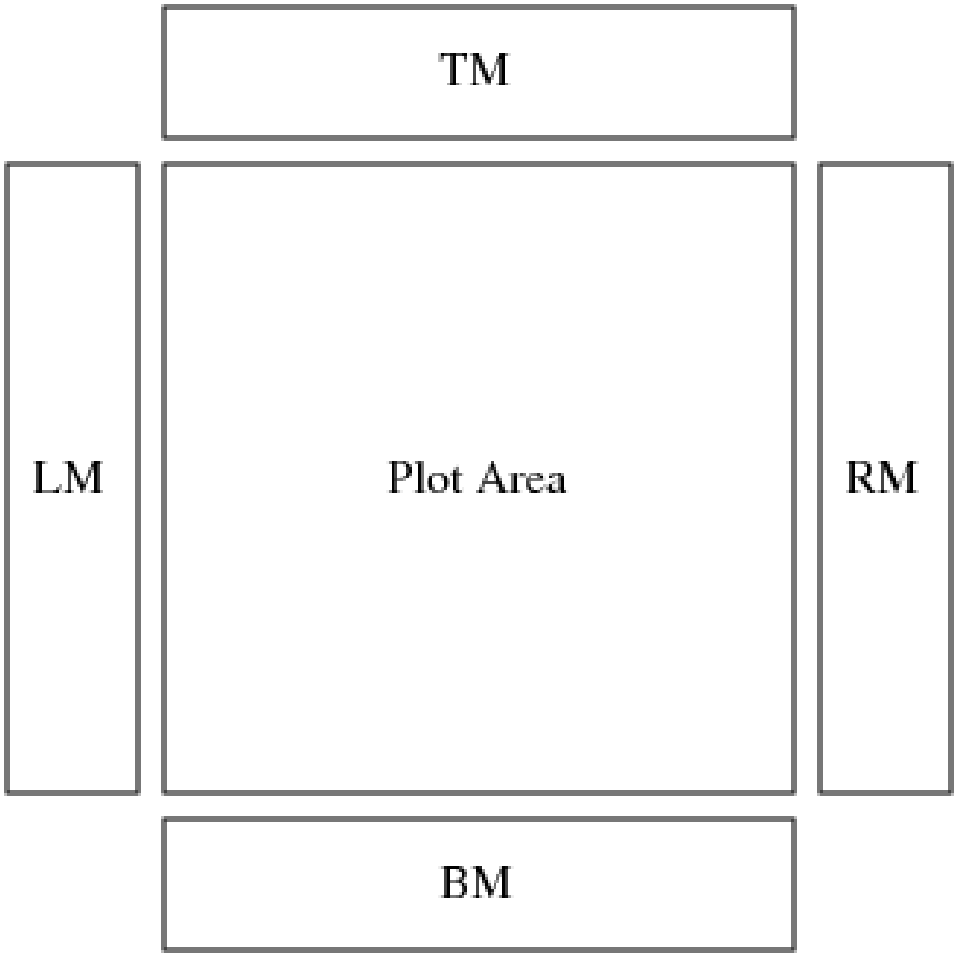
Figure 6.21 Insets Positioned Using Compass Points



Positioning the Inset in the Margins

You can also position the inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 6.22.

Figure 6.22 Positioning Insets in the Margins



For an example of an inset placed in the right margin, see Figure 6.19. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

If you are producing traditional graphics, you can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

NOTE: In this release of the CAPABILITY procedure, you cannot position insets by using coordinates when producing ODS Graphics output.

Data Unit Coordinates

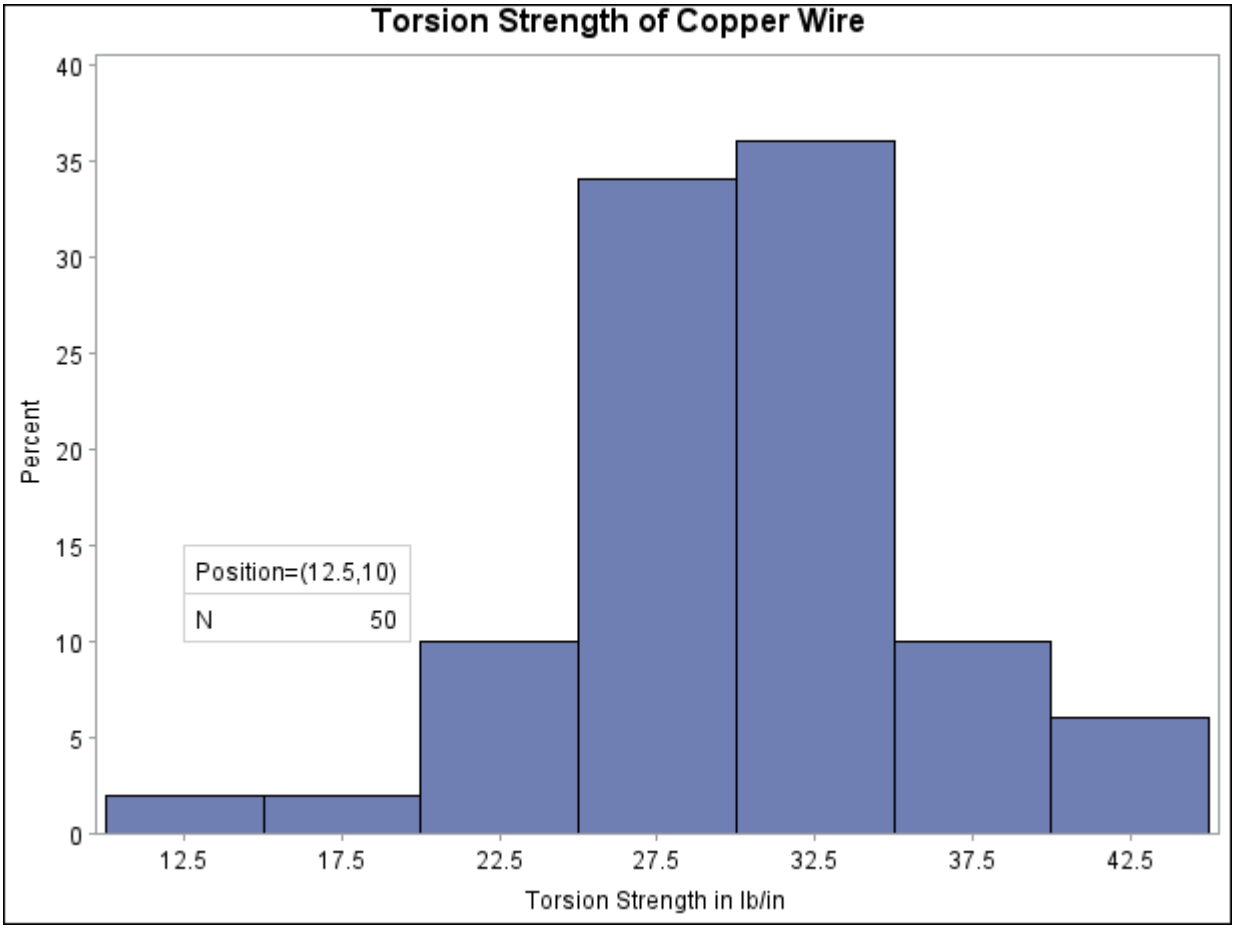
NOTE: See *Positioning the Inset* in the SAS/QC Sample Library.

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 12.5 on the horizontal axis and 10 on the vertical axis:

```
title 'Torsion Strength of Copper Wire';
proc capability data=Wire;
   histogram Strength;
   inset n / header   = 'Position=(12.5,10)'
             position = (12.5,10) data;
run;
```

The histogram is displayed in Figure 6.23. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

Figure 6.23 Inset Positioned Using Data Unit Coordinates



Axis Percent Unit Coordinates

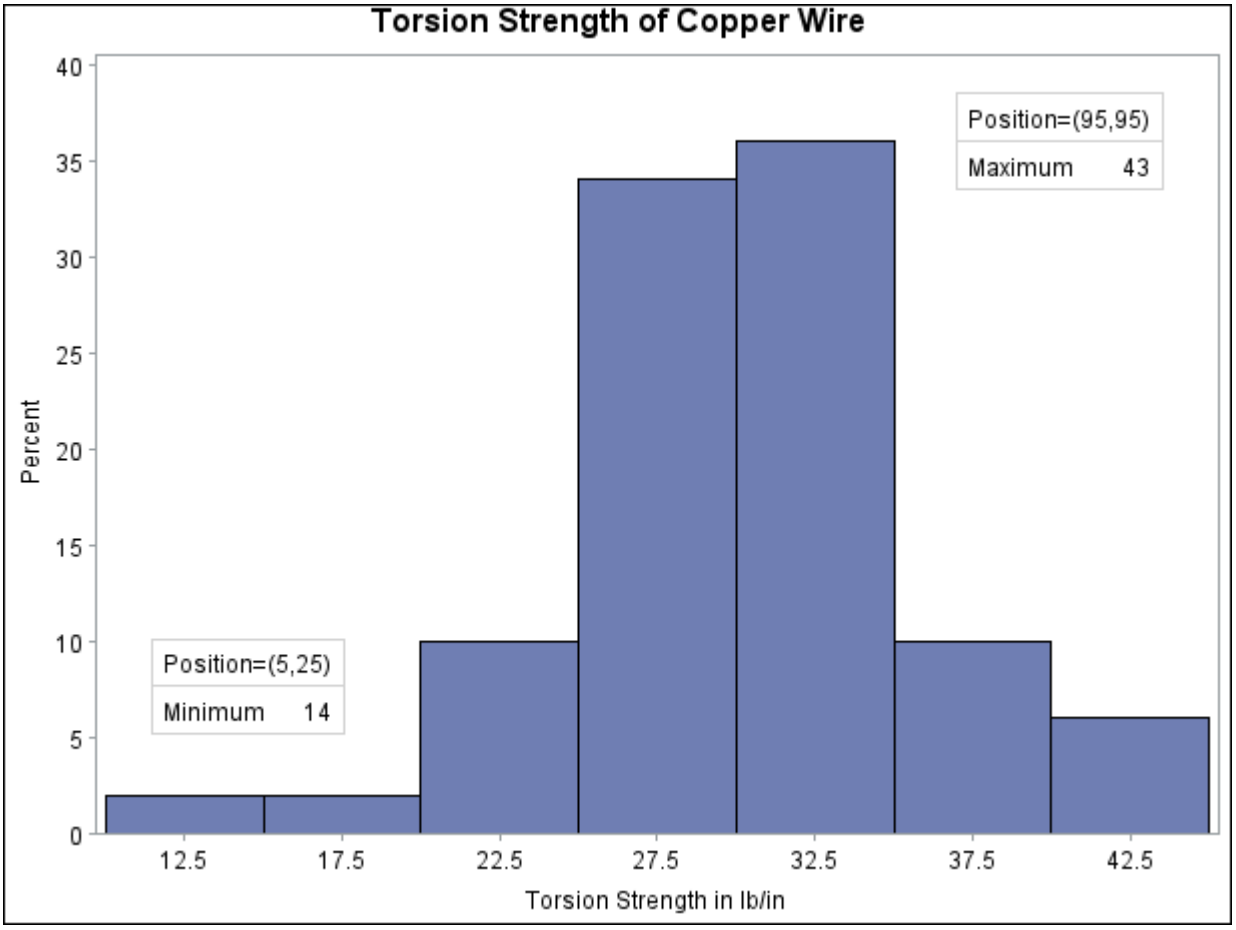
NOTE: See *Positioning the Inset* in the SAS/QC Sample Library.

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a histogram with two insets, both positioned using coordinates in axis percent units:

```
title 'Torsion Strength of Copper Wire';
proc capability data=Wire;
   histogram Strength;
   inset min / position = (5,25)
               header   = 'Position=(5,25)'
               refpoint = tl;
   inset max / position = (95,95)
               header   = 'Position=(95,95)'
               refpoint = tr;
run;
```

The display is shown in Figure 6.24. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

Figure 6.24 Inset Positioned Using Axis Percent Unit Coordinates



Examples: INSET Statement

This section provides advanced examples that use the INSET statement.

Example 6.15: Inset for Goodness-of-Fit Statistics

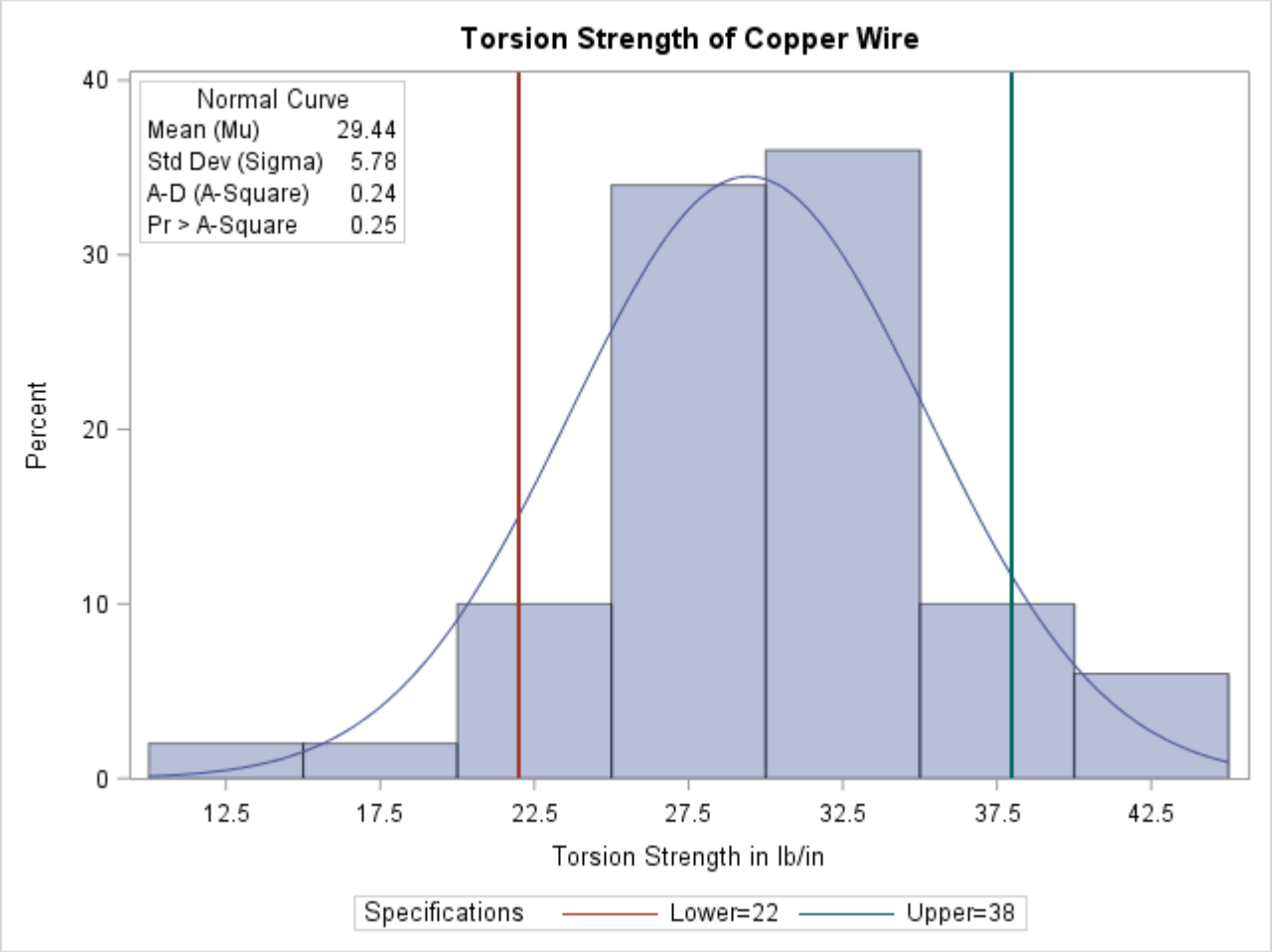
NOTE: See *Inset for Goodness-of-Fit Statistics* in the SAS/QC Sample Library.

This example fits a normal curve to the torsion strength data used in the section "Getting Started: INSET Statement" on page 386. The following statements fit a normal curve and request an inset summarizing the fitted curve with the mean, the standard deviation, and the Anderson-Darling goodness-of-fit test:

```
title 'Torsion Strength of Copper Wire';
proc capability data=Wire noprint;
   spec lsl=22 usl=38;
   histogram Strength / normal(noprint)
                        nocurvelegend
                        odstitle = title;
   inset normal(mu sigma ad adpval) / format = 7.2;
run;
```

The resulting histogram is displayed in Output 6.15.1. The NOCURVELEGEND option in the HISTOGRAM statement suppresses the default legend for curve parameters.

Output 6.15.1 Inset Table with Normal Curve Information



Example 6.16: Inset for Areas Under a Fitted Curve

NOTE: See *Inset for Areas Under a Fitted Curve* in the SAS/QC Sample Library.

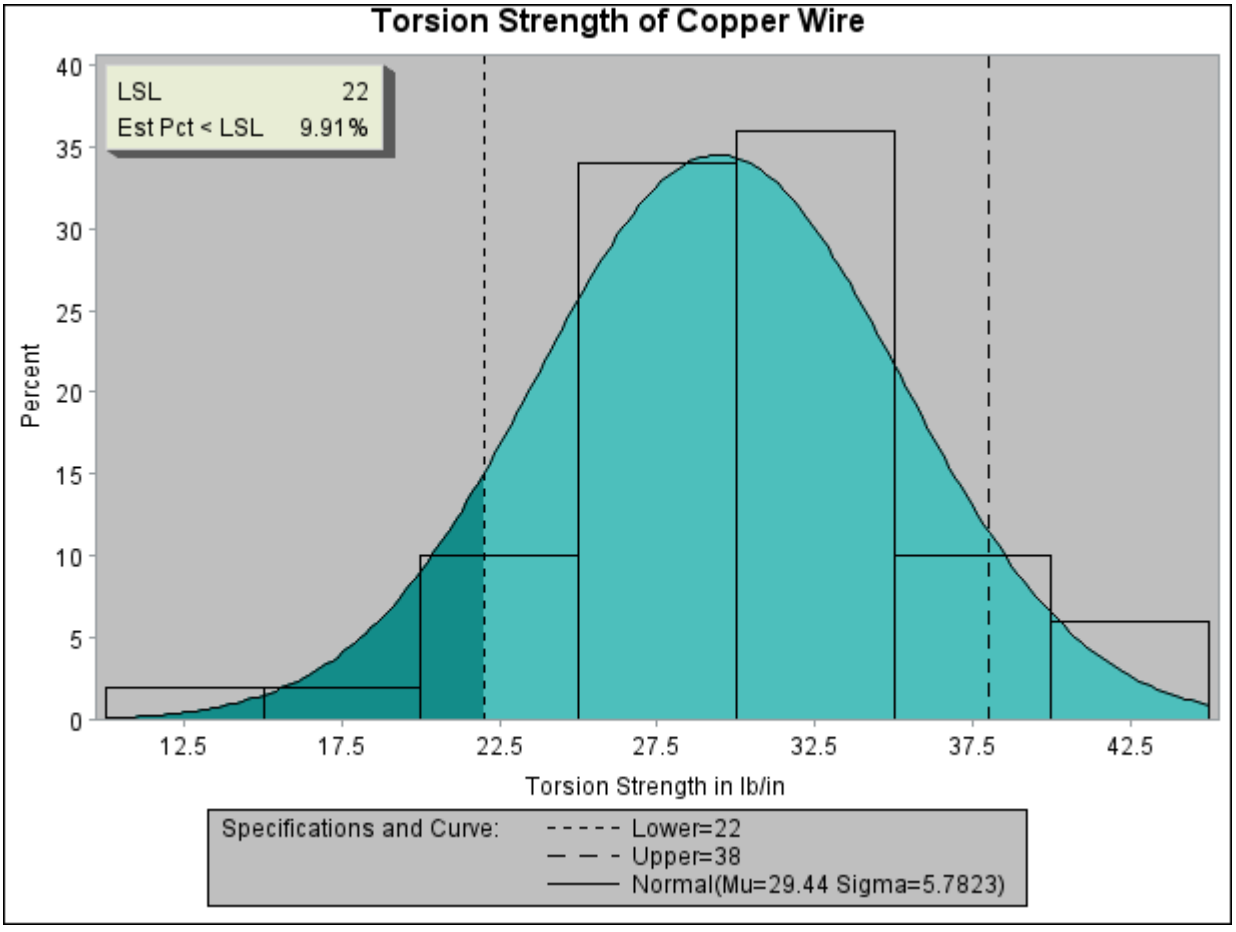
You can use the INSET keywords LSLPCT, USLPCT, and BETWEENPCT to inset legends for areas under histogram bars or fitted curves. The following statements create a histogram with an inset legend for the shaded area under the fitted normal curve to the left of the lower specification limit:

```
ods graphics off;
title 'Torsion Strength of Copper Wire';
legend2 FRAME CFRAME=ligr CBORDER=black POSITION=center;
proc capability data=Wire noprint;
   spec lsl=22 llsl=2  clsl=black cleft=vibg
        usl=38 lusl=20 cusl=black;
   histogram Strength /  cframe = ligr
                         cfill   = bibg
                         legend = legend2
                         normal(color=black noprint fill);
   inset  lsl='LSL' lslpct / cfill=ywh cshadow=dagr;
run;
```

The histogram is displayed in Output 6.16.1. The LSLPCT keyword in the INSET statement requests a legend for the area under the curve to the left of the lower specification limit. The CLEFT= option is used to fill the area under the normal curve to the left of the line, and the CFILL= color is used to fill the remaining area. If the FILL *normal-option* were not specified, the CLEFT= and CFILL= colors would be applied to the corresponding areas under the histogram, not the normal curve, and the inset box would reflect the area under the histogram bars.

You can use the USLPCT keyword in the INSET statement to request a legend for the area to the right of an upper specification limit, and you can use the BETWEENPCT keyword to request a legend for the area between the lower and upper limits. By default, the legend requested with each of the keywords LSLPCT, USLPCT, and BETWEENPCT displays a rectangle that matches the color of the corresponding area. You can substitute a customized label for each rectangle by specifying the keyword followed by an equal sign (=) and the label in quotes.

Output 6.16.1 Displaying Areas Under the Normal Curve



INTERVALS Statement: CAPABILITY Procedure

Overview: INTERVALS Statement

The INTERVALS statement tabulates various statistical intervals for selected process variables. The types of intervals you can request include

- approximate simultaneous prediction intervals for future observations
- prediction intervals for the mean of future observations
- statistical tolerance intervals that contain at least a specified proportion of the population
- confidence intervals for the population mean
- prediction intervals for the standard deviation of future observations
- confidence intervals for the population standard deviation

These intervals are computed assuming the data are sampled from a normal population. See Hahn and Meeker (1991) for a detailed discussion of these intervals.

You can use options in the INTERVALS statement to

- specify which intervals to compute
- provide probability or confidence levels for intervals
- suppress printing of output tables
- create an output data set containing interval information
- specify interval type (one-sided lower, one-sided upper, or two-sided)

Getting Started: INTERVALS Statement

This section introduces the INTERVALS statement with simple examples that illustrate commonly used options. Complete syntax for the INTERVALS statement is presented in the section "Syntax: INTERVALS Statement" on page 417.

Computing Statistical Intervals

NOTE: See *Calculating Various Statistical Intervals* in the SAS/QC Sample Library.

The following statements create the data set Cans, which contains measurements (in ounces) of the fluid weights of 100 drink cans. The filling process is assumed to be in statistical control.

```
data Cans;
   label Weight = "Fluid Weight (ounces)";
   input Weight @@;
   datalines;
12.07  12.02  12.00  12.01  11.98  11.96  12.04  12.05  12.01  11.97
12.03  12.03  12.00  12.04  11.96  12.02  12.06  12.00  12.02  11.91
12.05  11.98  11.91  12.01  12.06  12.02  12.05  11.90  12.07  11.98
12.02  12.11  12.00  11.99  11.95  11.98  12.05  12.00  12.10  12.04
12.06  12.04  11.99  12.06  11.99  12.07  11.96  11.97  12.00  11.97
12.09  11.99  11.95  11.99  11.99  11.96  11.94  12.03  12.09  12.03
11.99  12.00  12.05  12.04  12.05  12.01  11.97  11.93  12.00  11.97
12.13  12.07  12.00  11.96  11.99  11.97  12.05  11.94  11.99  12.02
11.95  11.99  11.91  12.06  12.03  12.06  12.05  12.04  12.03  11.98
12.05  12.05  12.11  11.96  12.00  11.96  11.96  12.00  12.01  11.98
;
```

Note that this data set is introduced in "Computing Descriptive Statistics" on page 197 of "PROC CAPABILITY and General Statements" on page 195. The analysis in that section provides evidence that the weight measurements are normally distributed.

By default, the INTERVALS statement computes and prints the six intervals described in the entry for the METHODS= option. The following statements tabulate these intervals for the variable Weight:

```
title 'Statistical Intervals for Fluid Weight';
proc capability data=Cans noprint;
   intervals Weight;
run;
```

The intervals are displayed in Figure 6.27.

Figure 6.25 Statistical Intervals for Weight

Statistical Intervals for Fluid Weight

The CAPABILITY Procedure
Two-Sided Statistical Intervals for Weight Assuming Normality

Approximate Prediction Interval Containing All of k Future Observations		
Confidence	k	Prediction Limits
99.00%	1	11.89 12.13
99.00%	2	11.87 12.14
99.00%	3	11.87 12.15
95.00%	1	11.92 12.10
95.00%	2	11.90 12.12
95.00%	3	11.89 12.12
90.00%	1	11.93 12.09
90.00%	2	11.92 12.10
90.00%	3	11.91 12.11

Figure 6.25 *continued*

Prediction Interval Containing the Mean of k Future Observations

Confidence	k	Prediction Limits	
99.00%	1	11.89	12.13
99.00%	2	11.92	12.10
99.00%	3	11.94	12.08
95.00%	1	11.92	12.10
95.00%	2	11.94	12.08
95.00%	3	11.95	12.06
90.00%	1	11.93	12.09
90.00%	2	11.95	12.06
90.00%	3	11.96	12.05

Tolerance Interval Containing At Least Proportion p of the Population

Confidence	p	Tolerance Limits	
99.00%	0.900	11.92	12.10
99.00%	0.950	11.90	12.12
99.00%	0.990	11.86	12.15
95.00%	0.900	11.92	12.10
95.00%	0.950	11.90	12.11
95.00%	0.990	11.87	12.15
90.00%	0.900	11.92	12.09
90.00%	0.950	11.91	12.11
90.00%	0.990	11.88	12.14

Confidence Limits Containing the Mean

Confidence	Confidence Limits	
99.00%	11.997	12.022
95.00%	12.000	12.019
90.00%	12.002	12.017

Prediction Interval Containing the Standard Deviation of k Future Observations

Confidence	k	Prediction Limits	
99.00%	2	0.0003	0.1348
99.00%	3	0.0033	0.1110
95.00%	2	0.0015	0.1069
95.00%	3	0.0075	0.0919
90.00%	2	0.0030	0.0932
90.00%	3	0.0106	0.0825

Figure 6.25 *continued*

Confidence Limits Containing the Standard Deviation		
Confidence	Confidence Limits	
99.00%	0.040	0.057
95.00%	0.041	0.055
90.00%	0.042	0.053

Computing One-Sided Lower Prediction Limits

NOTE: See *Calculating Various Statistical Intervals* in the SAS/QC Sample Library.

You can specify options after the slash (/) in the INTERVALS statement to control the computation and printing of intervals. The following statements produce a table of one-sided lower prediction limits for the mean, which is displayed in Figure 6.26:

```
title 'Statistical Intervals for Fluid Weight';
proc capability data=Cans noprint;
   intervals Weight / methods = 1 2
                      type    = lower;
run;
```

The METHODS= option specifies which intervals to compute, and the TYPE= option requests one-sided lower limits. All the options available in the INTERVALS statement are listed in "Summary of Options" on page 417 and are described in "Dictionary of Options" on page 418.

Figure 6.26 One-Sided Lower Prediction Limits for the Mean

Statistical Intervals for Fluid Weight

The CAPABILITY Procedure
One-Sided Lower Statistical Intervals for Weight Assuming Normality

Approximate Prediction Limit For All of k Future Observations		
Confidence	k	Lower Limit
99.00%	1	11.90
99.00%	2	11.89
99.00%	3	11.88
95.00%	1	11.93
95.00%	2	11.92
95.00%	3	11.91
90.00%	1	11.95
90.00%	2	11.93
90.00%	3	11.92

Figure 6.26 *continued*

Prediction Limit For the Mean of k Future Observations		
Confidence	k	Lower Limit
99.00%	1	11.90
99.00%	2	11.93
99.00%	3	11.94
95.00%	1	11.93
95.00%	2	11.95
95.00%	3	11.96
90.00%	1	11.95
90.00%	2	11.97
90.00%	3	11.97

Syntax: INTERVALS Statement

The syntax for the INTERVALS statement is as follows:

INTERVALS < *variables* > < / *options* > **;**

You can specify INTERVAL as an alias for INTERVALS. You can use any number of INTERVALS statements in the CAPABILITY procedure. The components of the INTERVALS statement are described as follows.

variables
gives a list of variables for which to compute intervals. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, the variables can be any numeric variable in the input data set. If you do not specify a list of variables, then by default the INTERVALS statement computes intervals for all variables in the VAR statement (or all numeric variables in the input data set if you do not use a VAR statement).

options
alter the defaults for computing and printing intervals and for creating output data sets.

Summary of Options

The following tables list the INTERVALS statement options by function. For complete descriptions, see "Dictionary of Options" on page 418.

Table 6.49 INTERVAL Statement Options

Option	Description
ALPHA=	specifies probability or confidence levels associated with the intervals
K=	specifies values of k for prediction intervals
METHODS=	specifies which intervals are computed
NOPRINT	suppresses the output tables
OUTINTERVALS=	specifies an output data set containing interval information
P=	specifies values of p for tolerance intervals
TYPE=	specifies the type of intervals (one-sided lower, one-sided upper, or two-sided)

Dictionary of Options

The following entries provide detailed descriptions of options in the INTERVALS statement.

ALPHA=*value-list*

specifies values of α, the probability or confidence associated with the interval. For example, the following statements tabulate the default intervals at probability or confidence levels of $\alpha = 0.05$, $\alpha = 0.10$, $\alpha = 0.15$, and $\alpha = 0.20$:

```
proc capability data=steel;
   intervals width / alpha = 0.05 0.10 0.15 0.20;
run;
```

Note that some references use $\gamma = 1 - \alpha$ to denote probability or confidence levels. Values for the ALPHA= option must be between 0.00001 to 0.99999. By default, values of 0.01, 0.05, and 0.10 are used.

K=*value-list*

specifies values of k for prediction intervals. Default values of 1, 2, and 3 are used for the prediction interval for k future observations and for the prediction interval for the mean of k future observations. Default values of 2 and 3 are used for the prediction interval for the standard deviation of k future observations. The values must be integers.

METHODS=*indices*

METHOD=*indices*

specifies which intervals are computed. The indices can range from 1 to 6, and they correspond to the intervals described in Table 6.50.

Table 6.50 Intervals Computed for METHOD=*Index*

Index	Statistical Interval
1	approximate simultaneous prediction interval for k future observations
2	prediction interval for the mean of k future observations
3	statistical tolerance interval that contains at least proportion p of the population
4	confidence interval for the population mean
5	prediction interval for the standard deviation of k future observations
6	confidence interval for the population standard deviation

For example, the following statements tabulate confidence limits for the population mean (METHOD=4) and confidence limits for the population standard deviation (METHOD=6):

```
proc capability data=steel;
   intervals width / methods=4 6;
run;
```

Formulas for the intervals are given in "Methods for Computing Statistical Intervals" on page 420. By default, the procedure computes all six intervals.

NOPRINT
suppresses the tables produced by default. This option is useful when you only want to save the interval information in an OUTINTERVALS= data set.

OUTINTERVALS=*SAS-data-set*

OUTINTERVAL=*SAS-data-set*

OUTINT=*SAS-data-set*
specifies an output SAS data set containing the intervals and related information. For example, the following statements create a data set named ints containing intervals for the variable width:

```
proc capability data=steel;
   intervals width / outintervals=ints;
run;
```

See "OUTINTERVALS= Data Set" on page 423 for details.

P=*value-list*
specifies values of p for the tolerance intervals. These values must be between 0.00001 to 0.99999. Note that the P= option applies only to the tolerance intervals (METHODS=3). By default, values of 0.90, 0.95, and 0.99 are used.

TYPE=LOWER | UPPER | TWOSIDED
determines whether the intervals computed are one-sided lower, one-sided upper, or two-sided intervals, respectively. See "Computing One-Sided Lower Prediction Limits" on page 416 for an example. The default interval type is TWOSIDED.

Details: INTERVALS Statement

This section provides details on the following topics:

- formulas for statistical intervals
- OUTINTERVALS= data sets

Methods for Computing Statistical Intervals

The formulas for statistical intervals given in this section use the following notation:

Notation	Definition
n	number of nonmissing values for a variable
$\bar{X}$	mean of variable
s	standard deviation of variable
z_α	100αth percentile of the standard normal distribution
$t_\alpha(\nu)$	100αth percentile of the central t distribution with ν degrees of freedom
$t'_\alpha(\delta, \nu)$	100αth percentile of the noncentral t distribution with noncentrality parameter δ and ν degrees of freedom
$F_\alpha(\nu_1, \nu_2)$	100αth percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator
$\chi^2_\alpha(\nu)$	100αth percentile of the χ^2 distribution with ν degrees of freedom
$\chi^2_\alpha(\delta, \nu)$	100αth percentile of the noncentral χ^2 distribution with noncentrality parameter δ and ν degrees of freedom

The values of the variable are assumed to be independent and normally distributed. The intervals are computed using the degrees of freedom as the divisor for the standard deviation s. This divisor corresponds to the default of VARDEF=DF in the PROC CAPABILITY statement. If you specify another value for the VARDEF= option, intervals are not computed.

You select the intervals to be computed with the METHODS= option. The next six sections give computational details for each of the METHODS= options.

METHODS=1

This requests an approximate simultaneous prediction interval for k future observations. Two-sided intervals are computed using the conservative approximations

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{2k}}(n-1)s\sqrt{1+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{2k}}(n-1)s\sqrt{1+\frac{1}{n}}$$

One-sided limits are computed using the conservative approximation

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{k}}(n-1)s\sqrt{1+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{k}}(n-1)s\sqrt{1+\frac{1}{n}}$$

Hahn (1970b) states that these approximations are satisfactory except for combinations of small n, large k, and large α. Refer also to Hahn (1969, 1970a) and Hahn and Meeker (1991).

METHODS=2

This requests a prediction interval for the mean of k future observations. Two-sided intervals are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{2}}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{2}}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

One-sided limits are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\alpha}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\alpha}(n-1)s\sqrt{\frac{1}{k}+\frac{1}{n}}$$

METHODS=3

This requests a statistical tolerance interval that contains at least proportion p of the population. Two-sided intervals are computed as

$$\text{Lower Limit} = \bar{X} - ks$$

$$\text{Upper Limit} = \bar{X} + ks$$

where k is the solution of the integral equation

$$\sqrt{\frac{2n}{\pi}}\int_0^\infty P\left(\chi^2_{n-1} > \frac{(n-1)\chi^2_p(z^2,1)}{k^2}\right)e^{-\frac{1}{2}nz^2}dz = 1-\alpha$$

One-sided limits are computed as

$$\text{Lower Limit} = \bar{X} - g'(p\,;n\,;1-\alpha)s$$

$$\text{Upper Limit} = \bar{X} + g'(p\,;n\,;1-\alpha)s$$

where $g'(p\,;n\,;1-\alpha) = \frac{1}{\sqrt{n}}t'_{1-\alpha}(z_p\sqrt{n}, n-1)$.

For a thorough discussion of tolerance intervals and tolerance limits, see Krishnamoorthy and Mathew (2009).

METHODS=4

This requests a confidence interval for the population mean. Two-sided intervals are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\frac{\alpha}{2}}(n-1)\frac{s}{\sqrt{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\frac{\alpha}{2}}(n-1)\frac{s}{\sqrt{n}}$$

One-sided limits are computed as

$$\text{Lower Limit} = \bar{X} - t_{1-\alpha}(n-1)\frac{s}{\sqrt{n}}$$

$$\text{Upper Limit} = \bar{X} + t_{1-\alpha}(n-1)\frac{s}{\sqrt{n}}$$

METHODS=5

This requests a prediction interval for the standard deviation of k future observations. Two-sided intervals are computed as

$$\text{Lower Limit} = s\left(F_{1-\frac{\alpha}{2}}(n-1,k-1)\right)^{-\frac{1}{2}}$$

$$\text{Upper Limit} = s\left(F_{1-\frac{\alpha}{2}}(k-1,n-1)\right)^{\frac{1}{2}}$$

One-sided limits are computed as

$$\text{Lower Limit} = s\left(F_{1-\alpha}(n-1,k-1)\right)^{-\frac{1}{2}}$$

$$\text{Upper Limit} = s\left(F_{1-\alpha}(k-1,n-1)\right)^{\frac{1}{2}}$$

METHODS=6

This requests a confidence interval for the population standard deviation. Two-sided intervals are computed as

$$\text{Lower Limit} = s\sqrt{\frac{n-1}{\chi^2_{1-\frac{\alpha}{2}}(n-1)}}$$

$$\text{Upper Limit} = s\sqrt{\frac{n-1}{\chi^2_{\frac{\alpha}{2}}(n-1)}}$$

One-sided limits are computed as

$$\text{Lower Limit} = s\sqrt{\frac{n-1}{\chi^2_{1-\alpha}(n-1)}}$$

$$\text{Upper Limit} = s\sqrt{\frac{n-1}{\chi^2_{\alpha}(n-1)}}$$

OUTINTERVALS= Data Set

Each INTERVALS statement can create an output data set specified with the OUTINTERVALS= option. The OUTINTERVALS= data set contains statistical intervals and related parameters.

The number of observations in the OUTINTERVALS= data set depends on the number of variables analyzed, the number of tests specified, and the results of the tests. The OUTINTERVALS= data set is constructed as follows:

- The OUTINTERVALS= data set contains a group of observations for each variable analyzed.
- Each group contains one or more observations for each interval you specify with the METHODS= option. The actual number depends upon the number of combinations of the ALPHA=, K=, and P= values.

The following variables are saved in the OUTINTERVALS= data set:

Variable	Description
ALPHA	value of α associated with the intervals
_K_	value of K= for the prediction intervals
LOWER	lower endpoint of interval
METHOD	interval index (1–6)
_P_	value of P= for the tolerance intervals
TYPE	type of interval (ONESIDED or TWOSIDED)
UPPER	upper endpoint of interval
VAR	variable name

If you use a BY statement, the BY variables are also saved in the OUTINTERVALS= data set.

ODS Tables

The following table summarizes the ODS tables that you can request with the INTERVALS statement.

Table 6.51 ODS Tables Produced with the INTERVALS Statement

Table Name	Description	Option
Intervals1	prediction interval for future observations	METHODS=1
Intervals2	prediction interval for mean	METHODS=2
Intervals3	tolerance interval for proportion of population	METHODS=3
Intervals4	confidence limits for mean	METHODS=4
Intervals5	prediction interval for standard deviation	METHODS=5
Intervals6	confidence limits for standard deviation	METHODS=6

OUTPUT Statement: CAPABILITY Procedure

Overview: OUTPUT Statement

You can use the OUTPUT statement to save summary statistics in a SAS data set. This information can then be used to create customized reports or to save historical information about a process.

You can use options in the OUTPUT statement to

- specify the statistics to save in the output data set
- specify the name of the output data set
- compute and save percentiles not automatically computed by the CAPABILITY procedure

Getting Started: OUTPUT Statement

This section introduces the OUTPUT statement with simple examples that illustrate commonly used options. Complete syntax for the OUTPUT statement is presented in the section "Syntax: OUTPUT Statement" on page 427, and advanced examples are given in the section "Examples: OUTPUT Statement" on page 434.

Saving Summary Statistics in an Output Data Set

NOTE: See *Saving CAPABILITY Output in a Data Set* in the SAS/QC Sample Library.

An automobile manufacturer producing seat belts saves summary information in an output data set with the CAPABILITY procedure. The following statements create the data set Belts, which contains the breaking strengths (Strength) and widths (Width) of a sample of 50 belts:

```
data Belts;
   label Strength = 'Breaking Strength (lb/in)'
         Width    = 'Width in Inches';
   input Strength Width @@;
   datalines;
1243.51  3.036  1221.95  2.995  1131.67  2.983  1129.70  3.019
1198.08  3.106  1273.31  2.947  1250.24  3.018  1225.47  2.980
1126.78  2.965  1174.62  3.033  1250.79  2.941  1216.75  3.037
1285.30  2.893  1214.14  3.035  1270.24  2.957  1249.55  2.958
1166.02  3.067  1278.85  3.037  1280.74  2.984  1201.96  3.002
1101.73  2.961  1165.79  3.075  1186.19  3.058  1124.46  2.929
1213.62  2.984  1213.93  3.029  1289.59  2.956  1208.27  3.029
1247.48  3.027  1284.34  3.073  1209.09  3.004  1146.78  3.061
1224.03  2.915  1200.43  2.974  1183.42  3.033  1195.66  2.995
1258.31  2.958  1136.05  3.022  1177.44  3.090  1246.13  3.022
1183.67  3.045  1206.50  3.024  1195.69  3.005  1223.49  2.971
1147.47  2.944  1171.76  3.005  1207.28  3.065  1131.33  2.984
1215.92  3.003  1202.17  3.058
;
```

The following statements produce two output data sets containing summary statistics:

```
proc capability data=Belts;
   var Strength Width;
   output out=Means    mean=smean wmean;
   output out=Strstats mean=smean std=sstd min=smin max=smax;
run;

proc print data=Means;
run;

proc print data=Strstats;
run;
```

Note that if you specify an OUTPUT statement, you must also specify a VAR statement. You can use multiple OUTPUT statements with a single procedure statement. Each OUTPUT statement creates a new data set. The OUT= option specifies the name of the output data set. In this case, two data sets, Means and Strstats, are created. See Figure 6.27 for a listing of Means and Figure 6.28 for a listing of Strstats.

Summary statistics are saved in an output data set by specifying *keyword=names* after the OUT= option. In the preceding statements, the first OUTPUT statement specifies the keyword MEAN followed by the names smean and wmean. The second OUTPUT statement specifies the keywords MEAN, STD, MIN, and MAX, for which the names smean, sstd, smin, and smax are given.

The keyword specifies the statistic to be saved in the output data set, and the names determine the names for the new variables. The first name listed after a keyword contains that statistic for the first variable listed in the VAR statement; the second name contains that statistic for the second variable in the VAR statement, and so on.

Thus, the data set Means contains the mean of Strength in a variable named smean and the mean of Width in a variable named wmean. The data set Strstats contains the mean, standard deviation, minimum value, and maximum value of Strength in the variables smean, sstd, smin, and smax, respectively.

Figure 6.27 Listing of the Output Data Set Means

Statistical Intervals for Fluid Weight

Obs	smean	wmean
1	1205.75	3.00584

Figure 6.28 Listing of the Output Data Set Strstats

Statistical Intervals for Fluid Weight

Obs	smean	sstd	smax	smin
1	1205.75	48.3290	1289.59	1101.73

Saving Percentiles in an Output Data Set

NOTE: See *Saving CAPABILITY Output in a Data Set* in the SAS/QC Sample Library.

The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 75th, 90th, 95th, and 99th percentiles for each variable. You can save these percentiles in an output data set by specifying the appropriate keywords. For example, the following statements create an output data set named Pctlstr containing the 5th and 95th percentiles of the variable Strength:

```
proc capability data=Belts noprint;
   var Strength Width;
   output out=Pctlstr p5=p5str p95=p95str;
run;

proc print data=Pctlstr;
run;
```

The output data set Pctlstr is listed in Figure 6.29.

Figure 6.29 Listing of the Output Data Set Pctlstr

Statistical Intervals for Fluid Weight

Obs	p95str	p5str
1	1284.34	1126.78

You can use the PCTLPTS=, PCTLPRE=, and PCTLNAME= options to save percentiles not automatically computed by the CAPABILITY procedure. For example, the following statements create an output data set named Pctls containing the 20th and 40th percentiles of the variables Strength and Width:

```
proc capability data=Belts noprint;
   var Strength Width;
   output out=Pctls pctlpts  = 20 40
                    pctlpre  = S W
                    pctlname = pct20 pct40;
run;

proc print data=Pctls;
run;
```

The PCTLPTS= option specifies the percentiles to compute (in this case, the 20th and 40th percentiles). The PCTLPRE= and PCTLNAME= options build the names for the variables containing the percentiles. The PCTLPRE= option gives prefixes for the new variables, and the PCTLNAME= option gives a suffix to add to the prefix. Note that if you use the PCTLPTS= specification, you must also use the PCTLPRE= specification. For details on these options, see the section "Syntax: OUTPUT Statement" on page 427.

The preceding OUTPUT statement saves the 20th and 40th percentiles of Strength and Width in the variables spct20, wpct20, spct40, and wpct40. The output data set Pctls is listed in Figure 6.30.

Figure 6.30 Listing of the Output Data Set Pctls

Statistical Intervals for Fluid Weight

Obs	Spct20	Wpct20	Spct40	Wpct40
1	1165.91	2.9595	1199.26	2.995

Syntax: OUTPUT Statement

The syntax for the OUTPUT statement is as follows:

OUTPUT < **OUT=***SAS-data-set* > < *keyword1=names . . . keywordk=names* > < *percentile-options* > **;**

You can use any number of OUTPUT statements in the CAPABILITY procedure. Each OUTPUT statement creates a new data set containing the statistics specified in that statement. When you use the OUTPUT statement, you must also use the VAR statement. In addition, the OUTPUT statement must contain at least one of the following:

- a specification of the form *keyword=names*
- the PCTLPTS= and PCTLPRE= options

You can use the OUT= option to specify the name of the output data set:

OUT=*SAS-data-set*
specifies the name of the output data set. To create a permanent SAS data set, specify a two-level name. See *SAS Statements: Reference* for more information on permanent SAS data sets. For example, the previous statements create an output data set named Summary. If the OUT= option is omitted, then by default the new data set is named using the DATA*n* convention.

A *keyword=names* specification selects a statistic to be included in the output data set and gives names to the new variables that contain the statistics. Specify a *keyword* for each desired statistic, an equal sign, and the *names* of the variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains the statistic for the first variable listed in the VAR statement; the second variable contains the statistic for the second variable in the VAR statement, and so on. The list of *names* following the equal sign can be shorter than the list of variables in the VAR statement. In this case, the procedure uses the *names* in the order in which the variables are listed in the VAR statement. Consider the following example:

```
proc capability noprint;
   var length width height;
   output out=summary mean=mlength mwidth;
run;
```

The variables mlength and mwidth contain the means for length and width. The mean for height is computed by the procedure but is not saved in the output data set.

Table 6.52 lists all keywords available in the OUTPUT statement grouped by type. Formulas for selected statistics are given in the section "Details: CAPABILITY Procedure" on page 219.

Table 6.52 OUTPUT Statement Statistic Keywords

Keyword	Description
Descriptive Statistics	
CSS	Sum of squares corrected for the mean
CV	Percent coefficient of variation

Table 6.52 (continued)

Keyword	Description
GEOMEAN	Geometric mean
KURTOSIS | KURT	Kurtosis
MAX	Largest (maximum) value
MEAN	Mean
MIN	Smallest (minimum) value
MODE	Most frequent value (if not unique, the smallest mode)
N	Number of observations on which calculations are based
NMISS	Number of missing values
NOBS	Number of observations
RANGE	Range
SKEWNESS | SKEW	Skewness
STD | STDDEV	Standard deviation
STDMEAN | STDERR	Standard error of the mean
SUM	Sum
SUMWGT	Sum of weights
USS	Uncorrected sum of squares
VAR	Variance
Quantile Statistics	
MEDIAN | P50 | Q2	Median (50th percentile)
P1	1st percentile
P5	5th percentile
P10	10th percentile
P90	90th percentile
P95	95th percentile
P99	99th percentile
Q1 | P25	Lower quartile (25th percentile)
Q3 | P75	Upper quartile (75th percentile)
QRANGE	Interquartile range (Q3 – Q1)
Robust Statistics	
GINI	Gini's mean difference
MAD	Median absolute difference
QN	2nd variation of median absolute difference
SN	1st variation of median absolute difference
STD_GINI	Standard deviation for Gini's mean difference
STD_MAD	Standard deviation for median absolute difference
STD_QN	Standard deviation for the second variation of the median absolute difference
STD_QRANGE	Estimate of the standard deviation, based on interquartile range
STD_SN	Standard deviation for the first variation of the median absolute difference

Table 6.52 (continued)

Keyword	Description
Hypothesis Test Statistics	
MSIGN	Sign statistic
NORMAL	Test statistic for normality. If the sample size is less than or equal to 2000, this is the Shapiro-Wilk W statistic. Otherwise, it is the Kolmogorov D statistic.
PNORMAL | PROBN	p-value for normality test
PROBM	Probability of a greater absolute value for the sign statistic
PROBS	Probability of a greater absolute value for the signed rank statistic
PROBT	Two-tailed p-value for Student's t statistic with $n-1$ degrees of freedom
SIGNRANK	Signed rank statistic
T	Student's t statistic to test the null hypothesis that the population mean is equal to μ_0
Specification Limits and Related Statistics	
LSL	Lower specification limit
PCTGTR	Percent of nonmissing observations greater than the upper specification limit
PCTLSS	Percent of nonmissing observations less than the lower specification limit
TARGET	Target value
USL	Upper specification limit
Capability Indices and Related Statistics	
CP	Capability index C_p
CPLCL	Lower confidence limit for C_p
CPUCL	Upper confidence limit for C_p
CPK	Capability index C_{pk} (also denoted CPK)
CPKLCL	Lower confidence limit for C_{pk}
CPKUCL	Upper confidence limit for C_{pk}
CPL	Capability index CPL
CPLLCL	Lower confidence limit for CPL
CPLUCL	Upper confidence limit for CPL
CPM	Capability index C_{pm}
CPMLCL	Lower confidence limit for C_{pm}
CPMUCL	Upper confidence limit for C_{pm}
CPU	Capability index CPU
CPULCL	Lower confidence limit for CPU
CPUUCL	Upper confidence limit for CPU
K	Capability index k (also denoted K)

The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th, and 99th percentiles for the data. You can save these statistics in an output data set by using *keyword=names* specifications. You can request additional percentiles by using the PCTLPTS= option. The following *percentile-options* are related to these additional percentiles:

CIPCTLDF=(*cipctl-options***)**

CIQUANTDF=(*cipctl-options***)**

requests distribution-free confidence limits for percentiles that are requested with the PCTLPTS= option. In other words, no specific parametric distribution such as the normal is assumed for the data. PROC CAPABILITY uses order statistics (ranks) to compute the confidence limits as described by Hahn and Meeker (1991). This option does not apply if you use a WEIGHT statement. You can specify the following *cipctl-options*:

ALPHA=*α*

specifies the level of significance α for $100(1-\alpha)\%$ confidence intervals. The value α must be between 0 and 1; the default value is 0.05, which results in 95% confidence intervals. The default value is the value of ALPHA= given in the PROC statement.

LOWERPRE=*prefixes*

specifies one or more prefixes that are used to create names for variables that contain the lower confidence limits. To save lower confidence limits for more than one analysis variable, specify a list of prefixes. The order of the prefixes corresponds to the order of the analysis variables in the VAR statement.

LOWERNAME=*suffixes*

specifies one or more suffixes that are used to create names for variables that contain the lower confidence limits. PROC CAPABILITY creates a variable name by combining the LOWERPRE= value and suffix name. Because the suffixes are associated with the requested percentiles, list the suffixes in the same order as the PCTLPTS= percentiles.

TYPE=*keyword*

specifies the type of confidence limit, where *keyword* is LOWER, UPPER, SYMMETRIC, or ASYMMETRIC. The default value is SYMMETRIC.

UPPERPRE=*prefixes*

specifies one or more prefixes that are used to create names for variables that contain the upper confidence limits. To save upper confidence limits for more than one analysis variable, specify a list of prefixes. The order of the prefixes corresponds to the order of the analysis variables in the VAR statement.

UPPERNAME=*suffixes*

specifies one or more suffixes that are used to create names for variables that contain the upper confidence limits. PROC CAPABILITY creates a variable name by combining the UPPERPRE= value and suffix name. Because the suffixes are associated with the requested percentiles, list the suffixes in the same order as the PCTLPTS= percentiles.

NOTE: See the entries for the PCTLPTS=, PCTLPRE=, and PCTLNAME= options for a detailed description of how variable names are created using prefixes, percentile values, and suffixes.

CIPCTLNORMAL=(*cipctl-options***)**

CIQUANTNORMAL=(*cipctl-options***)**

requests confidence limits based on the assumption that the data are normally distributed for percentiles that are requested with the PCTLPTS= option. The computational method is described in Section 4.4.1 of Hahn and Meeker (1991) and uses the noncentral *t* distribution as given by Odeh and Owen (1980). This option does not apply if you use a WEIGHT statement. You can specify the following *cipctl-options*:

ALPHA=*α*
specifies the level of significance α for $100(1-\alpha)\%$ confidence intervals. The value α must be between 0 and 1; the default value is 0.05, which results in 95% confidence intervals. The default value is the value of ALPHA= given in the PROC statement.

LOWERPRE=*prefixes*
specifies one or more prefixes that are used to create names for variables that contain the lower confidence limits. To save lower confidence limits for more than one analysis variable, specify a list of prefixes. The order of the prefixes corresponds to the order of the analysis variables in the VAR statement.

LOWERNAME=*suffixes*
specifies one or more suffixes that are used to create names for variables that contain the lower confidence limits. PROC CAPABILITY creates a variable name by combining the LOWERPRE= value and suffix name. Because the suffixes are associated with the requested percentiles, list the suffixes in the same order as the PCTLPTS= percentiles.

TYPE=*keyword*
specifies the type of confidence limit, where *keyword* is LOWER, UPPER, or TWOSIDED. The default is TWOSIDED.

UPPERPRE=*prefixes*
specifies one or more prefixes that are used to create names for variables that contain the upper confidence limits. To save upper confidence limits for more than one analysis variable, specify a list of prefixes. The order of the prefixes corresponds to the order of the analysis variables in the VAR statement.

UPPERNAME=*suffixes*
specifies one or more suffixes that are used to create names for variables that contain the upper confidence limits. PROC CAPABILITY creates a variable name by combining the UPPERPRE= value and suffix name. Because the suffixes are associated with the requested percentiles, list the suffixes in the same order as the PCTLPTS= percentiles.

NOTE: See the entries for the PCTLPTS=, PCTLPRE=, and PCTLNAME= options for a detailed description of how variable names are created using prefixes, percentile values, and suffixes.

PCTLGROUP=BYSTAT | BYVAR
specifies the order in which variables that you request with the PCTLPTS= option are added to the OUT= data set when the VAR statement lists more than one analysis variable. By default (or if you specify PCTLGROUP=BYSTAT), all variables that are associated with a percentile value are created consecutively. If you specify PCTLGROUP=BYVAR, all variables that are associated with an analysis variable are created consecutively.

Consider the following statements:

```
proc univariate data=Score;
   var PreTest PostTest;
   output out=ByStat pctlpts=20 40 pctlpre=Pre_ Post_;
   output out=ByVar pctlgroup=byvar pctlpts=20 40 pctlpre=Pre_ Post_;
run;
```

The order of variables in the data set ByStat is Pre_20, Post_20, Pre_40, Post_40. The order of variables in the data set ByVar is Pre_20, Pre_40, Post_20, Post_40.

PCTLNAME=*suffixes*

provides name suffixes for the new variables created by the PCTLPTS= option. These suffixes are appended to the prefixes you specify with the PCTLPRE= option, replacing the percentile values that are used as suffixes by default. List the suffixes in the same order in which you specify the percentiles. If you specify n suffixes with the PCTLNAME= option and m percentile values with the PCTLPTS= option, where $m > n$, the suffixes are used to name the first n percentiles, and the default names are used for the remaining $m - n$ percentiles. For example, consider the following statements:

```
proc capability;
   var length width height;
   output pctlpts  = 20 40
          pctlpre  = pl pw ph
          pctlname = twenty;
run;
```

The value "twenty" in the PCTLNAME= option is used for only the first percentile in the PCTLPTS= list. This suffix is appended to the values in the PCTLPRE= option to generate the new variable names pltwenty, pwtwenty, and phtwenty, which contain the 20th percentiles for length, width, and height, respectively. Because a second PCTLNAME= suffix is not specified, variable names for the 40th percentiles for length, width, and height are generated using the prefixes and percentile values. Thus, the output data set contains the variables pltwenty, pl40, pwtwenty, pw40, phtwenty, and ph40.

PCTLNDEC=*value*

specifies the number of decimal places in percentile values that are incorporated into percentile variable names. The default value is 1. For example, the following statements create two output data sets, each containing one percentile variable. The variable in data set short is named pwid85_1, while the one in data set long is named pwid85_125.

```
proc capability;
   var width;
   output out=short pctlpts=85.125 pctlpre=pwid;
   output out=long  pctlpts=85.125 pctlpre=pwid pctlndec=3;
run;
```

PCTLPRE=*prefixes*

specifies prefixes used to create variable names for percentiles requested with the PCTLPTS= option. The PCTLPRE= and PCTLPTS= options must be used together.

The procedure generates new variable names by using the prefix and the percentile values. If the specified percentile is an integer, the variable name is simply the prefix followed by the value. For noninteger percentiles, an underscore replaces the decimal point in the variable name, and decimal values are truncated to one decimal place. For example, the following statements create the variables pwid20, pwid33_3, pwid66_6, and pwid80 for the 20th, 33.33rd, 66.67th, and 80th percentiles of width, respectively:

```
proc capability noprint;
   var width;
   output pctlpts=20 33.33 66.67 80 pctlpre=pwid;
run;
```

If you request percentiles for more than one variable, you should list prefixes in the same order in which the variables appear in the VAR statement. For example, the following statements compute the 80th and 87.5th percentiles for length and width and save the new variables plength80, plength87_5, pwidth80, and pwidth87_5 in the output data set:

```
proc capability noprint;
   var length width;
   output pctlpts=80 87.5 pctlpre=plength pwidth;
run;
```

PCTLPTS=*percentiles*

specifies *percentiles* that are not automatically computed by the procedure. The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th, and 99th percentiles for the data. These can be saved in an output data set by using keyword=names specifications. The PCTLPTS= option generates additional percentiles and outputs them to a data set; these additional percentiles are not printed.

If you use the PCTLPTS= option, you must also use the PCTLPRE= option to provide a prefix for the new variable names. For example, to create variables that contain the 20th, 40th, 60th, and 80th percentiles of length, use the following statements:

```
proc capability noprint;
   var length;
   output pctlpts=20 40 60 80 pctlpre=plen;
run;
```

This creates the variables plen20, plen40, plen60, and plen80, whose values are the corresponding percentiles of length. In addition to specifying name prefixes with the PCTLPRE= option, you can also use the PCTLNAME= option to create name suffixes for the new variables created by the PCTLPTS= option.

Details: OUTPUT Statement

OUT= Data Set

The CAPABILITY procedure creates an OUT= data set for each OUTPUT statement. The new data set contains an observation for each combination of levels of the variables in the BY and CLASS statements, or a single observation if you do not specify a BY or CLASS statement. Thus, the number of observations in the new data set corresponds to the number of groups for which statistics are calculated. The variables in the new data set are as follows:

- variables in the BY statement. The values of these variables match the values in the corresponding BY group in the DATA= data set.
- variables in the CLASS statement. The values of these variables identify the CLASS level within a BY group in the DATA= data set that from which statistics are computed.
- variables in the ID statement. The values of these variables match those for the first observation in each BY group, or for the first observation in the data set if you do not specify a BY statement.
- variables created by selecting statistics in the OUTPUT statement. The values of the statistics are computed using all the nonmissing data, or statistics are computed for each BY group if you use a BY statement.
- variables created by requesting new percentiles with the PCTLPTS= option. The names of these new variables depend on the values of the PCTLPRE= and PCTLNAME= options.

If the output data set contains a percentile variable or a quartile variable, the percentile definition assigned with the PCTLDEF= option in the PROC CAPABILITY statement is recorded on the output data set label.

The values of variables requested with the statistics keywords CP, CPK, CPL, CPM, CPU, K, PCTGTR, and PCTLSS are missing unless you identify specification limits in a SPEC statement or in a SPEC= data set.

As an alternative to OUT= data sets, you can create an OUTTABLE= data set. The structure of the OUTTABLE= data set may be more appropriate when you are computing summary statistics and capability indices for multiple process variables. See "OUTTABLE= Data Set" on page 222.

Examples: OUTPUT Statement

This section provides additional examples of the OUTPUT statement.

Example 6.17: Computing Nonstandard Capability Indices

NOTE: See *Computing Nonstandard Capability Indices* in the SAS/QC Sample Library.

In recent years, a number of process capability indices that have been proposed in the research literature are gradually being introduced in applications. As shown in this example, you can compute such indices in the DATA step after using the OUTPUT statement in the CAPABILITY procedure to save various summary statistics.

Hardness measurements (in scaled units) for 50 titanium samples are saved as values of the variable Hardness in the following SAS data set:

```
data Titanium;
   label Hardness = 'Hardness Measurement';
   input Hardness @@;
   datalines;
1.38  1.49  1.43  1.60  1.59
1.34  1.44  1.64  1.83  1.57
1.45  1.74  1.61  1.39  1.63
```

```
1.73  1.61  1.35  1.51  1.47
1.46  1.41  1.56  1.40  1.58
1.43  1.53  1.53  1.58  1.62
1.58  1.46  1.26  1.57  1.41
1.53  1.36  1.63  1.36  1.66
1.49  1.55  1.67  1.41  1.39
1.75  1.37  1.36  1.86  1.49
;
```

The target value for hardness is 1.6, and the lower and upper specification limits are 0.8 and 2.4, respectively. The samples are produced by an in-control process, and the measurements are assumed to be normally distributed.

The following statements use the OUTPUT statement to save various descriptive statistics and an estimate of the index C_{pm} in a data set named Indices:

```
proc capability data=Titanium noprint;
   var Hardness;
   specs lsl=0.8 target=1.6 usl=2.4;
   output out=Indices
      n       = n
      mean    = avg
      std     = std
      var     = var
      lsl     = lsl
      target  = t
      usl     = usl
      pnormal = pnormal
      cpm     = cpm ;
run;
```

In addition to C_{pm}, you want to report an estimate for the index C_{pmk}, which is defined as follows:

$$C_{pmk} = \frac{d - |\mu - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $d = (\text{USL} - \text{LSL})/2$, $m = (\text{USL} + \text{LSL})/2$, and μ and σ are the mean and standard deviation of the normal distribution. Refer to Section 3.6 of Kotz and Johnson (1993). A natural estimator for C_{pmk} is

$$\hat{C}_{pmk} = \frac{d - |\overline{X} - m|}{3\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - T)^2}}$$

The following statements compute this estimate:

```
data Indices;
   set Indices;
   d    = 0.5*( USL - LSL );
   m    = 0.5*( USL + LSL );
   num  = d - abs( avg - m );
   den  = 3 * sqrt( (n-1)*var/n + (avg-t)*(avg-t) );
   cpmk = num/den;
run;
```

```
title "Capability Analysis of Titanium Hardness";
proc print data=Indices noobs;
   var n avg std lsl t usl cpm cpmk pnormal;
run;
```

The results are listed in Output 6.17.1.

Output 6.17.1 Computation of C_{pmk}

Capability Analysis of Titanium Hardness

n	avg	std	lsl	t	usl	cpm	cpmk	pnormal
50	1.5212	0.13295	0.8	1.6	2.4	1.72545	1.56713	0.25111

Note that the p-value for the Kolmogorov-Smirnov test of normality is 0.25111, indicating that the assumption of normality is justified.

The following statements also compute an estimate of the index C_{pm} by using the SPECIALINDICES option:

```
proc capability data=Titanium specialindices;
   var Hardness;
   specs lsl=0.8 target=1.6 usl=2.4;
run;
```

Output 6.17.2 Computation of C_{pmk} by Using the SPECIALINDICES Option

Capability Analysis of Titanium Hardness

The CAPABILITY Procedure
Variable: Hardness (Hardness Measurement)

Process Capability Indices			
Index	Value	95% Confidence Limits	
Cp	2.005745	1.609575	2.401129
CPL	1.808179	1.438675	2.175864
CPU	2.203311	1.757916	2.646912
Cpk	1.808179	1.438454	2.177904
Cpm	1.725446	1.410047	2.066027

Example 6.18: Approximate Confidence Limits for Cpk

NOTE: See *Approximate Confidence Limits for Cpk* in the SAS/QC Sample Library.

This example illustrates how you can use the OUTPUT statement to compute confidence limits for the capability index C_{pk}.

You can request the approximate confidence limits given by Bissell (1990) with the keywords CPKLCL and CPKUCL in the OUTPUT statement. However, this is not the only method that has been proposed for computing confidence limits for C_{pk}. Zhang, Stenback, and Wardrop (1990), referred to here as ZSW,

proposed approximate confidence limits of the form

$$\widehat{C}_{pk} \pm k\widehat{\sigma}_{pk}$$

where $\widehat{\sigma}_{pk}$ is an estimator of the standard deviation of $\widehat{C}_{pk}$. Equation (8) of ZSW provides an approximation to the variance of $\widehat{C}_{pk}$ from which one can obtain $100\gamma\%$ confidence limits for C_{pk} as

$$\begin{aligned}
\text{LCL} &= \widehat{C}_{pk}\left[1 - \Phi^{-1}((1-\gamma)/2)\sqrt{\frac{n-1}{n-3} - \frac{(n-1)\Gamma^2((n-2)/2)}{2\Gamma^2((n-1)/2)}}\right] \\
\text{UCL} &= \widehat{C}_{pk}\left[1 + \Phi^{-1}(1-(1-\gamma)/2)\sqrt{\frac{n-1}{n-3} - \frac{(n-1)\Gamma^2((n-2)/2)}{2\Gamma^2((n-1)/2)}}\right]
\end{aligned}$$

This assumes that $\widehat{C}_{pk}$ is normally distributed. You can also compute approximate confidence limits based on equation (6) of ZSW, which provides an exact expression for the variance of $\widehat{C}_{pk}$.

The following program uses the methods of Bissell (1990) and ZSW to compute approximate confidence limits for C_{pk} for the variable Hardness in the data set Titanium (see Example 6.17).

```
proc capability data=Titanium noprint;
   var Hardness;
   specs lsl=0.8 usl=2.4;
   output out=Summary
       n    = n
       mean = mean
       std  = std
       lsl  = lsl
       usl  = usl
       cpk  = cpk
       cpklcl = cpklcl
       cpkucl = cpkucl
       cpl  = cpl
       cpu  = cpu ;

data Summary;
   set Summary;
   length Method $ 16;

   Method = "Bissell";
   lcl = cpklcl;
   ucl = cpkucl;
   output;

   * Assign confidence level;
   level = 0.95;
   aux   = probit( 1 - (1-level)/2 );

   Method = "ZSW Equation 6";
```

```
   zsw = log(0.5*n-0.5)
         + ( 2*(lgamma(0.5*n-1)-lgamma(0.5*n-0.5)) );
   zsw = sqrt((n-1)/(n-3)-exp(zsw));
   lcl = cpk*(1-aux*zsw);
   ucl = cpk*(1+aux*zsw);
   output;

   Method = "ZSW Equation 8";
   ds = 3*(cpu+cpl)/2;
   ms = 3*(cpl-cpu)/2;
   f1 = (1/3)*sqrt((n-1)/2)*gamma((n-2)/2)*(1/gamma((n-1)/2));
   f2 = sqrt(2/n)*(1/gamma(0.5))*exp(-n*0.5*ms*ms);
   f3 = ms*(1-(2*probnorm(-sqrt(n)*ms)));
   ex = f1*(ds-f2-f3);
   sd = ((n-1)/(9*(n-3)))*(ds**2-(2*ds*(f2+f3))+ms**2+(1/n));
   sd = sd-(ex*ex);
   sd = sqrt(sd);
   lcl = cpk-aux*sd;
   ucl = cpk+aux*sd;
   output;
run;

title "Approximate 95% Confidence Limits for Cpk";
proc print data = Summary noobs;
   var Method lcl cpk ucl;
run;
```

The results are shown in Output 6.18.1.

Output 6.18.1 Approximate Confidence Limits for C_{pk}

Approximate 95% Confidence Limits for Cpk

Method	lcl	cpk	ucl
Bissell	1.43845	1.80818	2.17790
ZSW Equation 6	1.43596	1.80818	2.18040
ZSW Equation 8	1.42419	1.80818	2.19217

Note that there is fairly close agreement in the three methods.

You can display the confidence limits computed using Bissell's approach on plots produced by the CAPABILITY procedure by specifying the keywords CPKLCL and CPKUCL in the INSET statement.

The following statements also compute an estimate of the index C_{pk} along with approximate limits by using the SPECIALINDICES option:

```
proc capability data=Titanium specialindices;
   var Hardness;
   specs lsl=0.8 usl=2.4;
run;
```

Output 6.18.2 Approximate Confidence Limits for C_{pk} using the SPECIALINDICES option

Approximate 95% Confidence Limits for Cpk

The CAPABILITY Procedure
Variable: Hardness (Hardness Measurement)

Process Capability Indices			
Index	**Value**	**95% Confidence Limits**	
Cp	2.005745	1.609575	2.401129
CPL	1.808179	1.438675	2.175864
CPU	2.203311	1.757916	2.646912
Cpk	1.808179	1.438454	2.177904

PPPLOT Statement: CAPABILITY Procedure

Overview: PPPLOT Statement

The PPPLOT statement creates a probability-probability plot (also referred to as a P-P plot or percent plot), which compares the empirical cumulative distribution function (ecdf) of a variable with a specified theoretical cumulative distribution function such as the normal. If the two distributions match, the points on the plot form a linear pattern that passes through the origin and has unit slope. Thus, you can use a P-P plot to determine how well a theoretical distribution models a set of measurements.

You can specify one of the following theoretical distributions with the PPPLOT statement:

- beta
- exponential
- gamma
- Gumbel
- inverse Gaussian
- lognormal
- normal
- generalized Pareto
- power function
- Rayleigh
- Weibull

You can use options in the PPPLOT statement to do the following:

- specify or estimate parameters for the theoretical distribution
- request graphical enhancements

You can also create a comparative P-P plot by using the PPPLOT statement in conjunction with a CLASS statement.

You have three alternatives for producing P-P plots with the PPPLOT statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: Probability-probability plots should not be confused with probability plots, which compare a set of ordered measurements with *percentiles* from a specified distribution. You can create probability plots with the PROBPLOT statement.

Getting Started: PPPLOT Statement

The following example illustrates the basic syntax of the PPPLOT statement. For complete details of the PPPLOT statement, see the section "Syntax: PPPLOT Statement" on page 442.

Creating a Normal Probability-Probability Plot

NOTE: See *Creating P-P Plots* in the SAS/QC Sample Library.

The distances between two holes cut into 50 steel sheets are measured and saved as values of the variable Distance in the following data set: [3]

[3] These data are also used to create Q-Q plots in "QQPLOT Statement: CAPABILITY Procedure" on page 493.

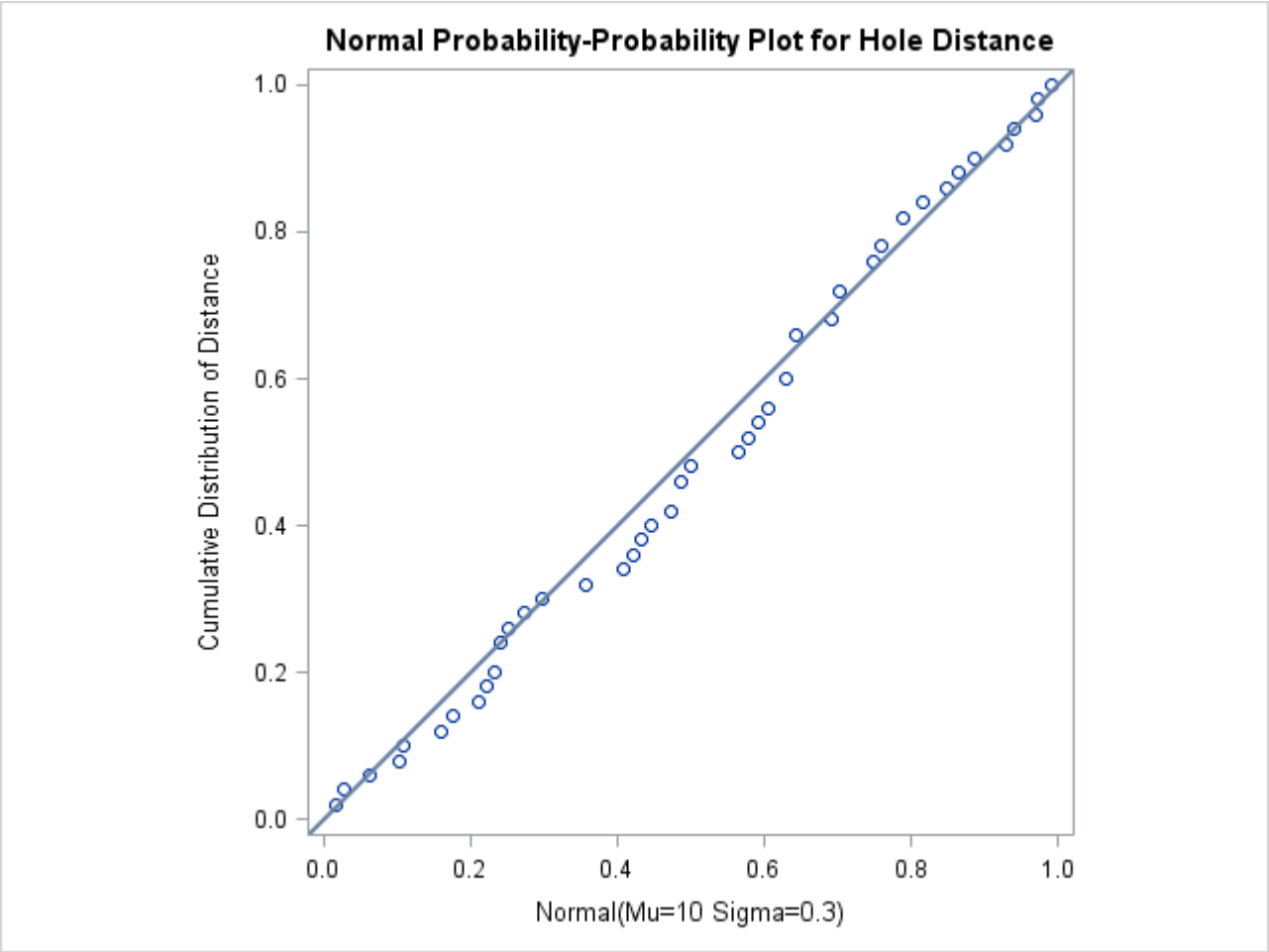
```
data Sheets;
   input Distance @@;
   label Distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;
```

The cutting process is in statistical control. As a preliminary step in a capability analysis of the process, it is decided to check whether the distances are normally distributed. The following statements create a P-P plot, shown in Figure 6.31, which is based on the normal distribution with mean $\mu = 10$ and standard deviation $\sigma = 0.3$:

```
title 'Normal Probability-Probability Plot for Hole Distance';
proc capability data=Sheets noprint;
   ppplot Distance / normal(mu=10 sigma=0.3)
                     square
                     odstitle=title;
run;
```

The NORMAL option in the PPPLOT statement requests a P-P plot based on the normal cumulative distribution function, and the MU= and SIGMA= *normal-options* specify μ and σ. Note that a P-P plot is always based on a *completely specified* distribution, in other words, a distribution with specific parameters. In this example, if you did not specify the MU= and SIGMA= *normal-options*, the sample mean and sample standard deviation would be used for μ and σ.

Figure 6.31 Normal P-P Plot with Diagonal Reference Line



The linearity of the pattern in Figure 6.31 is evidence that the measurements are normally distributed with mean 10 and standard deviation 0.3. The SQUARE option displays the plot in a square format.

Syntax: PPPLOT Statement

The syntax for the PPPLOT statement is as follows:

PPPLOT < *variables* > < / *options* > **;**

You can specify the keyword PP as an alias for PPPLOT, and you can use any number of PPPLOT statements in the CAPABILITY procedure. The components of the PPPLOT statement are described as follows.

variables

are the process variables for which to create P-P plots. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, the variables can be any numeric variables in the input data set. If you do not specify a list of variables, then by default, the procedure creates a P-P plot for each variable listed in the VAR statement or for each numeric variable in the input data set if you do not specify a VAR statement. For example, each of the following PPPLOT statements produces two P-P plots, one for length and one for width:

```
proc capability data=measures;
   var length width;
   ppplot;
run;

proc capability data=measures;
   ppplot length width;
run;
```

options

specify the theoretical distribution for the plot or add features to the plot. If you specify more than one variable, the options apply equally to each variable. Specify all options after the slash (/) in the PPPLOT statement. You can specify only one option naming a distribution, but you can specify any number of other options. The distributions available are the beta, exponential, gamma, Gumbel, inverse Gaussian, lognormal, normal, generalized Pareto, power function, Rayleigh, and Weibull. By default, the procedure produces a P-P plot based on the normal distribution.

In the following example, the NORMAL, MU= and SIGMA= options request a P-P plot based on the normal distribution with mean 10 and standard deviation 0.3. The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc capability data=measures;
   ppplot length width / normal(mu=10 sigma=0.3)
                         square
                         ctext=blue;
run;
```

Summary of Options

The following tables list the PPPLOT statement options by function. For complete descriptions, see the section "Dictionary of Options" on page 447.

Distribution Options

Table 6.53 summarizes the options for requesting a specific theoretical distribution.

Table 6.53 Options for Specifying a Theoretical Distribution

Option	Description
BETA(*beta-options*)	specifies beta P-P plot
EXPONENTIAL(*exponential-options*)	specifies exponential P-P plot
GAMMA(*gamma-options*)	specifies gamma P-P plot
GUMBEL(*Gumbel-options*)	specifies Gumbel P-P plot
IGAUSS(*iGauss-options*)	specifies inverse Gaussian P-P plot
LOGNORMAL(*lognormal-options*)	specifies lognormal P-P plot
NORMAL(*normal-options*)	specifies normal P-P plot
PARETO(*Pareto-options*)	specifies generalized Pareto P-P plot
POWER(*power-options*)	specifies power function P-P plot
RAYLEIGH(*Rayleigh-options*)	specifies Rayleigh P-P plot
WEIBULL(*Weibull-options*)	specifies Weibull P-P plot

Table 6.54 summarizes options that specify distribution parameters and control the display of the diagonal distribution reference line. Specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to request a normal P-P plot:

```
proc capability data=measures;
   ppplot length / normal(mu=10 sigma=0.3 color=red);
run;
```

The MU= and SIGMA= *normal-options* specify μ and σ for the normal distribution, and the COLOR= *normal-option* specifies the color for the line.

Table 6.54 Distribution Options

Option	Description
Distribution Reference Line Options	
COLOR=	specifies color of distribution reference line
L=	specifies line type of distribution reference line
NOLINE	suppresses the distribution reference line
SYMBOL=	specifies plotting character for line printer plots
W=	specifies width of distribution reference line
Beta-Options	
ALPHA=	specifies shape parameter α
BETA=	specifies shape parameter β
SIGMA=	specifies scale parameter σ
THETA=	specifies lower threshold parameter θ
Exponential-Options	
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ
Gamma-Options	
ALPHA=	specifies shape parameter α
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ

Table 6.54 (*continued*)

Option	Description
Gumbel-Options	
MU=	specifies location parameter μ
SIGMA=	specifies scale parameter σ
IGauss-Options	
LAMBDA=	specifies shape parameter λ
MU=	specifies mean μ
Lognormal-Options	
SIGMA=	specifies shape parameter σ
THETA=	specifies threshold parameter θ
ZETA=	specifies scale parameter ζ
Normal-Options	
MU=	specifies mean μ
SIGMA=	specifies standard deviation σ
Pareto-Options	
ALPHA=	specifies shape parameter α
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ
Power-Options	
ALPHA=	specifies shape parameter α
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ
Rayleigh-Options	
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ
Weibull-Options	
C=	specifies shape parameter c
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ

General Options

Table 6.55 lists options that control the appearance of the plots.

Table 6.55 General PPPLOT Statement Options

Option	Description
General Plot Layout Options	
CONTENTS=	specifies table of contents entry for P-P plot grouping
HREF=	specifies reference lines perpendicular to the horizontal axis
HREFLABELS=	specifies line labels for HREF= lines
NOFRAME	suppresses frame around plotting area
SQUARE	displays P-P plot in square format
VREF=	specifies reference lines perpendicular to the vertical axis

Table 6.55 (*continued*)

Option	Description
VREFLABELS=	specifies line labels for VREF= lines
Graphics Options	
ANNOTATE=	provides an annotate data set
CAXIS=	specifies color for axis
CFRAME=	specifies color for frame
CHREF=	specifies colors for HREF= lines
CTEXT=	specifies color for text
CVREF=	specifies colors for VREF= lines
DESCRIPTION=	specifies description for plot in graphics catalog
FONT=	specifies software font for text
HAXIS=	specifies AXIS statement for horizontal axis
HEIGHT=	specifies height of text used outside framed areas
HMINOR=	specifies number of minor tick marks on horizontal axis
HREFLABPOS=	specifies position for HREF= line labels
INFONT=	specifies software font for text inside framed areas
INHEIGHT=	specifies height of text inside framed areas
LHREF=	specifies line styles for HREF= lines
LVREF=	specifies line styles for VREF= lines
NAME=	specifies name for plot in graphics catalog
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
TURNVLABELS	turns and vertically strings out characters in labels for vertical axis
VAXIS=	specifies AXIS statement for vertical axis
VAXISLABEL=	specifies label for vertical axis
VMINOR=	specifies number of minor tick marks on vertical axis
VREFLABPOS=	specifies position for VREF= line labels
WAXIS=	specifies line thickness for axes and frame
Options for ODS Graphics Output	
ODSFOOTNOTE=	specifies footnote displayed on P-P plot
ODSFOOTNOTE2=	specifies secondary footnote displayed on P-P plot
ODSTITLE=	specifies title displayed on P-P plot
ODSTITLE2=	specifies secondary title displayed on P-P plot
Options for Comparative Plots	
ANNOKEY	applies annotation requested in ANNOTATE= data set to key cell only
CFRAMESIDE=	specifies color for filling row label frames
CFRAMETOP=	specifies color for filling column label frames
CPROP=	specifies color for proportion of frequency bar
CTEXTSIDE=	specifies color for row labels
CTEXTTOP=	specifies color for column labels
INTERTILE=	specifies distance between tiles in comparative plot
NCOLS=	specifies number of columns in comparative plot

Table 6.55 (*continued*)

Option	Description
NROWS=	specifies number of rows in comparative plot
OVERLAY	overlays plots for different class levels (ODS Graphics only)
Options for Line Printer Charts	
HREFCHAR=	specifies line character for HREF= lines
NOOBSLEGEND	suppresses legend for hidden points
PPSYMBOL=	specifies character for plotted points
VREFCHAR=	specifies line character for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the options specific to the PPPLOT statement. See "Dictionary of Common Options: CAPABILITY Procedure" on page 534 for detailed descriptions of options common to all the plot statements.

ALPHA=*value*

specifies the shape parameter α ($\alpha > 0$) for P-P plots requested with the BETA, GAMMA, PARETO, and POWER options. For examples, see the entries for the distribution options.

BETA<(*beta-options***)>**

creates a beta P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical beta cdf value

$$B_{\alpha\beta}\left(\frac{x_{(i)}-\theta}{\sigma}\right) = \int_{\theta}^{x_{(i)}} \frac{(t-\theta)^{\alpha-1}(\theta+\sigma-t)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}}dt$$

where $B_{\alpha\beta}(\cdot)$ is the normalized incomplete beta function, $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, and

θ = lower threshold parameter
σ = scale parameter ($\sigma > 0$)
α = first shape parameter ($\alpha > 0$)
β = second shape parameter ($\beta > 0$)

You can specify α, β, σ, and θ with the ALPHA=, BETA=, SIGMA=, and THETA= *beta-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / beta(theta=1 sigma=2 alpha=3 beta=4);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$, $\sigma = 1$, and maximum likelihood estimates are calculated for α and β.

IMPORTANT: If the default unit interval (0,1) does not adequately describe the range of your data, then you should specify THETA=θ and SIGMA=σ so that your data fall in the interval $(\theta, \theta + \sigma)$.

If the data are beta distributed with parameters α, β, σ, and θ, then the points on the plot for ALPHA=α, BETA=β, SIGMA=σ, and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified beta distribution is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

BETA=*value*

specifies the shape parameter β ($\beta > 0$) for P-P plots requested with the BETA distribution option. See the preceding entry for the BETA distribution option for an example.

C=*value*

specifies the shape parameter c ($c > 0$) for P-P plots requested with the WEIBULL option. See the entry for the WEIBULL option for examples.

EXPONENTIAL< (*exponential-options***) >**

EXP< (*exponential-options***) >**

creates an exponential P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical exponential cdf value

$$F(x_{(i)}) = 1 - \exp\left(-\frac{x_{(i)} - \theta}{\sigma}\right)$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)

You can specify σ and θ with the SIGMA= and THETA= *exponential-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / exponential(theta=1 sigma=2);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$ and a maximum likelihood estimate is calculated for σ.

IMPORTANT: Your data must be greater than or equal to the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, specify θ with the THETA= option.

If the data are exponentially distributed with parameters σ and θ, the points on the plot for SIGMA=σ and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified exponential distribution is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

GAMMA< (*gamma-options*) >

creates a gamma P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical gamma cdf value

$$G_\alpha\left(\frac{x_{(i)}-\theta}{\sigma}\right) = \int_\theta^{x_{(i)}} \frac{1}{\sigma\Gamma(\alpha)}\left(\frac{t-\theta}{\sigma}\right)^{\alpha-1} \exp\left(-\frac{t-\theta}{\sigma}\right) dt$$

where $G_\alpha(\cdot)$ is the normalized incomplete gamma function, and

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

You can specify α, σ, and θ with the ALPHA=, SIGMA=, and THETA= *gamma-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / gamma(alpha=1 sigma=2 theta=3);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$ and maximum likelihood estimates are calculated for α and σ.

IMPORTANT: Your data must be greater than or equal to the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, specify θ with the THETA= option.

If the data are gamma distributed with parameters α, σ, and θ, the points on the plot for ALPHA=α, SIGMA=σ, and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified gamma distribution is a good fit. You can specify the SHAPE= option as an alias for the ALPHA= option, the SCALE= option as an alias for the SIGMA= option, and the THRESHOLD= option as an alias for the THETA= option.

GUMBEL< (*Gumbel-options*) >

creates a Gumbel P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical Gumbel cdf value

$$F(x_{(i)}) = \exp\left(-e^{-(x_{(i)}-\mu)/\sigma}\right)$$

where

μ = location parameter
σ = scale parameter ($\sigma > 0$)

You can specify μ and σ with the MU= and SIGMA= *Gumbel-options*. By default, maximum likelihood estimates are computed for μ and σ.

If the data are Gumbel distributed with parameters μ and σ, the points on the plot for MU=μ and SIGMA=σ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified Gumbel distribution is a good fit.

IGAUSS< (*iGauss-options*) >

creates an inverse Gaussian P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical inverse Gaussian cdf value

$$F(x_{(i)}) = \Phi\left\{\sqrt{\frac{\lambda}{x_{(i)}}}\left(\frac{x_{(i)}}{\mu} - 1\right)\right\} + e^{2\lambda/\mu}\Phi\left\{-\sqrt{\frac{\lambda}{x_{(i)}}}\left(\frac{x_{(i)}}{\mu} + 1\right)\right\}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and

μ = mean parameter ($\mu > 0$)
λ = shape parameter ($\lambda > 0$)

You can specify known values for μ and λ with the MU= and LAMBDA= *iGauss-options*. By default, the sample mean is calculated for μ and a maximum likelihood estimate is computed for and λ.

If the data are inverse Gaussian distributed with parameters μ and λ, the points on the plot for MU=μ and LAMBDA=λ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified inverse Gaussian distribution is a good fit.

LAMBDA=*value*

specifies the shape parameter λ ($\lambda > 0$) for P-P plots requested with the IGAUSS option. Enclose the LAMBDA= option in parentheses after the IGAUSS distribution keyword. If you do not specify a value for λ, the procedure calculates a maximum likelihood estimate.

LOGNORMAL< (*lognormal-options*) >

LNORM< (*lognormal-options*) >

creates a lognormal P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical lognormal cdf value

$\Phi\left(\frac{\log(x_{(i)}-\theta)-\zeta}{\sigma}\right)$

where $\Phi(\cdot)$ is the cumulative standard normal distribution function, and

θ = threshold parameter

ζ = scale parameter
σ = shape parameter ($\sigma > 0$)

You can specify θ, ζ, and σ with the THETA=, ZETA=, and SIGMA= *lognormal-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / lognormal(theta=1 zeta=2);
run;
```

If you do not specify values for these parameters, then by default, $\theta = 0$ and maximum likelihood estimates are calculated for σ and ζ.

IMPORTANT: Your data must be greater than the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, specify θ with the THETA= option.

If the data are lognormally distributed with parameters σ, θ, and ζ, the points on the plot for SIGMA=σ, THETA=θ, and ZETA=ζ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified lognormal distribution is a good fit. You can specify the SHAPE= option as an alias for the SIGMA= option, the SCALE= option as an alias for the ZETA= option, and the THRESHOLD= option as an alias for the THETA= option.

MU=*value*
specifies the parameter μ for a P-P plot requested with the GUMBEL, IGAUSS, and NORMAL options. For examples, see Figure 6.31, or Figure 6.32 and Figure 6.33. For the normal and inverse Gaussian distributions, the default value of μ is the sample mean. If you do not specify a value for μ for the Gumbel distribution, the procedure calculates a maximum likelihood estimate.

NOLINE
suppresses the diagonal reference line.

NOOBSLEGEND
NOOBSL
suppresses the legend that indicates the number of hidden observations in a legacy line printer plot. This option is ignored unless you specify the LINEPRINTER option in the PROC CAPABILITY statement.

NORMAL<(*normal-options***)>**
NORM<(*normal-options* **)>**
creates a normal P-P plot. By default, if you do not specify a distribution option, the procedure displays a normal P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical normal cdf value

$$\Phi\left(\frac{x_{(i)}-\mu}{\sigma}\right) = \int_{-\infty}^{x_{(i)}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution function, and

μ = location parameter or mean
σ = scale parameter or standard deviation ($\sigma > 0$)

You can specify μ and σ with the MU= and SIGMA= *normal-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / normal(mu=1 sigma=2);
run;
```

By default, the sample mean and sample standard deviation are used for μ and σ.

If the data are normally distributed with parameters μ and σ, the points on the plot for MU=μ and SIGMA=σ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified normal distribution is a good fit. For an example, see Figure 6.31.

PARETO< (*Pareto-options*) >
creates a generalized Pareto P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical generalized Pareto cdf value

$$F(x_{(i)}) = 1 - \left(1 - \frac{\alpha(x_{(i)} - \theta)}{\sigma}\right)^{\frac{1}{\alpha}}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter

The parameter θ for the generalized Pareto distribution must be less than the minimum data value. You can specify θ with the THETA= *Pareto-option*. The default value for θ is 0. In addition, the generalized Pareto distribution has a shape parameter α and a scale parameter σ. You can specify these parameters with the ALPHA= and SIGMA= *Pareto-options*. By default, maximum likelihood estimates are computed for α and σ.

If the data are generalized Pareto distributed with parameters θ, σ, and α, the points on the plot for THETA=θ, SIGMA=σ, and ALPHA=α tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified generalized Pareto distribution is a good fit.

POWER< (*power-options*) >
creates a power function P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical power function cdf value

$$F(x_{(i)}) = \left(\frac{x_{(i)} - \theta}{\sigma}\right)^{\alpha}$$

where

θ = lower threshold parameter (lower endpoint)
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

The power function distribution is bounded below by the parameter θ and above by the value $\theta + \sigma$. You can specify θ and σ by using the THETA= and SIGMA= *power-options*. The default values for θ and σ are 0 and 1, respectively.

You can specify a value for the shape parameter, α, with the ALPHA= *power-option*. If you do not specify a value for α, the procedure calculates a maximum likelihood estimate.

The power function distribution is a special case of the beta distribution with its second shape parameter, $\beta = 1$.

If the data are power function distributed with parameters θ, σ, and α, the points on the plot for THETA=θ, SIGMA=σ, and ALPHA=α tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified power function distribution is a good fit.

PPSYMBOL='*character*'
specifies the character used to plot the points in a legacy line printer plot. The default is the plus sign (+). This option is ignored unless you specify the LINEPRINTER option in the PROC CAPABILITY statement.

RAYLEIGH< (*Rayleigh-options*) **>**
creates a Rayleigh P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical Rayleigh cdf value

$$F(x_{(i)}) = 1 - e^{-(x_{(i)}-\theta)^2/(2\sigma^2)}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)

The parameter θ for the Rayleigh distribution must be less than the minimum data value. You can specify θ with the THETA= *Rayleigh-option*. The default value for θ is 0. You can specify σ with the SIGMA= *Rayleigh-option*. By default, a maximum likelihood estimate is computed for σ.

If the data are Rayleigh distributed with parameters θ and σ, the points on the plot for THETA=θ and SIGMA=σ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified Rayleigh distribution is a good fit.

SIGMA=*value*

specifies the parameter σ, where $\sigma > 0$. When used with the BETA, EXPONENTIAL, GAMMA, GUMBEL, NORMAL, PARETO, POWER, RAYLEIGH, and WEIBULL options, the SIGMA= option specifies the scale parameter. When used with the LOGNORMAL option, the SIGMA= option specifies the shape parameter. Enclose the SIGMA= option in parentheses after the distribution keyword. For an example of the SIGMA= option used with the NORMAL option, see Figure 6.31.

SQUARE

displays the P-P plot in a square frame. The default is a rectangular frame. See Figure 6.31 for an example.

SYMBOL='*character*'

specifies the character used for the diagonal reference line in legacy line printer plots. The default character is the first letter of the distribution option keyword. This option is ignored unless you specify the LINEPRINTER option in the PROC CAPABILITY statement.

THETA=*value*

THRESHOLD=*value*

specifies the lower threshold parameter θ for plots requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, PARETO, POWER, RAYLEIGH, and WEIBULL options.

WEIBULL< (*Weibull-options***) >**

WEIB< (*Weibull-options***) >**

creates a Weibull P-P plot. To create the plot, the n nonmissing observations are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

The y-coordinate of the ith point is the empirical cdf value $\frac{i}{n}$. The x-coordinate is the theoretical Weibull cdf value

$$F(x_{(i)}) = 1 - \exp\left(-\left(\frac{x_{(i)}-\theta}{\sigma}\right)^c\right)$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)

You can specify c, σ, and θ with the C=, SIGMA=, and THETA= *Weibull-options*, as illustrated in the following example:

```
proc capability data=measures;
   ppplot width / weibull(theta=1 sigma=2);
run;
```

If you do not specify values for these parameters, then by default $\theta = 0$ and maximum likelihood estimates are calculated for σ and c.

IMPORTANT: Your data must be greater than or equal to the lower threshold θ. If the default $\theta = 0$ is not an adequate lower bound for your data, you should specify θ with the THETA= option.

If the data are Weibull distributed with parameters c, σ, and θ, the points on the plot for C=c, SIGMA=σ, and THETA=θ tend to fall on or near the diagonal line $y = x$, which is displayed by default. Agreement between the diagonal line and the point pattern is evidence that the specified Weibull distribution is a good fit. You can specify the SHAPE= option as an alias for the C= option, the SCALE= option as an alias for the SIGMA= option, and the THRESHOLD= option as an alias for the THETA= option.

ZETA=*value*

specifies a value for the scale parameter ζ for lognormal P-P plots requested with the LOGNORMAL option.

Details: PPPLOT Statement

This section provides details on the following topics:

- construction and interpretation of P-P plots
- comparison of P-P plots with Q-Q plots
- distributions supported by the PPPLOT statement
- graphical enhancements of P-P plots

Construction and Interpretation of P-P Plots

A P-P plot compares the empirical cumulative distribution function (ecdf) of a variable with a specified theoretical cumulative distribution function $F(\cdot)$. The ecdf, denoted by $F_n(x)$, is defined as the proportion of nonmissing observations less than or equal to x, so that $F_n(x_{(i)}) = \frac{i}{n}$.

To construct a P-P plot, the n nonmissing values are first sorted in increasing order:

$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$

Then the ith ordered value $x_{(i)}$ is represented on the plot by the point whose x-coordinate is $F(x_{(i)})$ and whose y-coordinate is $\frac{i}{n}$.

Like Q-Q plots and probability plots, P-P plots can be used to determine how well a theoretical distribution models a data distribution. If the theoretical cdf reasonably models the ecdf in all respects, including location and scale, the point pattern on the P-P plot is linear through the origin and has unit slope.

NOTE: See *Interpreting P-P Plots* in the SAS/QC Sample Library.

Unlike Q-Q and probability plots, P-P plots are not invariant to changes in location and scale. For example, the data in the section "Getting Started: PPPLOT Statement" on page 440 are reasonably described by a normal distribution with mean 10 and standard deviation 0.3. It is instructive to display these data on normal P-P plots with a different mean and standard deviation, as created by the following statements:

```
data Sheets;
   input Distance @@;
   label Distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;


proc capability data=Sheets noprint;
   ppplot Distance / normal(mu=9.5 sigma=0.3) square;
   ppplot Distance / normal(mu=10 sigma=0.5) square;
run;
```

The ODS GRAPHICS ON statement specified before the PROC CAPABILITY statement enables ODS Graphics, so the P-P plots are created using ODS Graphics instead of traditional graphics. The resulting plots are shown in Figure 6.32 and Figure 6.33.

Figure 6.32 Normal P-P Plot with Mean Specified Incorrectly

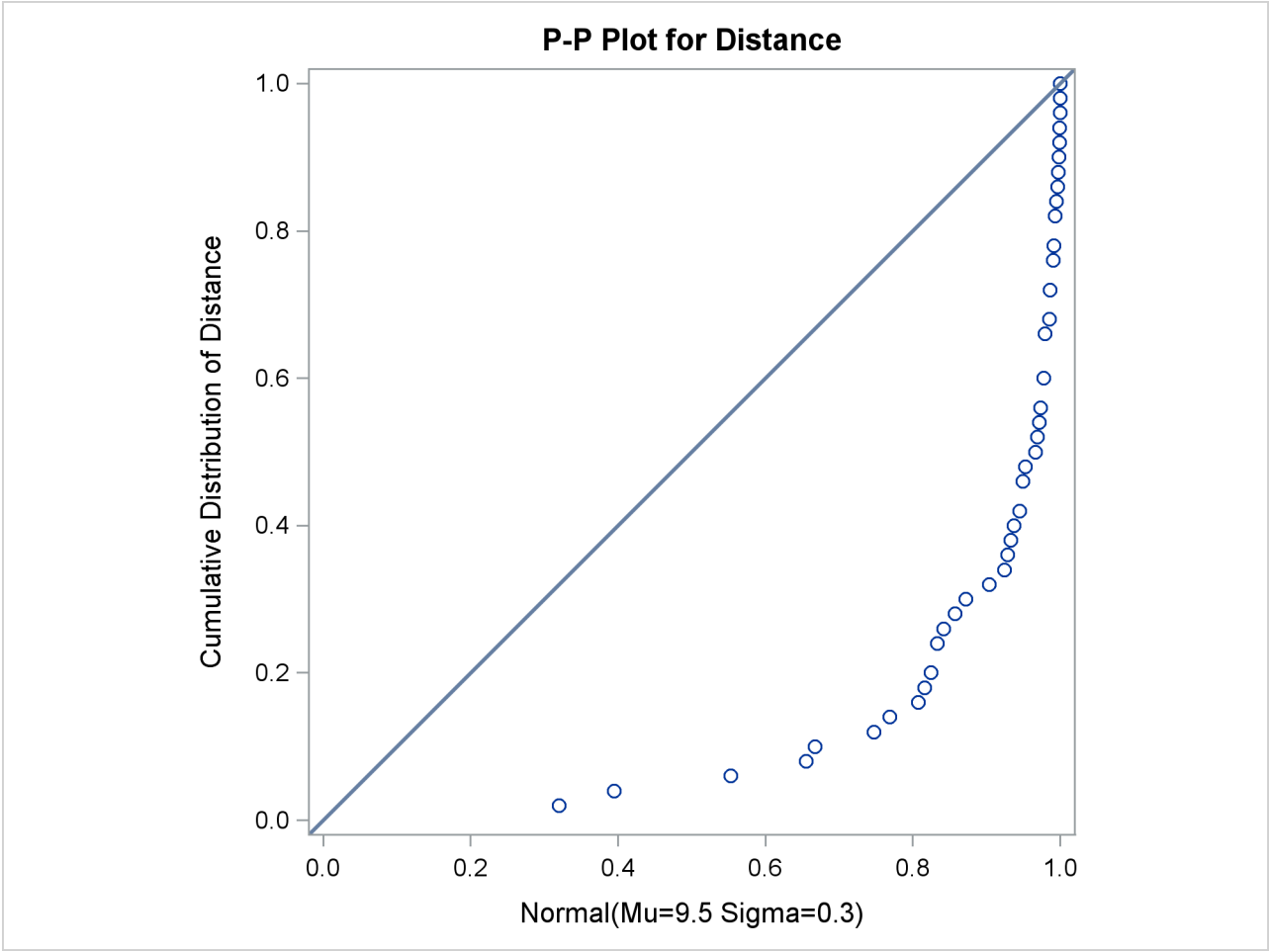
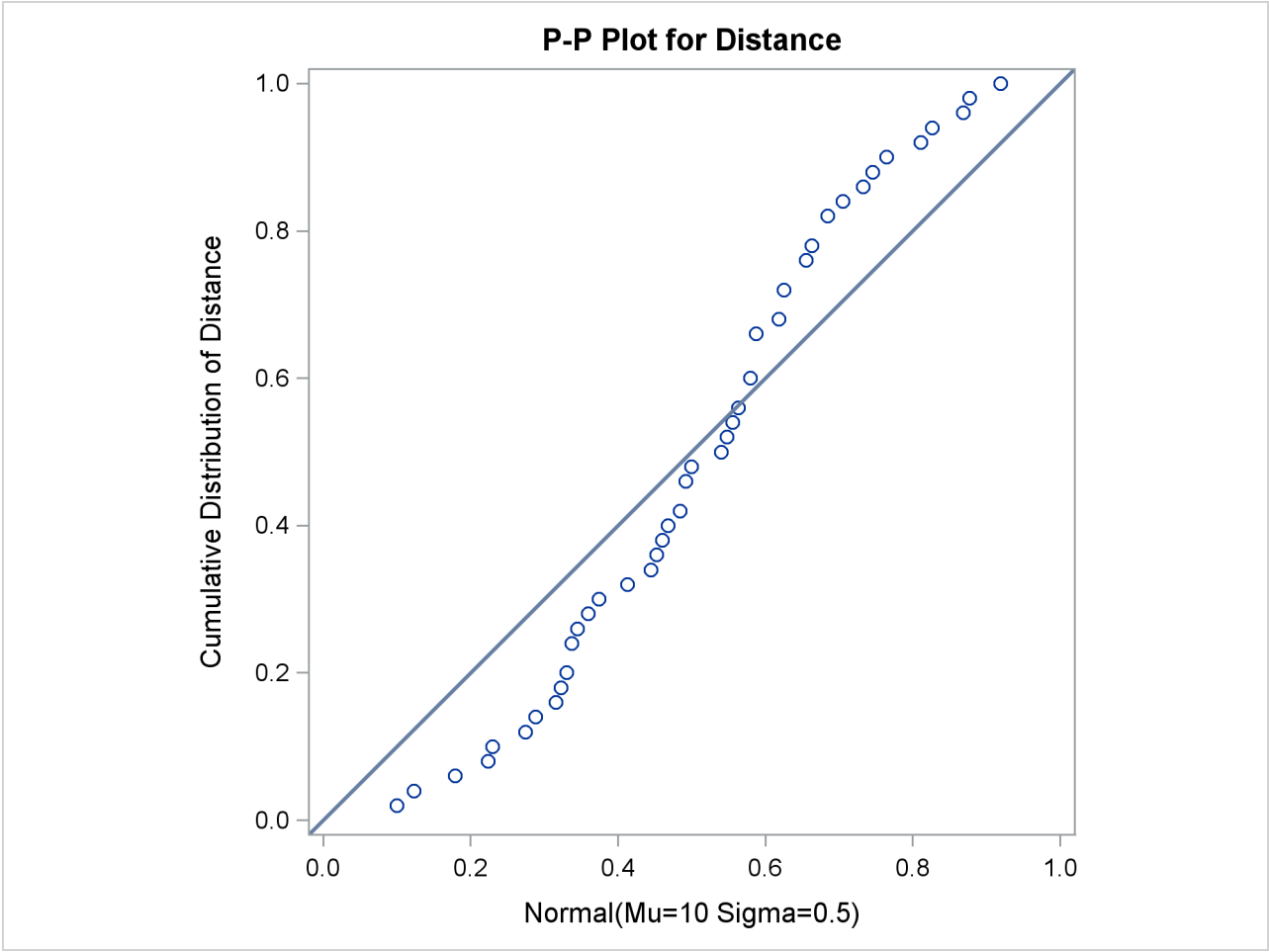


Figure 6.33 Normal P-P Plot with Standard Deviation Specified Incorrectly



Specifying a mean of 9.5 instead of 10 results in the plot shown in Figure 6.32, while specifying a standard deviation of 0.5 instead of 0.3 results in the plot shown in Figure 6.33. Both plots clearly reveal the model misspecification.

Comparison of P-P Plots and Q-Q Plots

A P-P plot compares the empirical cumulative distribution function of a data set with a specified theoretical cumulative distribution function $F(\cdot)$. A Q-Q plot compares the quantiles of a data distribution with the quantiles of a standardized theoretical distribution from a specified family of distributions. There are three important differences in the way P-P plots and Q-Q plots are constructed and interpreted:

- The construction of a Q-Q plot does not require that the location or scale parameters of $F(\cdot)$ be specified. The theoretical quantiles are computed from a standard distribution within the specified family. A linear point pattern indicates that the specified family reasonably describes the data distribution, and the location and scale parameters can be estimated visually as the intercept and slope of the linear pattern. In contrast, the construction of a P-P plot requires the location and scale parameters of $F(\cdot)$ to evaluate the cdf at the ordered data values.

- The linearity of the point pattern on a Q-Q plot is unaffected by changes in location or scale. On a P-P plot, changes in location or scale do not necessarily preserve linearity.
- On a Q-Q plot, the reference line representing a particular theoretical distribution depends on the location and scale parameters of that distribution, having intercept and slope equal to the location and scale parameters. On a P-P plot, the reference line for any distribution is always the diagonal line $y = x$.

Consequently, you should use a Q-Q plot if your objective is to compare the data distribution with a family of distributions that vary only in location and scale, particularly if you want to estimate the location and scale parameters from the plot.

An advantage of P-P plots is that they are discriminating in regions of high probability density, because in these regions the empirical and theoretical cumulative distributions change more rapidly than in regions of low probability density. For example, if you compare a data distribution with a particular normal distribution, differences in the middle of the two distributions are more apparent on a P-P plot than on a Q-Q plot.

For further details on P-P plots, refer to Gnanadesikan (1997) and Wilk and Gnanadesikan (1968).

Summary of Theoretical Distributions

You can use the PPPLOT statement to request P-P plots based on the theoretical distributions summarized in the following table:

Table 6.56 Distributions and Parameters

Family	Distribution Function $F(x)$	Range	Parameters		
			Location	Scale	Shape
Beta	$\int_{\theta}^{x} \frac{(t-\theta)^{\alpha-1}(\theta+\sigma-t)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} dt$	$\theta < x < \theta+\sigma$	θ	σ	α, β
Exponential	$1 - \exp\left(-\frac{x-\theta}{\sigma}\right)$	$x \geq \theta$	θ	σ	
Gamma	$\int_{\theta}^{x} \frac{1}{\sigma\Gamma(\alpha)} \left(\frac{t-\theta}{\sigma}\right)^{\alpha-1} \exp\left(-\frac{t-\theta}{\sigma}\right) dt$	$x > \theta$	θ	σ	α
Gumbel	$\exp\left(-e^{(x-\mu)/\sigma}\right)$	all x	μ	σ	
Inverse Gaussian	$\Phi\left\{\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu}-1\right)\right\} + e^{2\lambda/\mu}\Phi\left\{-\sqrt{\frac{\lambda}{x}}\left(\frac{x}{\mu}+1\right)\right\}$	$x > 0$	μ		λ
Lognormal	$\int_{\theta}^{x} \frac{1}{\sigma\sqrt{2\pi}(t-\theta)} \exp\left(-\frac{(\log(t-\theta)-\zeta)^2}{2\sigma^2}\right) dt$	$x > \theta$	θ	ζ	σ
Normal	$\int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) dt$	all x	μ	σ	
Generalized Pareto	$1 - \left(1 - \frac{\alpha(x-\theta)}{\sigma}\right)^{1/\alpha}$	all x	θ	σ	α
Power Function	$\left(\frac{x-\theta}{\sigma}\right)^{\alpha}$	$\theta < x < \theta+\sigma$	θ	σ	α

Table 6.56 (*continued*)

Family	Distribution Function $F(x)$	Range	Parameters		
			Location	Scale	Shape
Rayleigh	$1 - e^{-(x-\theta)^2/(2\sigma^2)}$	$x \geq \theta$	θ	σ	
Weibull	$1 - \exp\left(-\left(\frac{x-\theta}{\sigma}\right)^c\right)$	$x > \theta$	θ	σ	c

You can request these distributions with the BETA, EXPONENTIAL, GAMMA, GUMBEL, IGAUSS, NORMAL, LOGNORMAL, PARETO, POWER, RAYLEIGH, and WEIBULL options, respectively. If you do not specify a distribution option, a normal P-P plot is created.

To create a P-P plot, you must provide all of the parameters for the theoretical distribution. If you do not specify parameters, then default values or estimates are substituted, as summarized by the following table:

Table 6.57 Defaults for Parameters

Family	Default Values	Estimated Values
Beta	$\theta = 0, \sigma = 1$	maximum likelihood estimates for α and β
Exponential	$\theta = 0$	maximum likelihood estimate for σ
Gamma	$\theta = 0$	maximum likelihood estimates for σ and α
Gumbel	None	maximum likelihood estimates for μ and σ
Inverse Gaussian	None	sample estimate for μ, maximum likelihood estimate for λ
Lognormal	$\theta = 0$	maximum likelihood estimates for σ and ζ
Normal	None	sample estimates for μ and σ
Generalized Pareto	$\theta = 0$	maximum likelihood estimates for σ and α
Power Function	$\theta = 0, \sigma = 1$	maximum likelihood estimate for α
Rayleigh	$\theta = 0$	maximum likelihood estimate for σ
Weibull	$\theta = 0$	maximum likelihood estimates for σ and c

Specification of Symbol Markers

If you produce traditional graphics, you can use options in the SYMBOL1 statement to specify the appearance of the symbol marker for the points. The V= option specifies the symbol, the C= option specifies the color, and the H= option specifies the height. Refer to *SAS/GRAPH: Reference* for details concerning these options. If you produce a line printer plot, you can use the PPSYMBOL= option in the PPPLOT statement to specify the character used to plot the points.

Specification of the Distribution Reference Line

If you produce traditional graphics, you can control the color, type, and width of the diagonal distribution reference line by specifying the COLOR=, L=, and W= options in parentheses after the distribution option in the PPPLOT statement. Alternatively, you can control these features with the C=, L=, and W= options in the SYMBOL4 statement. Refer to *SAS/GRAPH: Reference* for details concerning these options. If you produce

a line printer plot, you can specify the character used for the line with the SYMBOL= option enclosed in parentheses after the distribution option in the PPPLOT statement.

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. PPPLOT options used to control the appearance of traditional graphics are ignored for ODS Graphics output.

When ODS Graphics is in effect, the PPPLOT statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 6.58.

Table 6.58 ODS Graphics Produced by the PPPLOT Statement

ODS Graph Name	Plot Description
PPPlot	P-P plot

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

PROBPLOT Statement: CAPABILITY Procedure

Overview: PROBPLOT Statement

The PROBPLOT statement creates a probability plot, which compares ordered values of a variable with percentiles of a specified theoretical distribution such as the normal. If the data distribution matches the theoretical distribution, the points on the plot form a linear pattern. Thus, you can use a probability plot to determine how well a theoretical distribution models a set of measurements.

You can specify one of the following theoretical distributions with the PROBPLOT statement:

- beta
- exponential
- gamma
- Gumbel
- three-parameter lognormal
- normal

- generalized Pareto
- power function
- Rayleigh
- two-parameter Weibull
- three-parameter Weibull

You can use options in the PROBPLOT statement to do the following:

- specify or estimate shape parameters for the theoretical distribution
- display a reference line corresponding to specified or estimated location and scale parameters for the theoretical distribution
- request graphical enhancements

You can also create a comparative probability plot by using the PROBPLOT statement in conjunction with a CLASS statement.

You have three alternatives for producing probability plots the PROBPLOT statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: Probability plots are similar to Q-Q plots, which you can create with the QQPLOT statement (see "QQPLOT Statement: CAPABILITY Procedure" on page 493). Probability plots are preferable for graphical estimation of percentiles, whereas Q-Q plots are preferable for graphical estimation of distribution parameters and capability indices.

Getting Started: PROBPLOT Statement

The following examples illustrate the basic syntax of the PROBPLOT statement. For complete details of the PROBPLOT statement, see the section "Syntax: PROBPLOT Statement" on page 468. Advanced examples are provided on the section "Examples: PROBPLOT Statement" on page 490.

Creating a Normal Probability Plot

NOTE: See *Creating a Normal Probability Plot* in the SAS/QC Sample Library.

The diameters of 50 steel rods are measured and saved as values of the variable Diameter in the following data set:[4]

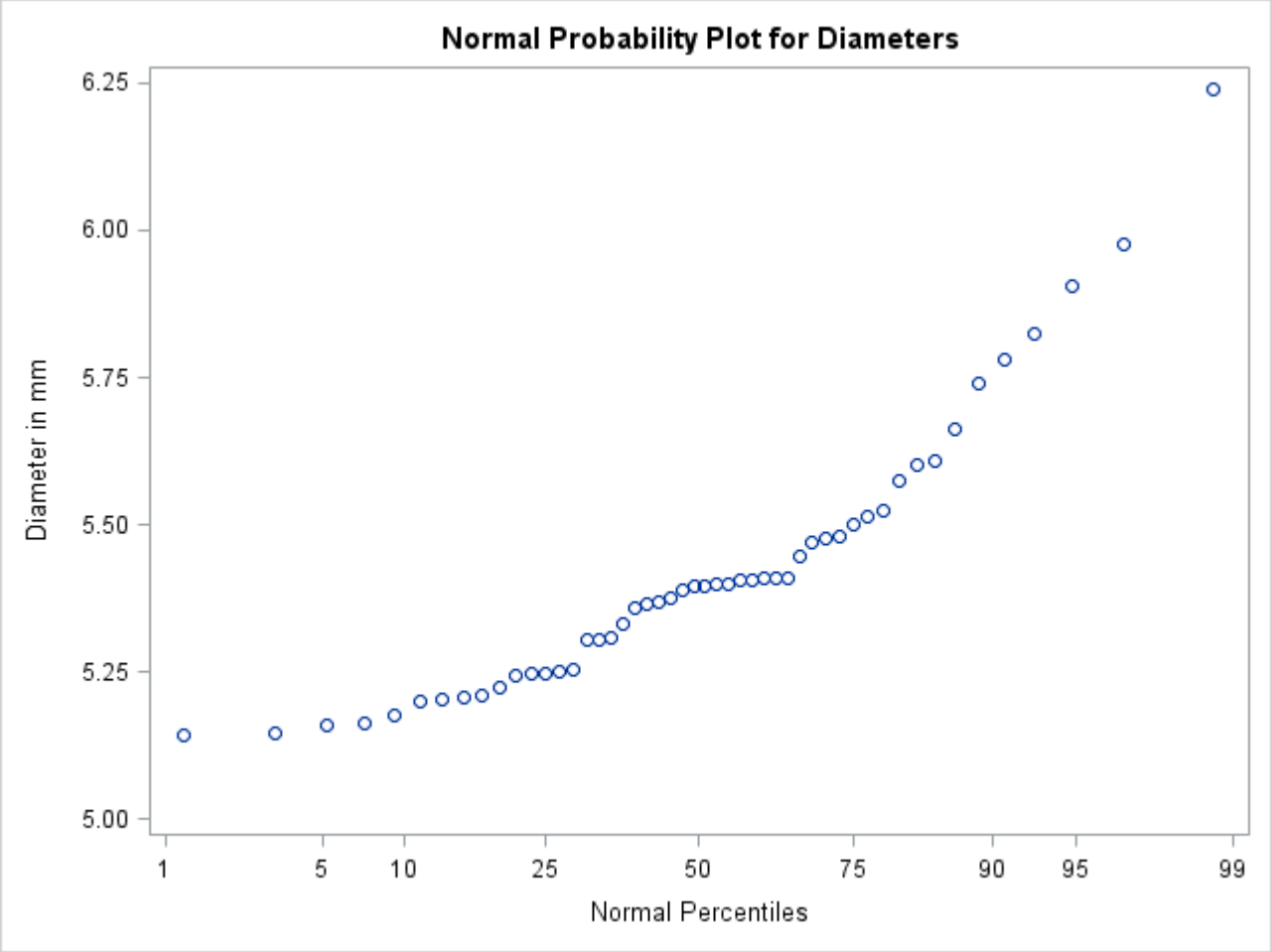
```
data Rods;
   input Diameter @@;
   label Diameter='Diameter in mm';
   datalines;
5.501  5.251  5.404  5.366  5.445
5.576  5.607  5.200  5.977  5.177
5.332  5.399  5.661  5.512  5.252
5.404  5.739  5.525  5.160  5.410
5.823  5.376  5.202  5.470  5.410
5.394  5.146  5.244  5.309  5.480
5.388  5.399  5.360  5.368  5.394
5.248  5.409  5.304  6.239  5.781
5.247  5.907  5.208  5.143  5.304
5.603  5.164  5.209  5.475  5.223
;
```

The process producing the rods is in statistical control, and as a preliminary step in a capability analysis of the process, you decide to check whether the diameters are normally distributed. The following statements create the normal probability plot shown in Figure 6.34:

```
title 'Normal Probability Plot for Diameters';
proc capability data=Rods noprint;
   probplot Diameter / odstitle=title;
run;
```

[4] This data set is analyzed using quantile-quantile plots in Example 6.21 and Example 6.22.

Figure 6.34 Normal Probability Plot Created with Traditional Graphics



Note that the PROBPLOT statement creates a normal probability plot for Diameter by default.

The nonlinearity of the point pattern indicates a departure from normality. Because the point pattern is curved with slope increasing from left to right, a theoretical distribution that is skewed to the right, such as a lognormal distribution, should provide a better fit than the normal distribution. This possibility is explored in the next example.

Creating Lognormal Probability Plots

NOTE: See *Creating Lognormal Probability Plots* in the SAS/QC Sample Library.

When you request a lognormal probability plot, you must specify the shape parameter σ for the lognormal distribution (see Table 6.62 for the equation). The value of σ must be positive, and typical values of σ range from 0.1 to 1.0. Alternatively, you can specify that σ is to be estimated from the data.

The following statements illustrate the first approach by creating a series of three lognormal probability plots for the variable Diameter introduced in the preceding example:

```
proc capability data=Rods noprint;
   probplot Diameter / lognormal(sigma=0.2 0.5 0.8)
                       href = 95
                       square;
run;
```

The LOGNORMAL option requests plots based on the lognormal family of distributions, and the SIGMA= option requests plots for σ equal to 0.2, 0.5, and 0.8. The SQUARE option displays the probability plot in a square format and the HREF= option requests a reference line at the 95th percentile.

The resulting plots are displayed in Figure 6.35, Figure 6.36, and Figure 6.37, respectively. The value $\sigma = 0.5$ in Figure 6.36 produces the most linear pattern.

Figure 6.35 Probability Plot Based on Lognormal Distribution with $\sigma = 0.2$

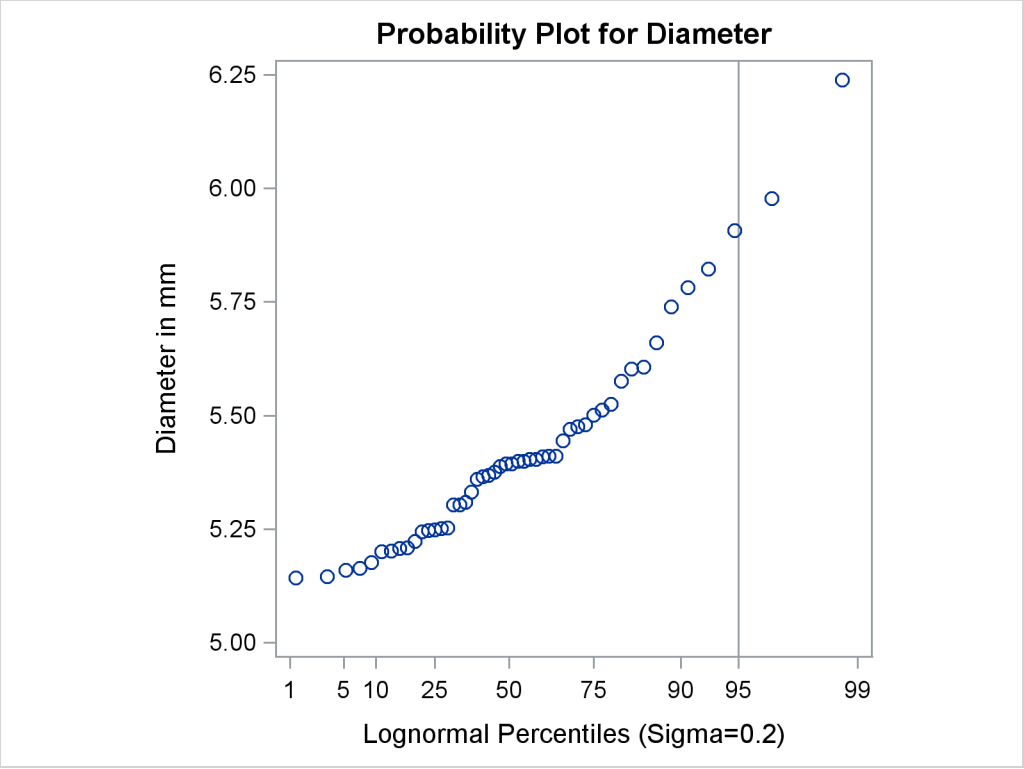


Figure 6.36 Probability Plot Based on Lognormal Distribution with $\sigma = 0.5$

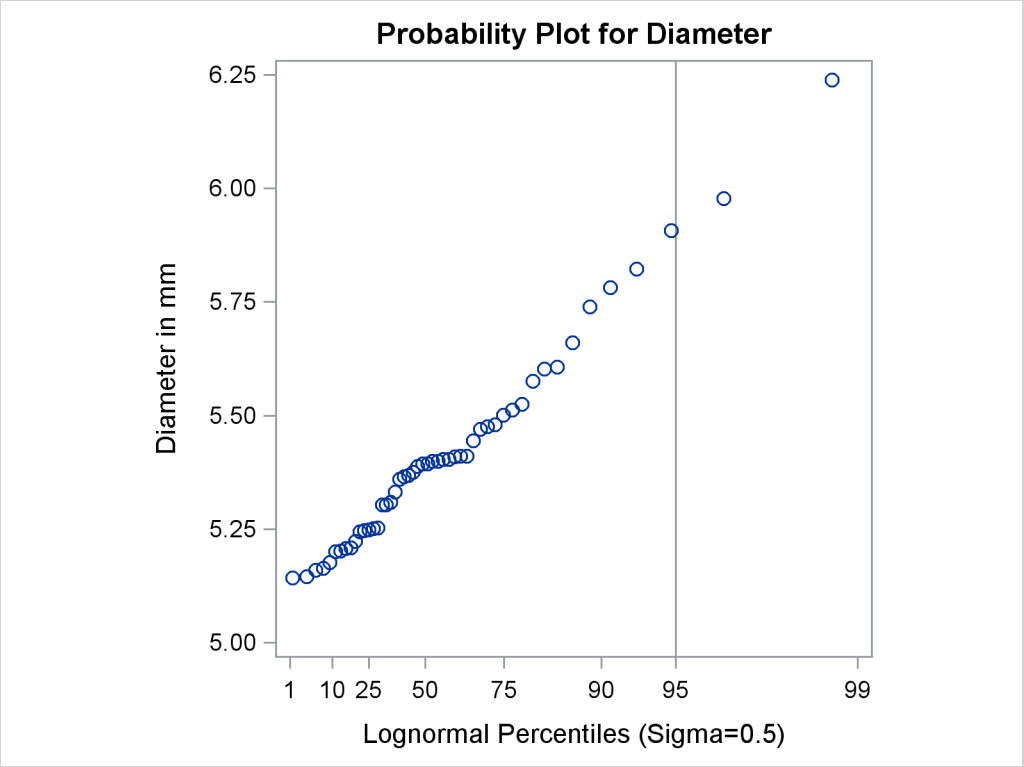
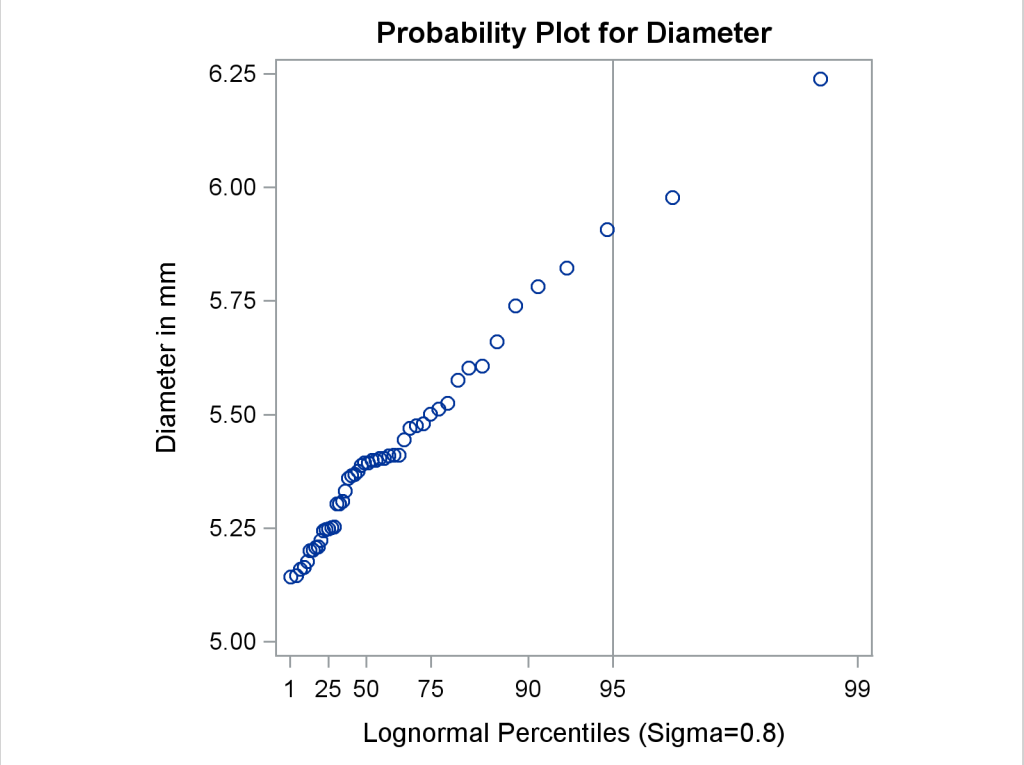


Figure 6.37 Probability Plot Based on Lognormal Distribution with $\sigma = 0.8$



Based on Figure 6.36, the 95th percentile of the diameter distribution is approximately 5.9 mm, because this is the value corresponding to the intersection of the point pattern with the reference line.

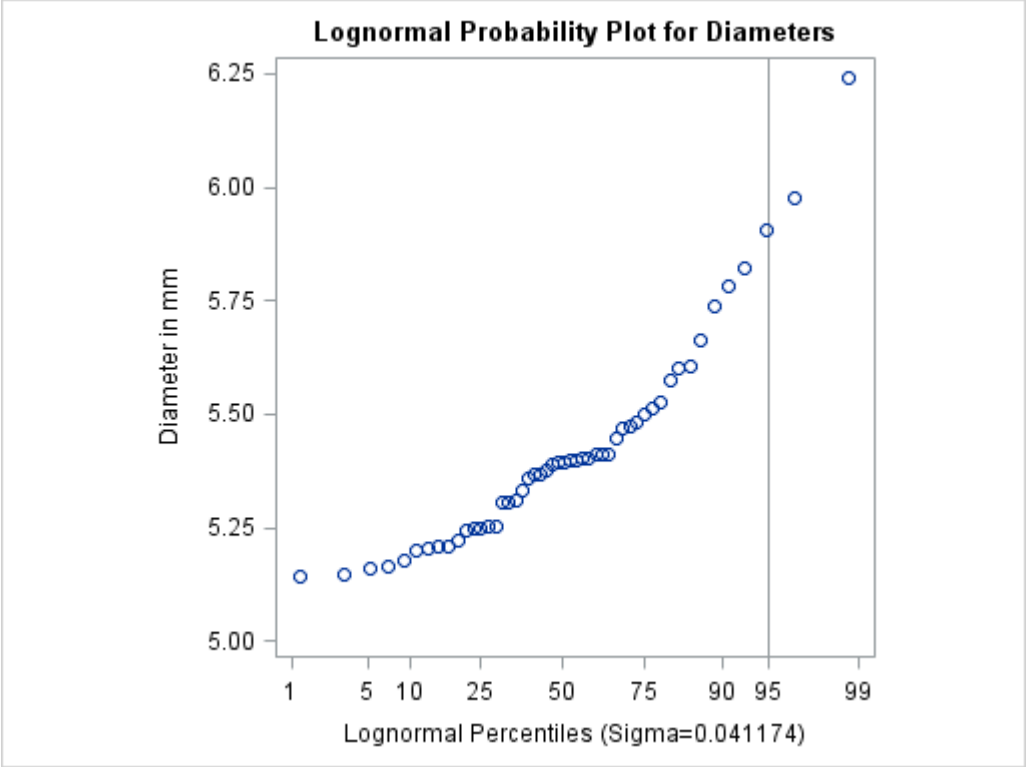
The following statements illustrate how you can create a lognormal probability plot for Diameter using a local maximum likelihood estimate for σ.

```
title 'Lognormal Probability Plot for Diameters';
proc capability data=Rods noprint;
   probplot Diameter / lognormal(sigma=est)
                       href     = 95
                       odstitle = title
                       square;
run;
```

The plot is displayed in Figure 6.38.

Note that the maximum likelihood estimate of σ (in this case 0.041) does not necessarily produce the most linear point pattern. This example is continued in Example 6.20.

Figure 6.38 Probability Plot Based on Lognormal Distribution with Estimated σ



Syntax: PROBPLOT Statement

The syntax for the PROBPLOT statement is as follows:

PROBPLOT < *variables* > < / *options* > **;**

You can specify the keyword PROB as an alias for PROBPLOT, and you can use any number of PROBPLOT statements in the CAPABILITY procedure. The components of the PROBPLOT statement are described as follows.

variables

are the process variables for which to create probability plots. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, the variables can be any numeric variables in the input data set. If you do not specify a list of variables, then by default the procedure creates a probability plot for each variable listed in the VAR statement, or for each numeric variable in the DATA= data set if you do not specify a VAR statement. For example, each of the following PROBPLOT statements produces two probability plots, one for length and one for width:

```
proc capability data=measures;
   var length width;
   probplot;
run;

proc capability data=measures;
   probplot length width;
run;
```

options

specify the theoretical distribution for the plot or add features to the plot. If you specify more than one variable, the options apply equally to each variable. Specify all options after the slash (/) in the PROBPLOT statement. You can specify only one option naming the distribution in each PROBPLOT statement, but you can specify any number of other options. The distributions available are the beta, exponential, gamma, Gumbel, lognormal, normal, generalized Pareto, power function, Rayleigh, two-parameter Weibull, and three-parameter Weibull. By default, the procedure produces a plot for the normal distribution.

In the following example, the NORMAL option requests a normal probability plot for each variable, while the MU= and SIGMA= *normal-options* request a distribution reference line corresponding to the normal distribution with $\mu = 10$ and $\sigma = 0.3$. The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc capability data=measures;
   probplot length1 length2 / normal(mu=10 sigma=0.3)
                              square
                              ctext=blue;
run;
```

Summary of Options

The following tables list the PROBPLOT statement options by function. For complete descriptions, see the section "Dictionary of Options" on page 473.

Distribution Options

Table 6.59 summarizes the options for requesting a specific theoretical distribution.

Table 6.59 Options for Specifying a Theoretical Distribution

Option	Description
BETA(*beta-options*)	specifies beta probability plot for shape parameters α, β specified with mandatory ALPHA= and BETA= *beta-options*
EXPONENTIAL(*exponential-options*)	specifies exponential probability plot
GAMMA(*gamma-options*)	specifies gamma probability plot for shape parameter α specified with mandatory ALPHA= *gamma-option*
GUMBEL(*Gumbel-options*)	specifies Gumbel probability plot
LOGNORMAL(*lognormal-options*)	specifies lognormal probability plot for shape parameter σ specified with mandatory SIGMA= *lognormal-option*
NORMAL(*normal-options*)	specifies normal probability plot
PARETO(*Pareto-options*)	specifies generalized Pareto probability plot for shape parameter α specified with mandatory ALPHA= *Pareto-option*
POWER(*power-options*)	specifies power function probability plot for shape parameter α specified with mandatory ALPHA= *power-option*
RAYLEIGH(*Rayleigh-options*)	specifies Rayleigh probability plot
WEIBULL(*Weibull-options*)	specifies three-parameter Weibull probability plot for shape parameter c specified with mandatory C= *Weibull-option*
WEIBULL2(*Weibull2-options*)	specifies two-parameter Weibull probability plot

Table 6.60 summarizes options that specify distribution parameters and control the display of a distribution reference line. Specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to request a normal probability plot with a distribution reference line:

```
proc capability data=measures;
   probplot length / normal(mu=10 sigma=0.3 color=red);
run;
```

The MU= and SIGMA= *normal-options* display a distribution reference line that corresponds to the normal distribution with mean $\mu_0 = 10$ and standard deviation $\sigma_0 = 0.3$, and the COLOR= *normal-option* specifies the color for the line.

Table 6.60 Distribution Options

Option	Description
Distribution Reference Line Options	
COLOR=	specifies color of distribution reference line
L=	specifies line type of distribution reference line
SYMBOL=	specifies plotting character for line printer plots
W=	specifies width of distribution reference line
Beta-Options	
ALPHA=	specifies mandatory shape parameter α
BETA=	specifies mandatory shape parameter β
SIGMA=	specifies σ_0 for distribution reference line
THETA=	specifies θ_0 for distribution reference line
Exponential-Options	
SIGMA=	specifies σ_0 for distribution reference line
THETA=	specifies θ_0 for distribution reference line
Gamma-Options	
ALPHA=	specifies mandatory shape parameter α
SIGMA=	specifies σ_0 for distribution reference line
THETA=	specifies θ_0 for distribution reference line
Gumbel-Options	
MU=	specifies location parameter μ
SIGMA=	specifies scale parameter σ
Lognormal-Options	
SIGMA=	specifies mandatory shape parameter σ
SLOPE=	specifies slope of distribution reference line
THETA=	specifies θ_0 for distribution reference line
ZETA=	specifies ζ_0 for distribution reference line (slope is $\exp(\zeta_0)$)
Normal-Options	
MU=	specifies μ_0 for distribution reference line
SIGMA=	specifies σ_0 for distribution reference line
Pareto-Options	
ALPHA=	specifies mandatory shape parameter α
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ
Power-Options	
ALPHA=	specifies mandatory shape parameter α
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ

Table 6.60 (*continued*)

Option	Description
Rayleigh-Options	
SIGMA=	specifies scale parameter σ
THETA=	specifies threshold parameter θ
Weibull-Options	
C=	specifies mandatory shape parameter c
SIGMA=	specifies σ_0 for distribution reference line
THETA=	specifies θ_0 for distribution reference line
Weibull2-Options	
C=	specifies c_0 for distribution reference line (slope is $1/c_0$)
SIGMA=	specifies σ_0 for distribution reference line (intercept is $\log(\sigma_0)$)
SLOPE=	specifies slope of distribution reference line
THETA=	specifies known lower threshold θ_0

General Options

Table 6.61 lists options that control the appearance of the plots.

Table 6.61 General PROBPLOT Statement Options

Option	Description
General Plot Layout Options	
CONTENTS=	specifies table of contents entry for probability plot grouping
GRID	draws grid lines perpendicular to the percentile axis
HREF=	specifies reference lines perpendicular to the horizontal axis
HREFLABELS=	specifies line labels for HREF= lines
LEGEND=	identifies LEGEND statement
NADJ=	adjusts sample size (N) when computing percentiles
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOLINELEGEND	suppresses distribution reference line information in legend
NOSPECLEGEND	suppresses specifications information in legend
PCTLMINOR	requests minor tick marks for percentile axis
PCTLORDER=	specifies tick mark labels for percentile axis
RANKADJ=	adjusts ranks when computing percentiles
ROTATE	switches horizontal and vertical axes
SQUARE	displays plot in square format
VREF=	specifies reference lines perpendicular to the vertical axis
VREFLABELS=	specifies line labels for VREF= lines
Graphics Options	
ANNOTATE=	specifies annotate data set
CAXIS=	specifies color for axis
CFRAME=	specifies color for frame
CGRID=	specifies color for grid lines

Table 6.61 (*continued*)

Option	Description
CHREF=	specifies colors for HREF= lines
CTEXT=	specifies color for text
CSTATREF=	specifies colors for STATREF= lines
CVREF=	specifies colors for VREF= lines
DESCRIPTION=	specifies description for plot in graphics catalog
FONT=	specifies software font for text
HAXIS=	specifies AXIS statement for horizontal axis
HEIGHT=	specifies height of text used outside framed areas
HMINOR=	specifies number of horizontal minor tick marks
HREFLABPOS=	specifies position for HREF= line labels
INFONT=	specifies software font for text inside framed areas
INHEIGHT=	specifies height of text inside framed areas
LGRID=	specifies a line type for grid lines
LHREF=	specifies line styles for HREF= lines
LSTATREF=	specifies line styles for STATREF= lines
LVREF=	specifies line styles for VREF= lines
NAME=	specifies name for plot in graphics catalog
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
STATREF=	specifies reference lines at values of summary statistics
STATREFLABELS=	specifies labels for STATREF= lines
STATREFSUBCHAR=	specifies substitution character for displaying statistic values in STATREFLABELS= labels
TURNVLABELS	turns and vertically strings out characters in labels for vertical axis
VAXIS=	specifies AXIS statement for vertical axis
VAXISLABEL=	specifies label for vertical axis
VMINOR=	specifies number of vertical minor tick marks
VREFLABPOS=	specifies horizontal position of labels for VREF= lines
WAXIS=	specifies line thickness for axes and frame
WGRID=	specifies line thickness for grid
Options for ODS Graphics Output	
ODSFOOTNOTE=	specifies footnote displayed on probability plot
ODSFOOTNOTE2=	specifies secondary footnote displayed on probability plot
ODSTITLE=	specifies title displayed on probability plot
ODSTITLE2=	specifies secondary title displayed on probability plot
Options for Comparative Plots	
ANNOKEY	applies annotation to key cell only
CFRAMESIDE=	specifies color for filling frame for row labels
CFRAMETOP=	specifies color for filling frame for column labels
CPROP=	specifies color for proportion of frequency bar
CTEXTSIDE=	specifies color for row labels
CTEXTTOP=	specifies color for column labels

Table 6.61 (*continued*)

Option	Description
INTERTILE=	specifies distance between tiles
NCOLS=	specifies number of columns in comparative probability plot
NROWS=	specifies number of rows in comparative probability plot
OVERLAY	overlays plots for different class levels (ODS Graphics only)
Options to Enhance Line Printer Plots	
GRIDCHAR=	specifies character for GRID lines
HREFCHAR=	specifies character for HREF= lines
NOOBSLEGEND	suppresses legend for hidden points
PROBSYMBOL=	specifies character for plotted points
VREFCHAR=	specifies character for VREF= lines

Dictionary of Options

The following sections provide detailed descriptions of options specific to the PROBPLOT statement. See "Dictionary of Common Options: CAPABILITY Procedure" on page 534 for detailed descriptions of options common to all the plot statements.

General Options

You can specify the following options whether you are producing ODS Graphics output or traditional graphics:

ALPHA=*value-list*|**EST**

specifies values for a mandatory shape parameter α ($\alpha > 0$) for probability plots requested with the BETA, GAMMA, PARETO, and POWER options. A plot is created for each value specified. For examples, see the entries for the distribution options. If you specify ALPHA=EST, a maximum likelihood estimate is computed for α.

BETA(ALPHA=*value-list*|**EST BETA=***value-list*|**EST <** *beta-options* **>)**

creates a beta probability plot for each combination of the shape parameters α and β given by the mandatory ALPHA= and BETA= options. If you specify ALPHA=EST and BETA=EST, a plot is created based on maximum likelihood estimates for α and β. In the following examples, the first PROBPLOT statement produces one plot, the second statement produces four plots, the third statement produces six plots, and the fourth statement produces one plot:

```
proc capability data=measures;
   probplot width / beta(alpha=2 beta=2);
   probplot width / beta(alpha=2 3 beta=1 2);
   probplot width / beta(alpha=2 to 3 beta=1 to 2 by 0.5);
   probplot width / beta(alpha=est beta=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $B^{-1}_{\alpha\beta}\left(\frac{i-0.375}{n+0.25}\right)$, where $B^{-1}_{\alpha\beta}(\cdot)$ is the inverse normalized incomplete beta

function, n is the number of nonmissing observations, and α and β are the shape parameters of the beta distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for ALPHA=α and BETA=β tends to be linear with intercept θ and slope σ if the data are beta distributed with the specific density function

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and

θ = lower threshold parameter
σ = scale parameter ($\sigma > 0$)
α = first shape parameter ($\alpha > 0$)
β = second shape parameter ($\beta > 0$)

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To obtain graphical estimates of α and β, specify lists of values for the ALPHA= and BETA= options, and select the combination of α and β that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *beta-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *beta-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / beta(alpha=2 beta=3 theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the beta distribution with parameters α, β, θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

BETA=*value-list*|**EST**
specifies values for the shape parameter β ($\beta > 0$) for probability plots requested with the BETA distribution option. A plot is created for each value specified with the BETA= option. If you specify BETA=EST, a maximum likelihood estimate is computed for β. For examples, see the preceding entry for the BETA option.

C=*value(-list)*|**EST**
specifies the shape parameter c ($c > 0$) for probability plots requested with the WEIBULL and WEIBULL2 options. You must specify C= as a *Weibull-option* with the WEIBULL option; in this situation it accepts a list of values, or if you specify C=EST, a maximum likelihood estimate is computed for c. You can optionally specify C=*value* or C=EST as a *Weibull2-option* with the WEIBULL2 option to request a distribution reference line; in this situation, you must also specify SIGMA=*value* or SIGMA=EST.

For example, the first PROBPLOT statement below creates three three-parameter Weibull plots corresponding to the shape parameters $c = 1$, $c = 2$, and $c = 3$. The second PROBPLOT statement

creates a single three-parameter Weibull plot corresponding to an estimated value of c. The third PROBPLOT statement creates a single two-parameter Weibull plot with a distribution reference line corresponding to $c_0 = 2$ and $\sigma_0 = 3$.

```
proc capability data=measures;
   probplot width / weibull(c=1 2 3);
   probplot width / weibull(c=est);
   probplot width / weibull2(c=2 sigma=3);
run;
```

EXPONENTIAL<**(***exponential-options*)**>**

EXP(<*exponential-options*>**)**

creates an exponential probability plot. To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $-\log\left(1 - \frac{i-0.375}{n+0.25}\right)$, where n is the number of nonmissing observations. The horizontal axis is scaled in percentile units.

The point pattern on the plot tends to be linear with intercept θ and slope σ if the data are exponentially distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where θ is a threshold parameter, and σ is a positive scale parameter.

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *exponential-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *exponential-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / exponential(theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the exponential distribution with parameters θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

GAMMA(ALPHA=*value-list***|EST** <*gamma-options*> **)**

creates a gamma probability plot for each value of the shape parameter α given by the mandatory ALPHA= option. If you specify ALPHA=EST, a plot is created based on a maximum likelihood estimate for α.

For example, the first PROBPLOT statement below creates three plots corresponding to $\alpha = 0.4$, $\alpha = 0.5$, and $\alpha = 0.6$. The second PROBPLOT statement creates a single plot.

```
proc capability data=measures;
   probplot width / gamma(alpha=0.4 to 0.6 by 0.2);
   probplot width / gamma(alpha=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $G_\alpha^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $G_\alpha^{-1}(\cdot)$ is the inverse normalized incomplete gamma function, n is the number of nonmissing observations, and α is the shape parameter of the gamma distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are gamma distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *gamma-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *gamma-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / gamma(alpha=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the gamma distribution with parameters α, θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

GRID

draws reference lines perpendicular to the percentile axis at major tick marks.

GUMBEL(< *Gumbel-options* **>)**

creates a Gumbel probability plot. To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $-\log\left(-\log\left(\frac{i-0.375}{n+0.25}\right)\right)$, where n is the number of nonmissing observations. The horizontal axis is scaled in percentile units.

The point pattern on the plot tends to be linear with intercept μ and slope σ if the data are Gumbel distributed with the specific density function

$$p(x) = \frac{e^{-(x-\mu)/\sigma}}{\sigma} \exp\left(-e^{-(x-\mu)/\sigma}\right)$$

where μ is a location parameter and σ is a positive scale parameter.

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to μ_0 and σ_0 with the *Gumbel-options* MU=μ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of μ_0 and σ_0 with the *Gumbel-options* MU=EST and SIGMA=EST. Specify these options in parentheses following the GUMBEL option.

Agreement between the reference line and the point pattern indicates that the Gumbel distribution with parameters μ_0 and σ_0 is a good fit.

LOGNORMAL(SIGMA=*value-list***|EST <** *lognormal-options* **>)**

LNORM(SIGMA=*value-list***|EST <** *lognormal-options* **>)**

creates a lognormal probability plot for each value of the shape parameter σ given by the mandatory SIGMA= option or its alias, the SHAPE= option. If you specify SIGMA=EST, a plot is created based on a maximum likelihood estimate for σ.

For example, the first PROBPLOT statement below produces two plots, and the second PROBPLOT statement produces a single plot:

```
proc capability data=measures;
   probplot width / lognormal(sigma=1.5 2.5 l=2);
   probplot width / lognormal(sigma=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $\exp\left(\sigma\,\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)\right)$, where $\Phi^{-1}(\cdot)$ is the inverse standard cumulative normal distribution, n is the number of nonmissing observations, and σ is the shape parameter of the lognormal distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for SIGMA=σ tends to be linear with intercept θ and slope $\exp(\zeta)$ if the data are lognormally distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = threshold parameter
ζ = scale parameter
σ = shape parameter ($\sigma > 0$)

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To obtain a graphical estimate of σ, specify a list of values for the SIGMA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and ζ_0 with the *lognormal-options* THETA=θ_0 and ZETA=ζ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and ζ_0 with the *lognormal-options* THETA=EST and ZETA=EST.

Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / lognormal(sigma=2 theta=3 zeta=0);
run;
```

Agreement between the reference line and the point pattern indicates that the lognormal distribution with parameters σ, θ_0, and ζ_0 is a good fit. See Example 6.20 for an example.

You can specify the THRESHOLD= option as an alias for the THETA= option and the SCALE= option as an alias for the ZETA= option.

MU=*value*|**EST**
specifies the mean μ_0 for a probability plot requested with the GUMBEL and NORMAL options. If you specify MU=EST, μ_0 is equal to the sample mean for the normal distribution. For the Gumbel distribution, a maximum likelihood estimate is calculated. See Example 6.19.

NADJ=*value*
specifies the adjustment value added to the sample size in the calculation of theoretical percentiles. The default is $\frac{1}{4}$, as recommended by Blom (1958). Also refer to Chambers et al. (1983) for additional information.

NOLEGEND
suppresses legends for specification limits, fitted curves, distribution lines, and hidden observations.

NOLINELEGEND
NOLINEL
suppresses the legend for the optional distribution reference line.

NORMAL<(*normal-options***)>**
NORM<(*normal-options***)>**
creates a normal probability plot. This is the default if you do not specify a distribution option. To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution, and n is the number of nonmissing observations. The horizontal axis is scaled in percentile units.

The point pattern on the plot tends to be linear with intercept μ and slope σ if the data are normally distributed with the specific

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad \text{for all } x$$

where μ is the mean and σ is the standard deviation ($\sigma > 0$).

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to μ_0 and σ_0 with the *normal-options* MU=μ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of μ_0 and σ_0 with the *normal-options* MU=EST and SIGMA=EST; the estimates of μ_0 and σ_0 are the sample mean and sample standard deviation.

Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot length / normal(mu=10 sigma=0.3);
   probplot length / normal(mu=est sigma=est);
run;
```

Agreement between the reference line and the point pattern indicates that the normal distribution with parameters μ_0 and σ_0 is a good fit.

NOSPECLEGEND
NOSPECL
suppresses the legend for specification limit reference lines.

PARETO(< *Pareto-options* **>)**
creates a generalized Pareto probability plot for each value of the shape parameter α given by the mandatory ALPHA= option. If you specify ALPHA=EST, a plot is created based on a maximum likelihood estimate for α.

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $(1-(1-\frac{i-0.375}{n+0.25})^{\alpha})/\alpha$ $(\alpha \neq 0)$ or $-\log(1-\frac{i-0.375}{n+0.25})$ $(\alpha = 0)$, where n is the number of nonmissing observations and α is the shape parameter of the generalized Pareto distribution. The horizontal axis is scaled in percentile units.

The point pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are generalized Pareto distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma}(1-\alpha(x-\theta)/\sigma)^{1/\alpha-1} & \text{if } \alpha \neq 0 \\ \frac{1}{\sigma}\exp(-(x-\theta)/\sigma) & \text{if } \alpha = 0 \end{cases}$$

where

θ = threshold parameter
σ = scale parameter $(\sigma > 0)$
α = shape parameter $(\alpha > 0)$

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *Pareto-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding

to estimated values of θ_0 and σ_0 with the *Pareto-options* THETA=EST and SIGMA=EST. Specify these options in parentheses following the PARETO option.

Agreement between the reference line and the point pattern indicates that the generalized Pareto distribution with parameters α, θ_0, and σ_0 is a good fit.

PCTLORDER=*value-list*

specifies the tick mark values labeled on the theoretical percentile axis. Because the values are percentiles, the labels must be between 0 and 100, exclusive. The values must be listed in increasing order and must cover the plotted percentile range. Otherwise, a default list is used. For example, consider the following:

```
proc capability data=measures;
   probplot length / pctlorder=1 10 25 50 75 90 99;
run;
```

Note that the ORDER= option in the AXIS statement is not supported by the PROBPLOT statement.

POWER(< *power-options* **>)**

creates a power function probability plot for each value of the shape parameter α given by the mandatory ALPHA= option. If you specify ALPHA=EST, a plot is created based on a maximum likelihood estimate for α.

To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $B^{-1}_{\alpha(1)}\left(\frac{i-0.375}{n+0.25}\right)$, where $B^{-1}_{\alpha(1)}(\cdot)$ is the inverse normalized incomplete beta function, n is the number of nonmissing observations, α is one shape parameter of the beta distribution, and the second shape parameter, $\beta = 1$. The horizontal axis is scaled in percentile units.

The point pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are power function distributed with the specific density function

$$p(x) = \begin{cases} \frac{\alpha}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1} & \text{for } \theta < x < \theta + \sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta + \sigma \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *power-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *power-options* THETA=EST and SIGMA=EST. Specify these options in parentheses following the POWER option.

Agreement between the reference line and the point pattern indicates that the power function distribution with parameters α, θ_0, and σ_0 is a good fit.

RANKADJ=*value*

specifies the adjustment value added to the ranks in the calculation of theoretical percentiles. The default is $-\frac{3}{8}$, as recommended by Blom (1958). Also refer to Chambers et al. (1983) for additional information.

RAYLEIGH(< *Rayleigh-options* **>)**

creates a Rayleigh probability plot. To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $\sqrt{-2\log\left(1-\frac{i-0.375}{n+0.25}\right)}$, where n is the number of nonmissing observations. The horizontal axis is scaled in percentile units.

The point pattern on the plot tends to be linear with intercept θ and slope σ if the data are Rayleigh distributed with the specific density function

$$p(x) = \begin{cases} \frac{x-\theta}{\sigma^2}\exp(-(x-\theta)^2/(2\sigma^2)) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where θ is a threshold parameter, and σ is a positive scale parameter.

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *Rayleigh-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Rayleigh-options* THETA=EST and SIGMA=EST. Specify these options in parentheses after the RAYLEIGH option.

Agreement between the reference line and the point pattern indicates that the Rayleigh distribution with parameters θ_0 and σ_0 is a good fit.

ROTATE

switches the horizontal and vertical axes so that the theoretical percentiles are plotted vertically while the data are plotted horizontally. Regardless of whether the plot has been rotated, horizontal axis options (such as HAXIS=) still refer to the horizontal axis, and vertical axis options (such as VAXIS=) still refer to the vertical axis. All other options that depend on axis placement adjust to the rotated axes.

SIGMA=*value-list***|EST**

specifies the value of the parameter σ, where $\sigma > 0$. Alternatively, you can specify SIGMA=EST to request a maximum likelihood estimate for σ_0. The interpretation and use of the SIGMA= option depend on the distribution option with which it is specified, as indicated by the following table.

Distribution Option	Use of the SIGMA= Option
BETA EXPONENTIAL GAMMA PARETO POWER RAYLEIGH WEIBULL	THETA=θ_0 and SIGMA=σ_0 request a distribution reference line corresponding to θ_0 and σ_0.
GUMBEL	MU=μ_0 and SIGMA=σ_0 request a distribution reference line corresponding to μ_0 and σ_0.
LOGNORMAL	SIGMA=$\sigma_1 \ldots \sigma_n$ requests n probability plots with shape parameters $\sigma_1 \ldots \sigma_n$. The SIGMA= option must be specified.
NORMAL	MU=μ_0 and SIGMA=σ_0 request a distribution reference line corresponding to μ_0 and σ_0. SIGMA=EST requests a line with σ_0 equal to the sample standard deviation.
WEIBULL2	SIGMA=σ_0 and C=c_0 request a distribution reference line corresponding to σ_0 and c_0.

In the following example, the first PROBPLOT statement requests a normal plot with a distribution reference line corresponding to $\mu_0 = 5$ and $\sigma_0 = 2$, and the second PROBPLOT statement requests a lognormal plot with shape parameter $\sigma = 3$:

```
proc capability data=measures;
   probplot length / normal(mu=5 sigma=2);
   probplot width  / lognormal(sigma=3);
run;
```

SLOPE=*value*|**EST**

specifies the slope for a distribution reference line requested with the LOGNORMAL and WEIBULL2 options. The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

When you use the SLOPE= option with the LOGNORMAL option, you must also specify a threshold parameter value θ_0 with the THETA= *lognormal-option* to request the line. The SLOPE= option is an alternative to the ZETA= *lognormal-option* for specifying ζ_0, because the slope is equal to $\exp(\zeta_0)$.

When you use the SLOPE= option with the WEIBULL2 option, you must also specify a scale parameter value σ_0 with the SIGMA= *Weibull2-option* to request the line. The SLOPE= option is an alternative to the C= *Weibull2-option* for specifying c_0, because the slope is equal to $1/c_0$. See "Location and Scale Parameters" on page 487.

For example, the first and second PROBPLOT statements below produce the same set of probability plots as the third and fourth PROBPLOT statements:

```
proc capability data=measures;
   probplot width / lognormal(sigma=2 theta=0 zeta=0);
   probplot width / weibull2(sigma=2 theta=0 c=0.25);
   probplot width / lognormal(sigma=2 theta=0 slope=1);
   probplot width / weibull2(sigma=2 theta=0 slope=4);
run;
```

SQUARE

displays the probability plot in a square frame. For an example, see Output 6.20.1. The default is a rectangular frame.

THETA=*value*|**EST**

THRESHOLD=*value*

specifies the lower threshold parameter θ for probability plots requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, PARETO, POWER, RAYLEIGH, WEIBULL, and WEIBULL2 options. When used with the WEIBULL2 option, the THETA= option specifies the known lower threshold θ_0, for which the default is 0. When used with the other distribution options, the THETA= option specifies θ_0 for a distribution reference line; alternatively in this situation, you can specify THETA=EST to request a maximum likelihood estimate for θ_0. To request the line, you must also specify a scale parameter. See Output 6.20.1 for an example of the THETA= option with a lognormal probability plot.

WEIBULL(C=*value-list*|**EST** < *Weibull-options* >**)**

WEIB(C=*value-list* < *Weibull-options* >**)**

creates a three-parameter Weibull probability plot for each value of the shape parameter c given by the mandatory C= option or its alias, the SHAPE= option. If you specify C=EST, a plot is created based on a maximum likelihood estimate for c. In the following example, the first PROBPLOT statement creates four plots, and the second PROBPLOT statement creates a single plot:

```
proc capability data=measures;
   probplot width / weibull(c=1.8 to 2.4 by 0.2 w=2);
   probplot width / weibull(c=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $\left(-\log\left(1-\frac{i-0.375}{n+0.25}\right)\right)^{\frac{1}{c}}$, where n is the number of nonmissing observations, and c is the Weibull distribution shape parameter. The horizontal axis is scaled in percentile units.

The point pattern on the plot for C=c tends to be linear with intercept θ and slope σ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)

The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

To obtain a graphical estimate of c, specify a list of values for the C= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *Weibull-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Weibull-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / weibull(c=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the Weibull distribution with parameters c, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

WEIBULL2<(*Weibull2-options*)>
W2<(*Weibull2-options*)>

creates a two-parameter Weibull probability plot. You should use the WEIBULL2 option when your data have a *known* lower threshold θ_0. You can specify the threshold value θ_0 with the THETA= *Weibull2-option* or its alias, the THRESHOLD= *Weibull2-option*. The default is $\theta_0 = 0$.

To create the plot, the observations are ordered from smallest to largest, and the log of the shifted ith ordered observation $x_{(i)}$, denoted by $\log(x_{(i)} - \theta_0)$, is plotted against the quantile $\log\left(-\log\left(1 - \frac{i-0.375}{n+0.25}\right)\right)$, where n is the number of nonmissing observations. The horizontal axis is scaled in percentile units. Note that the C= shape parameter option is not mandatory with the WEIBULL2 option.

The point pattern on the plot for THETA=θ_0 tends to be linear with intercept $\log(\sigma)$ and slope $\frac{1}{c}$ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right) & \text{for } x > \theta_0 \\ 0 & \text{for } x \le \theta_0 \end{cases}$$

where

θ_0 = known lower threshold
σ = scale parameter ($\sigma > 0$)
c = shape parameter ($c > 0$)

An advantage of the two-parameter Weibull plot over the three-parameter Weibull plot is that the parameters c and σ can be estimated from the slope and intercept of the point pattern. A disadvantage is that the two-parameter Weibull distribution applies only in situations where the threshold parameter is known.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to σ_0 and c_0 with the *Weibull2-options* SIGMA=σ_0 and C=c_0. Alternatively, you can add a distribution reference line corresponding to estimated values of σ_0 and c_0 with the *Weibull2-options* SIGMA=EST and C=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   probplot width / weibull2(theta=3 sigma=4 c=2);
run;
```

Agreement between the distribution reference line and the point pattern indicates that the Weibull distribution with parameters c_0, θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the SHAPE= option as an alias for the C= option.

ZETA=*value*|**EST**
specifies a value for the scale parameter ζ for lognormal probability plots requested with the LOGNORMAL option. Specify THETA=θ_0 and ZETA=ζ_0 to request a distribution reference line with intercept θ_0 and slope $\exp(\zeta_0)$. See Output 6.20.1 for an example.

Options for Traditional Graphics

You can specify the following options if you are producing traditional graphics:

CGRID=*color*
specifies the color for the grid lines requested by the GRID option.

LEGEND=*name* | **NONE**
specifies the name of a LEGEND statement describing the legend for specification limit reference lines and fitted curves. Specifying LEGEND=NONE is equivalent to specifying the NOLEGEND option.

LGRID=*linetype*
specifies the line type for the grid lines requested by the GRID option.

PCTLMINOR
requests minor tick marks for the percentile axis. See Output 6.20.1 for an example.

WGRID=*n*
specifies the width of the grid lines requested with the GRID option. If you use the WGRID= option, you do not need to specify the GRID option.

Options for Legacy Line Printer Plots

You can specify the following options if you are producing legacy line printer plots:

GRIDCHAR='*character*'
specifies the character used for the lines requested by the GRID option for a line printer plot. The default is the vertical bar (|).

NOOBSLEGEND
NOOBSL
suppresses the legend that indicates the number of hidden observations.

PROBSYMBOL='*character*'
specifies the character used to mark the points in a line printer plot. The default is the plus sign (+).

SYMBOL='*character*'
specifies the character used to display the distribution reference line in a line printer plot. The default character is the first letter of the distribution option keyword.

Details: PROBPLOT Statement

This section provides details on the following topics:

- distributions supported by the PROBPLOT statement
- SYMBOL statement options

Summary of Theoretical Distributions

You can use the PROBPLOT statement to request probability plots based on the theoretical distributions summarized in Table 6.62.

Table 6.62 Distributions and Parameters

Distribution	Density Function $p(x)$	Range	Parameters		
			Location	Scale	Shape
Beta	$\frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}}$	$\theta < x < \theta+\sigma$	θ	σ	α, β
Exponential	$\frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x \geq \theta$	θ	σ	
Gamma	$\frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x > \theta$	θ	σ	α
Gumbel	$\frac{e^{-(x-\mu)/\sigma}}{\sigma}\exp\left(-e^{-(x-\mu)/\sigma}\right)$	all x	μ	σ	
Lognormal (3-parameter)	$\frac{1}{\sigma\sqrt{2\pi}(x-\theta)}\exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right)$	$x > \theta$	θ	ζ	σ
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	all x	μ	σ	
Generalized Pareto	$\alpha \neq 0$ $\frac{1}{\sigma}(1-\alpha(x-\theta)/\sigma)^{1/\alpha-1}$ $\alpha = 0$ $\frac{1}{\sigma}\exp(-(x-\theta)/\sigma)$	$x > \theta$	θ	σ	α
Power Function	$\frac{\alpha}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}$	$x > \theta$	θ	σ	α
Rayleigh	$\frac{x-\theta}{\sigma^2}\exp(-(x-\theta)^2/(2\sigma^2))$	$x \geq \theta$	θ	σ	
Weibull (3-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right)$	$x > \theta$	θ	σ	c

Table 6.62 (*continued*)

Distribution	Density Function $p(x)$	Range	Parameters		
			Location	Scale	Shape
Weibull (2-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right)$	$x > \theta_0$ (known)	θ_0	σ	c

You can request these distributions with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, WEIBULL, and WEIBULL2 options, respectively. If you do not specify a distribution option, a normal probability plot is created.

Shape Parameters

Some of the distribution options in the PROBPLOT statement require you to specify one or two shape parameters in parentheses after the distribution keyword. These are summarized in Table 6.63.

Table 6.63 Shape Parameter Options for the PROBPLOT Statement

Distribution Keyword	Mandatory Shape Parameter Option	Range
BETA	ALPHA=α, BETA=β	$\alpha > 0, \beta > 0$
EXPONENTIAL	None	
GAMMA	ALPHA=α	$\alpha > 0$
GUMBEL	None	
LOGNORMAL	SIGMA=σ	$\sigma > 0$
NORMAL	None	
PARETO	ALPHA=α	$\alpha > 0$
POWER	ALPHA=α	$\alpha > 0$
RAYLEIGH	None	
WEIBULL	C=c	$c > 0$
WEIBULL2	None	

You can visually estimate the value of a shape parameter by specifying a list of values for the shape parameter option. The PROBPLOT statement produces a separate plot for each value. You can then use the value of the shape parameter producing the most nearly linear point pattern. Alternatively, you can request that the plot be created using an estimated shape parameter. For an example, see "Creating Lognormal Probability Plots" on page 464.

Location and Scale Parameters

If you specify the location and scale parameters for a distribution (or if you request estimates for these parameters), a diagonal distribution reference line is displayed on the plot. (An exception is the two-parameter Weibull distribution, for which a line is displayed when you specify or estimate the scale and shape parameters.) Agreement between this line and the point pattern indicates that the distribution with these parameters is a good fit. For illustrations, see Example 6.19 and Example 6.20.

The following table shows how the specified parameters determine the intercept[5] and slope of the line:

Table 6.64 Intercept and Slope of Distribution Reference Line

	Parameters			Linear Pattern	
Distribution	**Location**	**Scale**	**Shape**	**Intercept**	**Slope**
Beta	θ	σ	α , β	θ	σ
Exponential	θ	σ		θ	σ
Gamma	θ	σ	α	θ	σ
Gumbel	μ	σ		μ	σ
Lognormal	θ	ζ	σ	θ	$\exp(\zeta)$
Normal	μ	σ		μ	σ
Generalized Pareto	θ	σ	α	θ	σ
Power Function	θ	σ	α	θ	σ
Rayleigh	θ	σ		θ	σ
Weibull (3-parameter)	θ	σ	c	θ	σ
Weibull (2-parameter)	θ_0 (known)	σ	c	$\log(\sigma)$	$\frac{1}{c}$

For the LOGNORMAL and WEIBULL2 options, you can specify the slope directly with the SLOPE= option. That is, for the LOGNORMAL option, specifying THETA=θ_0 and SLOPE=$\exp(\zeta_0)$ displays the same line as specifying THETA=θ_0 and ZETA=ζ_0. For the WEIBULL2 option, specifying SIGMA=σ_0 and SLOPE=$\frac{1}{c_0}$ displays the same line as specifying SIGMA=σ_0 and C=c_0.

SYMBOL Statement Options

In earlier releases of SAS/QC software, graphical features of lower and upper specification lines and diagonal distribution reference lines were controlled with options in the SYMBOL2, SYMBOL3, and SYMBOL4 statements, respectively. These options are still supported, although they have been superseded by options in the PROBPLOT and SPEC statements. Table 6.65 summarizes the two sets of options. **NOTE:** These statements have no effect on ODS Graphics output.

[5] The intercept and slope are based on the quantile scale for the horizontal axis, which is displayed on a Q-Q plot; see "QQPLOT Statement: CAPABILITY Procedure" on page 493.

Table 6.65 SYMBOL Statement Options

Feature	Statement and Options	Alternative Statement and Options
Symbol markers	**SYMBOL1 Statement**	
character	VALUE=*special-symbol*	
color	COLOR=*color*	
font	FONT=*font*	
height	HEIGHT=*value*	
Lower specification line	**SPEC Statement**	**SYMBOL2 Statement**
position	LSL=*value*	
color	CLSL=*color*	COLOR=*color*
line type	LLSL=*linetype*	LINE=*linetype*
width	WLSL=*value*	WIDTH=*value*
Upper specification line	**SPEC Statement**	**SYMBOL3 Statement**
position	USL=*value*	
color	CUSL=*color*	COLOR=*color*
line type	LUSL=*linetype*	LINE=*linetype*
width	WUSL=*value*	WIDTH=*value*
Target reference line	**SPEC Statement**	
position	TARGET=*value*	
color	CTARGET=*color*	
line type	LTARGET=*linetype*	
width	WTARGET=*value*	
Distribution reference line	**PROBPLOT Statement**	**SYMBOL4 Statement**
color	COLOR=*color*	COLOR=*color*
line type	LINE=*linetype*	LINE=*linetype*
width	WIDTH=*value*	WIDTH=*value*

For an illustration of these options, see Example 6.19.

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. PROBPLOT options used to control the appearance of traditional graphics are ignored for ODS Graphics output.

When ODS Graphics is in effect, the PROBPLOT statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 6.66.

Table 6.66 ODS Graphics Produced by the PROBPLOT Statement

ODS Graph Name	Plot Description
ProbPlot	probability plot

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Examples: PROBPLOT Statement

This section provides advanced examples of the PROBPLOT statement.

Example 6.19: Displaying a Normal Reference Line

NOTE: See *Probability Plot with Normal Reference Line* in the SAS/QC Sample Library.

Measurements of the distance between two holes cut into 50 steel sheets are saved as values of the variable Distance in the following data set:

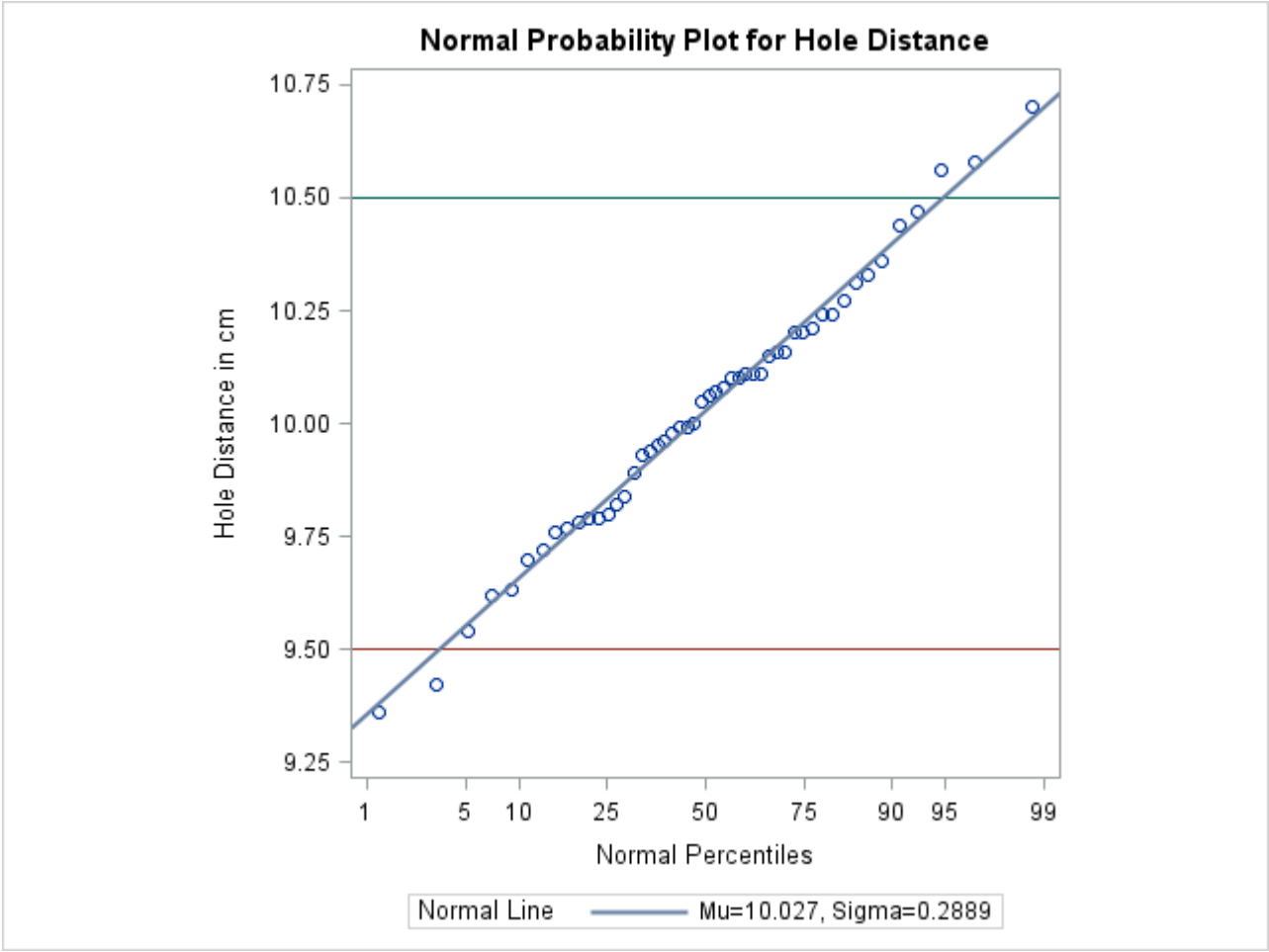
```
data Sheets;
   input Distance @@;
   label Distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;
```

The cutting process is in control, and you decide to check whether the process distribution is normal. The following statements create a normal probability plot for Distance with lower and upper specification lines at 9.5 cm and 10.5 cm:

```
title 'Normal Probability Plot for Hole Distance';
proc capability data=Sheets noprint;
   spec lsl=9.5 usl=10.5;
   probplot Distance / normal(mu=est sigma=est)
                       square
                       odstitle=title
                       nospeclegend;
run;
```

The plot is shown in Output 6.19.1. The MU= and SIGMA= *normal-options* request the diagonal reference line that corresponds to the normal distribution with estimated parameters $\hat{\mu} = 10.027$ and $\hat{\sigma} = 0.2889$. The LSL= and USL= SPEC statement options request the lower and upper specification lines. The SYMBOL statement specifies the symbol marker for the plotted points.

Output 6.19.1 Normal Reference Line



Example 6.20: Displaying a Lognormal Reference Line

NOTE: See *Creating Lognormal Probability Plots* in the SAS/QC Sample Library.

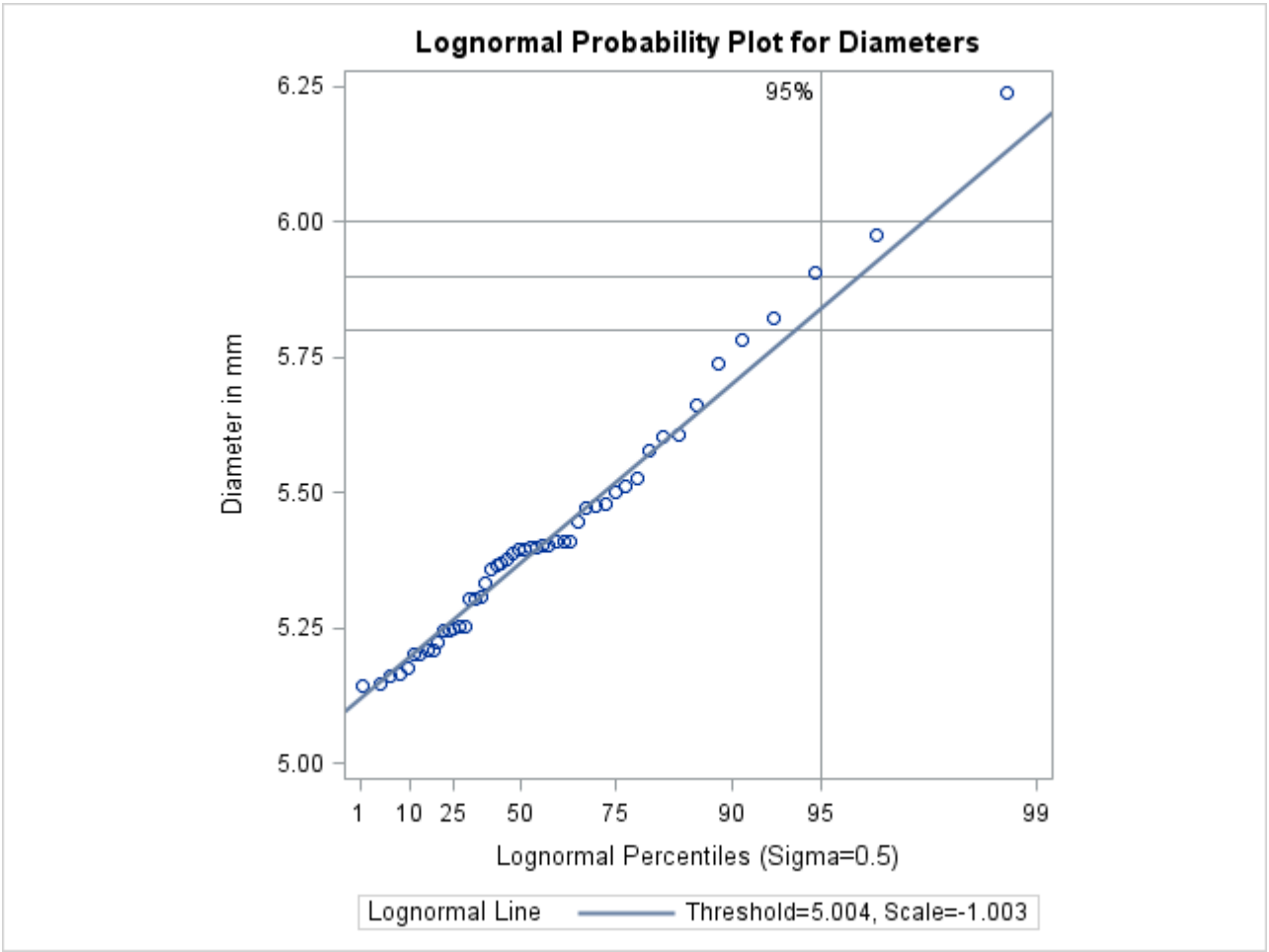
This example is a continuation of "Creating Lognormal Probability Plots" on page 464. Figure 6.36 shows that a lognormal distribution with shape parameter $\sigma = 0.5$ is a good fit for the distribution of Diameter in the data set Rods.

The lognormal distribution involves two other parameters: a threshold parameter θ and a scale parameter ζ. See Table 6.62 for the equation of the lognormal density function. The following statements illustrate how you can request a diagonal distribution reference line whose slope and intercept are determined by estimates of θ and ζ.

```
title 'Lognormal Probability Plot for Diameters';
proc capability data=Rods noprint;
   probplot Diameter / lognormal(sigma=0.5 theta=est zeta=est)
                       square
                       pctlminor
                       href      = 95
                       hreflabel = '95%'
                       vref      = 5.8 to 6.0 by 0.1
                       odstitle  = title;
run;
```

The plot is shown in Output 6.20.1.

Output 6.20.1 Lognormal Reference Line



The close agreement between the diagonal reference line and the point pattern indicates that the specific lognormal distribution with $\hat{\sigma} = 0.5$, $\hat{\theta} = 5.004$, and $\hat{\zeta} = -1.003$ is a good fit for the diameter measurements.

Specifying HREF=95 adds a reference line indicating the 95th percentile of the lognormal distribution. The HREFLABEL= option specifies a label for this line. The PCTLMINOR option displays minor tick marks on

the percentile axis. The VREF= option adds reference lines indicating diameter values of 5.8, 5.9, and 6.0, and the CHREF= and CVREF= options specify colors for the horizontal and vertical reference lines.

Based on the intersection of the diagonal reference line with the HREF= line, the estimated 95th percentile of the diameter distribution is 5.85 mm.

Note that you could also construct a similar plot in which all three parameters are estimated by substituting SIGMA=EST for SIGMA=0.5 in the preceding statements.

QQPLOT Statement: CAPABILITY Procedure

Overview: QQPLOT Statement

The QQPLOT statement creates a quantile-quantile plot (Q-Q plot), which compares ordered values of a variable with quantiles of a specified theoretical distribution such as the normal. If the data distribution matches the theoretical distribution, the points on the plot form a linear pattern. Thus, you can use a Q-Q plot to determine how well a theoretical distribution models a set of measurements.

You can specify one of the following theoretical distributions with the QQPLOT statement:

- beta
- exponential
- gamma
- Gumbel
- three-parameter lognormal
- normal
- generalized Pareto
- power function
- Rayleigh
- two-parameter Weibull
- three-parameter Weibull

You can use options in the QQPLOT statement to do the following:

- specify or estimate parameters for the theoretical distribution
- display a reference line corresponding to specific location and scale parameters for the theoretical distribution
- request graphical enhancements

You can also create a comparative Q-Q plot by using the QQPLOT statement in conjunction with a CLASS statement.

You have three alternatives for producing Q-Q plots with the QQPLOT statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: Q-Q plots are similar to probability plots, which you can create with the PROBPLOT statement (see "PROBPLOT Statement: CAPABILITY Procedure" on page 461). Q-Q plots are preferable for graphical estimation of distribution parameters and capability indices, whereas probability plots are preferable for graphical estimation of percentiles.

Getting Started: QQPLOT Statement

The following examples illustrate the basic syntax of the QQPLOT statement. For complete details of the QQPLOT statement, see the section "Syntax: QQPLOT Statement" on page 497. Advanced examples are provided on the section "Examples: QQPLOT Statement" on page 523.

Creating a Normal Quantile-Quantile Plot

NOTE: See *Creating Normal Q-Q Plots* in the SAS/QC Sample Library.

Measurements of the distance between two holes cut into 50 steel sheets are saved as values of the variable Distance in the following data set:

```
data Sheets;
   input Distance @@;
   label Distance='Hole Distance in cm';
   datalines;
 9.80 10.20 10.27  9.70  9.76
10.11 10.24 10.20 10.24  9.63
 9.99  9.78 10.10 10.21 10.00
 9.96  9.79 10.08  9.79 10.06
10.10  9.95  9.84 10.11  9.93
10.56 10.47  9.42 10.44 10.16
10.11 10.36  9.94  9.77  9.36
 9.89  9.62 10.05  9.72  9.82
 9.99 10.16 10.58 10.70  9.54
10.31 10.07 10.33  9.98 10.15
;
```

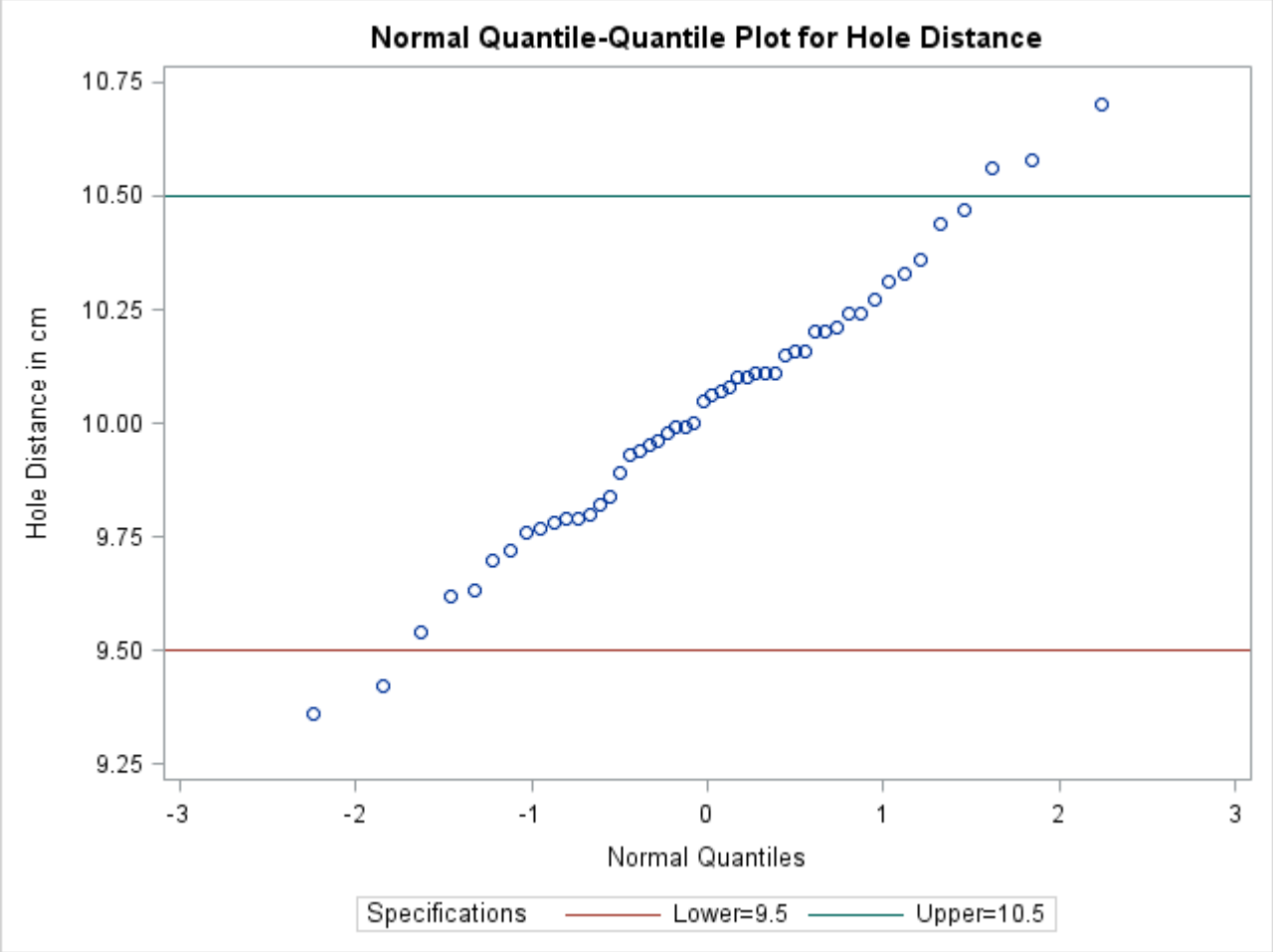
The cutting process is in control, and you decide to check whether the process distribution is normal. The following statements create a Q-Q plot for Distance, shown in Figure 6.39, with lower and upper specification lines at 9.5 cm and 10.5 cm:[6]

[6] For a P-P plot using these data, see Figure 6.31. For a probability plot using these data, see Example 6.20.

```
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=Sheets noprint;
   spec lsl=9.5 usl=10.5;
   qqplot Distance / odstitle=title;
run;
```

The plot compares the ordered values of Distance with quantiles of the normal distribution. The linearity of the point pattern indicates that the measurements are normally distributed. Note that a normal Q-Q plot is created by default. The specification lines are requested with the LSL= and USL= options in the SPEC statement.

Figure 6.39 Normal Quantile-Quantile Plot Created with Traditional Graphics



Adding a Distribution Reference Line

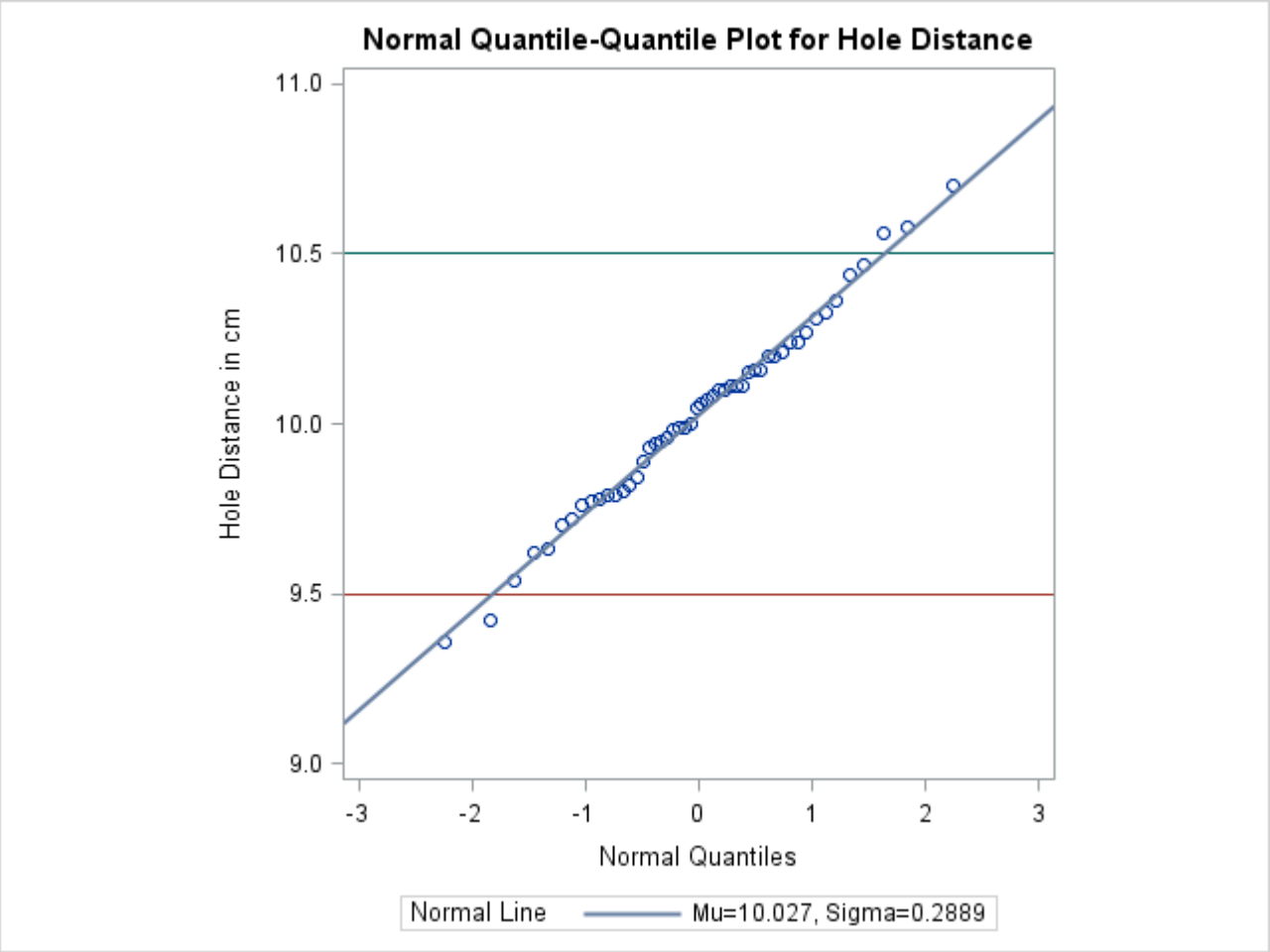
NOTE: See *Creating Normal Q-Q Plots* in the SAS/QC Sample Library.

In a normal Q-Q plot, the normal distribution with mean μ_0 and standard deviation σ_0 is represented by a reference line with intercept μ_0 and slope σ_0. The following statements reproduce the Q-Q plot in Figure 6.39, adding the line for which μ_0 and σ_0 are estimated by the sample mean and standard deviation:

```
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=Sheets noprint;
   spec lsl=9.5 usl=10.5;
   qqplot Distance / normal(mu=est sigma=est)
                     square
                     nospeclegend
                     odstitle=title;
run;
```

The plot is displayed in Figure 6.40.

Figure 6.40 Adding a Distribution Reference Line to a Q-Q Plot



Specifying MU=EST and SIGMA=EST with the NORMAL option requests the reference line (alternatively, you can specify numeric values for μ_0 and σ_0 with the MU= and SIGMA= options). The COLOR= and L= options specify the color of the line and the line type. The SQUARE option displays the plot in a square format, and the NOSPECLEGEND option suppresses the legend for the specification lines.

Syntax: QQPLOT Statement

The syntax for the QQPLOT statement is as follows:

QQPLOT < *variables* > < / *options* > **;**

You can specify the keyword QQ as an alias for QQPLOT, and you can use any number of QQPLOT statements in the CAPABILITY procedure. The components of the QQPLOT statement are described as follows.

variables

are the process variables for which to create Q-Q plots. If you specify a VAR statement, the variables must also be listed in the VAR statement. Otherwise, the variables can be any numeric variables in the input data set. If you do not specify a list of variables, then by default the procedure creates a Q-Q plot for each variable listed in the VAR statement, or for each numeric variable in the DATA= data set if you do not specify a VAR statement. For example, each of the following QQPLOT statements produces two Q-Q plots, one for length and one for width:

```
proc capability data=measures;
   var length width;
   qqplot;
run;

proc capability data=measures;
   qqplot length width;
run;
```

options

specify the theoretical distribution for the plot or add features to the plot. If you specify more than one variable, the options apply equally to each variable. Specify all options after the slash (/) in the QQPLOT statement. You can specify only one option naming the distribution in each QQPLOT statement, but you can specify any number of other options. The distributions available are the beta, exponential, gamma, Gumbel, lognormal, normal, generalized Pareto, power function, Rayleigh, two-parameter Weibull, and three-parameter Weibull. By default, the procedure produces a plot for the normal distribution.

In the following example, the NORMAL option requests a normal Q-Q plot for each variable. The MU= and SIGMA= *normal-options* request a distribution reference line with intercept 10 and slope 0.3 for each plot, corresponding to a normal distribution with mean $\mu = 10$ and standard deviation $\sigma = 0.3$. The SQUARE option displays the plot in a square frame, and the CTEXT= option specifies the text color.

```
proc capability data=measures;
   qqplot length1 length2 / normal(mu=10 sigma=0.3)
                            square
                            ctext=blue;
run;
```

Summary of Options

The following tables list the QQPLOT statement options by function. For complete descriptions, see "Dictionary of Options" on page 502.

Distribution Options

Table 6.67 summarizes the options for requesting a specific theoretical distribution.

Table 6.67 Options for Specifying a Theoretical Distribution

Option	Description
BETA(*beta-options*)	specifies beta Q-Q plot for shape parameters α, β specified with mandatory ALPHA= and BETA= *beta-options*
EXPONENTIAL(*exponential-options*)	specifies exponential Q-Q plot
GAMMA(*gamma-options*)	specifies gamma Q-Q plot for shape parameter α specified with mandatory ALPHA= *gamma-option*
GUMBEL(*Gumbel-options*)	specifies Gumbel Q-Q plot
LOGNORMAL(*lognormal-options*)	specifies lognormal Q-Q plot for shape parameter σ specified with mandatory SIGMA= *lognormal-option*
NORMAL(*normal-options*)	specifies normal Q-Q plot
PARETO(*Pareto-options*)	specifies generalized Pareto Q-Q plot for shape parameter α specified with mandatory ALPHA= *Pareto-option*
POWER(*power-options*)	specifies power function Q-Q plot for shape parameter α specified with mandatory ALPHA= *power-option*
RAYLEIGH(*Rayleigh-options*)	specifies Rayleigh Q-Q plot
WEIBULL(*Weibull-options*)	specifies three-parameter Weibull Q-Q plot for shape parameter c specified with mandatory C= *Weibull-option*
WEIBULL2(*Weibull2-options*)	specifies two-parameter Weibull Q-Q plot

Table 6.68 summarizes options that specify parameter values for theoretical distributions and that control the display of a distribution reference line. Specify these options in parentheses after the distribution option. For example, the following statements use the NORMAL option to request a normal Q-Q plot with a specific distribution reference line. The MU= and SIGMA= *normal-options* display a distribution reference line with intercept 10 and slope 0.3. The COLOR= *normal-option* draws the line in red.

```
proc capability data=measures;
   qqplot length / normal(mu=10 sigma=0.3 color=red);
run;
```

Table 6.68 Distribution Options

Option	Description
Distribution Reference Line Options	
COLOR=	specifies color of distribution reference line
L=	specifies line type of distribution reference line
SYMBOL=	specifies plotting character for line printer plots
W=	specifies width of distribution reference line
Beta-Options	
ALPHA=	specifies mandatory shape parameter α
BETA=	specifies mandatory shape parameter β
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ
Exponential-Options	
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ
Gamma-Options	
ALPHA=	specifies mandatory shape parameter α
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ
Gumbel-Options	
MU=	specifies reference line intercept μ
SIGMA=	specifies reference line slope σ
Lognormal-Options	
SIGMA=	specifies mandatory shape parameter σ
SLOPE=	specifies reference line slope
THETA=	specifies reference line intercept θ
ZETA=	specifies reference line slope $\exp(\zeta_0)$
Normal-Options	
CPKREF	specifies vertical reference lines at intersection of specification limits with distribution reference line
CPKSCALE	rescales horizontal axis in C_{pk} units
MU=	specifies reference line intercept μ
SIGMA=	specifies reference line slope σ
Pareto-Options	
ALPHA=	specifies mandatory shape parameter α
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ
Power-Options	
ALPHA=	specifies mandatory shape parameter α
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ
Rayleigh-Options	
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ

Table 6.68 (*continued*)

Option	Description
Weibull-Options	
C=	specifies mandatory shape parameter c
SIGMA=	specifies reference line slope σ
THETA=	specifies reference line intercept θ
Weibull2-Options	
C=	specifies c_0 for reference line (slope is $\frac{1}{c_0}$)
SIGMA=	specifies σ_0 for reference line (intercept is $\log(\sigma_0)$)
SLOPE=	specifies reference line slope
THETA=	specifies known lower threshold θ_0

General Options

Table 6.69 lists options that control the appearance of the plots.

Table 6.69 General QQPLOT Statement Options

Option	Description
General Plot Layout Options	
CONTENTS=	specifies table of contents entry for Q-Q plot grouping
HREF=	specifies reference lines perpendicular to the horizontal axis
HREFLABELS=	specifies labels for HREF= lines
LEGEND=	specifies LEGEND statement
NADJ=	adjusts sample size (N) when computing quantiles
NOFRAME	suppresses frame around plotting area
NOLEGEND	suppresses legend
NOLINELEGEND	suppresses distribution reference line information in legend
NOSPECLEGEND	suppresses specifications information in legend
PCTLAXIS	adds a nonlinear percentile axis
PCTLMINOR	adds minor tick marks to percentile axis
PCTLSCALE	replaces theoretical quantiles with percentiles
RANKADJ=	adjusts ranks when computing quantiles
ROTATE	switches horizontal and vertical axes
SQUARE	displays Q-Q plot in square format
VREF=	specifies reference lines perpendicular to the vertical axis
VREFLABELS=	specifies labels for VREF= lines
Graphics Options	
ANNOTATE=	specifies annotate data set
CAXIS=	specifies color for axis
CFRAME=	specifies color for frame
CGRID=	specifies color for grid lines
CHREF=	specifies colors for HREF= lines
CSTATREF=	specifies colors for STATREF= lines
CTEXT=	specifies color for text

Table 6.69 (*continued*)

Option	Description
CVREF=	specifies colors for VREF= lines
DESCRIPTION=	specifies description for plot in graphics catalog
FONT=	specifies software font for text
GRID	draws grid lines perpendicular to the quantile axis
HEIGHT=	specifies height of text used outside framed areas
HMINOR=	specifies number of horizontal minor tick marks
HREFLABPOS=	specifies vertical position of labels for HREF= lines
INFONT=	specifies software font for text inside framed areas
INHEIGHT=	specifies height of text inside framed areas
LGRID=	specifies a line type for grid lines
LHREF=	specifies line styles for HREF= lines
LSTATREF=	specifies line styles for STATREF= lines
LVREF=	specifies line styles for VREF= lines
NAME=	specifies name for plot in graphics catalog
NOHLABEL	suppresses label for horizontal axis
NOVLABEL	suppresses label for vertical axis
NOVTICK	suppresses tick marks and tick mark labels for vertical axis
STATREF=	specifies reference lines at values of summary statistics
STATREFLABELS=	specifies labels for STATREF= lines
STATREFSUBCHAR=	specifies substitution character for displaying statistic values in STATREFLABELS= labels
VAXIS=	specifies AXIS statement for vertical axis
VAXISLABEL=	specifies label for vertical axis
VMINOR=	specifies number of vertical minor tick marks
VREFLABPOS=	specifies horizontal position of labels for VREF= lines
WAXIS=	specifies line thickness for axes and frame
WGRID=	specifies thickness for grid lines
Options for ODS Graphics Output	
ODSFOOTNOTE=	specifies footnote displayed on Q-Q plot
ODSFOOTNOTE2=	specifies secondary footnote displayed on Q-Q plot
ODSTITLE=	specifies title displayed on Q-Q plot
ODSTITLE2=	specifies secondary title displayed on Q-Q plot
Options for Comparative Plots	
ANNOKEY	applies annotation requested in ANNOTATE= data set to key cell only
CFRAMESIDE=	specifies color for filling frame for row labels
CFRAMETOP=	specifies color for filling frame for column labels
CPROP=	specifies color for proportion of frequency bar
CTEXTSIDE=	specifies color for row labels
CTEXTTOP=	specifies color for column labels
INTERTILE=	specifies distance between tiles
NCOLS=	specifies number of columns in comparative Q-Q plot
NROWS=	specifies number of rows in comparative Q-Q plot
OVERLAY	overlays plots for different class levels (ODS Graphics only)

Table 6.69 (*continued*)

Option	Description
Options to Enhance Line Printer Plots	
HREFCHAR=	specifies line character for HREF= lines
NOOBSLEGEND	suppresses legend for hidden points
QQSYMBOL=	specifies character for plotted points
VREFCHAR=	specifies character for VREF= lines

Dictionary of Options

The following sections provide detailed descriptions of options specific to the QQPLOT statement. See "Dictionary of Common Options: CAPABILITY Procedure" on page 534 for detailed descriptions of options common to all the plot statements.

General Options

You can specify the following options whether you are producing ODS Graphics output or traditional graphics:

ALPHA=*value-list*|**EST**

specifies values for a mandatory shape parameter α ($\alpha > 0$) for Q-Q plots requested with the BETA, GAMMA, PARETO, and POWER options. A plot is created for each value specified. For examples, see the entries for the distribution options. If you specify ALPHA=EST, a maximum likelihood estimate is computed for α.

BETA(ALPHA=*value-list*|**EST BETA=***value-list*|**EST <** *beta-options* **>)**

creates a beta Q-Q plot for each combination of the shape parameters α and β given by the mandatory ALPHA= and BETA= options. If you specify ALPHA=EST and BETA=EST, a plot is created based on maximum likelihood estimates for α and β. In the following example, the first QQPLOT statement produces one plot, the second statement produces four plots, the third statement produces six plots, and the fourth statement produces one plot:

```
proc capability data=measures;
   qqplot width / beta(alpha=2 beta=2);
   qqplot width / beta(alpha=2 3 beta=1 2);
   qqplot width / beta(alpha=2 to 3 beta=1 to 2 by 0.5);
   qqplot width / beta(alpha=est beta=est);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $B^{-1}_{\alpha\beta}\left(\frac{i-0.375}{n+0.25}\right)$, where $B^{-1}_{\alpha\beta}(\cdot)$ is the inverse normalized incomplete beta function, n is the number of nonmissing observations, and α and β are the shape parameters of the beta distribution.

The point pattern on the plot for ALPHA=α and BETA=β tends to be linear with intercept θ and slope σ if the data are beta distributed with the specific density function

$$p(x) = \begin{cases} \frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}} & \text{for } \theta < x < \theta+\sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta+\sigma \end{cases}$$

where $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, and
θ = lower threshold parameter
σ = scale parameter ($\sigma > 0$)
α = first shape parameter ($\alpha > 0$)
β = second shape parameter ($\beta > 0$)

To obtain graphical estimates of α and β, specify lists of values for the ALPHA= and BETA= options, and select the combination of α and β that most nearly linearizes the point pattern. To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *beta-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and slope σ_0 with the *beta-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot width / beta(alpha=2 beta=3 theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the beta distribution with parameters α, β, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

BETA=*value-list*|**EST**
specifies values for the shape parameter β ($\beta > 0$) for Q-Q plots requested with the BETA distribution option. A plot is created for each value specified with the BETA= option. If you specify BETA=EST, a maximum likelihood estimate is computed for β. For examples, see the preceding entry for the BETA distribution option.

C=*value(-list)*|**EST**
specifies the shape parameter c ($c > 0$) for Q-Q plots requested with the WEIBULL and WEIBULL2 options. You must specify C= as a *Weibull-option* with the WEIBULL option; in this situation it accepts a list of values, or if you specify C=EST, a maximum likelihood estimate is computed for c. You can optionally specify C=*value* or C=EST as a *Weibull2-option* with the WEIBULL2 option to request a distribution reference line; in this situation, you must also specify SIGMA=*value* or SIGMA=EST. For an example, see Output 6.23.1.

CPKSCALE
rescales the quantile axis in C_{pk} units for plots requested with the NORMAL option. Specify CPKSCALE in parentheses after the NORMAL option. You can use the CPKSCALE option with the CPKREF option for graphical estimation of the capability indices *CPU*, *CPL*, and C_{pk}, as illustrated in Output 6.24.1.

EXPONENTIAL(< (*exponential-options*) **>**
EXP< (*exponential-options*) **>)**
creates an exponential Q-Q plot. To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $-\log\left(1 - \frac{i-0.375}{n+0.25}\right)$, where n is the number of nonmissing observations.

The pattern on the plot tends to be linear with intercept θ and slope σ if the data are exponentially distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma} \exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x \geq \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where θ is the threshold parameter, and σ is the scale parameter ($\sigma > 0$).

To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *exponential-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and slope σ_0 with the *exponential-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example: as in the following example:

```
proc capability data=measures;
   qqplot width / exponential(theta=4 sigma=5);
run;
```

Agreement between the reference line and the point pattern indicates that the exponential distribution with parameters θ_0 and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

GAMMA(ALPHA=*value-list***|EST <** *gamma-options* **>)**

creates a gamma Q-Q plot for each value of the shape parameter α given by the mandatory ALPHA= option or its alias, the SHAPE= option. The following example produces three probability plots:

```
proc capability data=measures;
   qqplot width / gamma(alpha=0.4 to 0.6 by 0.1);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $G_\alpha^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $G_\alpha^{-1}(\cdot)$ is the inverse normalized incomplete gamma function, n is the number of nonmissing observations, and α is the shape parameter of the gamma distribution.

The pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are gamma distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\Gamma(\alpha)} \left(\frac{x-\theta}{\sigma}\right)^{\alpha-1} \exp\left(-\frac{x-\theta}{\sigma}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *gamma-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *gamma-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot width / gamma(alpha=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the gamma distribution with parameters α, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

GUMBEL(< *Gumbel-options* **>)**

creates a Gumbel Q-Q plot. To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $-\log\left(-\log\left(\frac{i-0.375}{n+0.25}\right)\right)$, where n is the number of nonmissing observations.

The point pattern on the plot tends to be linear with intercept μ and slope σ if the data are Gumbel distributed with the specific density function

$$p(x) = \frac{e^{-(x-\mu)/\sigma}}{\sigma} \exp\left(-e^{-(x-\mu)/\sigma}\right)$$

where μ is a location parameter and σ is a positive scale parameter.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to μ_0 and σ_0 with the *Gumbel-options* MU=μ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of μ_0 and σ_0 with the *Gumbel-options* MU=EST and SIGMA=EST. Specify these options in parentheses following the GUMBEL option.

Agreement between the reference line and the point pattern indicates that the Gumbel distribution with parameters μ_0 and σ_0 is a good fit.

GRID

draws reference lines perpendicular to the quantile axis at major tick marks.

LEGEND=*name* **| NONE**

specifies the name of a LEGEND statement describing the legend for specification limit reference lines and fitted curves. Specifying LEGEND=NONE is equivalent to specifying the NOLEGEND option.

LOGNORMAL(SIGMA=*value-list***|EST <** *lognormal-options* **>)**

LNORM(SIGMA=*value-list***|EST <** *lognormal-options* **>)**

creates a lognormal Q-Q plot for each value of the shape parameter σ given by the mandatory SIGMA= option or its alias, the SHAPE= option. For example,

```
proc capability data=measures;
   qqplot width/ lognormal(shape=1.5 2.5);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the *i*th ordered observation is plotted against the quantile $\exp\left(\sigma\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)\right)$, where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution, n is the number of nonmissing observations, and σ is the shape parameter of the lognormal distribution.

The pattern on the plot for SIGMA=σ tends to be linear with intercept θ and slope $\exp(\zeta)$ if the data are lognormally distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

where
θ = threshold parameter
ζ = scale parameter
σ = shape parameter ($\sigma > 0$)

To obtain a graphical estimate of σ, specify a list of values for the SIGMA= option, and select the value that most nearly linearizes the point pattern. For an illustration, see Example 6.22.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to the threshold parameter θ_0 and the scale parameter ζ_0 with the *lognormal-options* THETA=θ_0 and ZETA=ζ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and ζ_0 with the *lognormal-options* THETA=EST and ZETA=EST. This line has intercept θ_0 and slope $\exp(\zeta_0)$. Agreement between the line and the point pattern indicates that the lognormal distribution with parameters σ, θ_0, and ζ_0 is a good fit. See Output 6.22.4 for an example. You can specify the THRESHOLD= option as an alias for the THETA= option and the SCALE= option as an alias for the ZETA= option.

You can also display the reference line by specifying THETA=θ_0, and you can specify the slope with the SLOPE= option. For example, the following two QQPLOT statements produce charts with identical reference lines:

```
proc capability data=measures;
   qqplot width / lognormal(sigma=2 theta=3 zeta=1);
   qqplot width / lognormal(sigma=2 theta=3 slope=2.718);
run;
```

MU=*value*|**EST**

specifies a value for the mean μ for a Q-Q plot requested with the GUMBEL and NORMAL options. For the normal distribution, you can specify MU=EST to request a distribution reference line with intercept equal to the sample mean, as illustrated in Figure 6.40. If you specify MU=EST for the Gumbel distribution, a maximum likelihood estimate is calculated.

NADJ=*value*

specifies the adjustment value added to the sample size in the calculation of theoretical quantiles. The default is $\frac{1}{4}$, as described by Blom (1958). Also refer to Chambers et al. (1983) for additional information.

NOLEGEND

LEGEND=NONE

suppresses legends for specification limits, fitted curves, distribution lines, and hidden observations. For an example, see Output 6.24.1.

NOLINELEGEND

NOLINEL

suppresses the legend for the optional distribution reference line.

NORMAL<(*normal-options***)>**

NORM<(*normal-options***)>**

creates a normal Q-Q plot. This is the default if you do not specify a distribution option. To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution, and n is the number of nonmissing observations.

The pattern on the plot tends to be linear with intercept μ and slope σ if the data are normally distributed with the specific density function

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad \text{for all } x$$

where μ is the mean, and σ is the standard deviation ($\sigma > 0$).

To assess the point pattern, you can add a diagonal distribution reference line with intercept μ_0 and slope σ_0 with the *normal-options* MU=μ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of μ_0 and σ_0 with the *normal-options* MU=EST and SIGMA=EST; the estimates of μ_0 and σ_0 are the sample mean and sample standard deviation. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot length / normal(mu=10 sigma=0.3);
run;
```

For an example, see "Adding a Distribution Reference Line" on page 495. Agreement between the reference line and the point pattern indicates that the normal distribution with parameters μ_0 and σ_0 is a good fit. You can specify MU=EST and SIGMA=EST to request a distribution reference line with the sample mean and sample standard deviation as the intercept and slope.

Other *normal-options* include CPKREF and CPKSCALE. The CPKREF option draws reference lines extending from the intersections of specification limits with the distribution reference line to the theoretical quantile axis. The CPKSCALE option rescales the theoretical quantile axis in C_{pk} units. You can use the CPKREF option with the CPKSCALE option for graphical estimation of the capability indices *CPU*, *CPL*, and C_{pk}, as illustrated in Output 6.24.1.

NOSPECLEGEND
NOSPECL

suppresses the legend for specification limit reference lines. For an example, see Figure 6.40.

PARETO(< *Pareto-options* **>)**

creates a generalized Pareto Q-Q plot for each value of the shape parameter α given by the mandatory ALPHA= option. If you specify ALPHA=EST, a plot is created based on a maximum likelihood estimate for α.

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $(1-(1-\frac{i-0.375}{n+0.25})^{\alpha})/\alpha$ $(\alpha \neq 0)$ or $-\log(1-\frac{i-0.375}{n+0.25})$ $(\alpha = 0)$, where n is the number of nonmissing observations and α is the shape parameter of the generalized Pareto distribution.

The point pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are generalized Pareto distributed with the specific density function

$$p(x) = \begin{cases} \frac{1}{\sigma}(1-\alpha(x-\theta)/\sigma)^{1/\alpha-1} & \text{if } \alpha \neq 0 \\ \frac{1}{\sigma}\exp(-(x-\theta)/\sigma) & \text{if } \alpha = 0 \end{cases}$$

where

θ = threshold parameter
σ = scale parameter $(\sigma > 0)$
α = shape parameter $(\alpha > 0)$

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *Pareto-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Pareto-options* THETA=EST and SIGMA=EST. Specify these options in parentheses following the PARETO option.

Agreement between the reference line and the point pattern indicates that the generalized Pareto distribution with parameters α, θ_0, and σ_0 is a good fit.

PCTLAXIS(*axis-options***)**

adds a nonlinear percentile axis along the frame of the Q-Q plot opposite the theoretical quantile axis. The added axis is identical to the axis for probability plots produced with the PROBPLOT statement. When using the PCTLAXIS option, you must specify HREF= values in quantile units, and you cannot use the NOFRAME option. You can specify the following *axis-options*:

CGRID=*color*

specifies the color used for grid lines.

GRID

draws grid lines perpendicular to the percentile axis at major tick marks.

GRIDCHAR=‘*character***’**

specifies the character used to draw grid lines associated with the percentile axis on line printer plots.

LABEL='*string*'
specifies the label for the percentile axis.

LGRID=*linetype*
specifies the line type used for grid lines associated with the percentile axis.

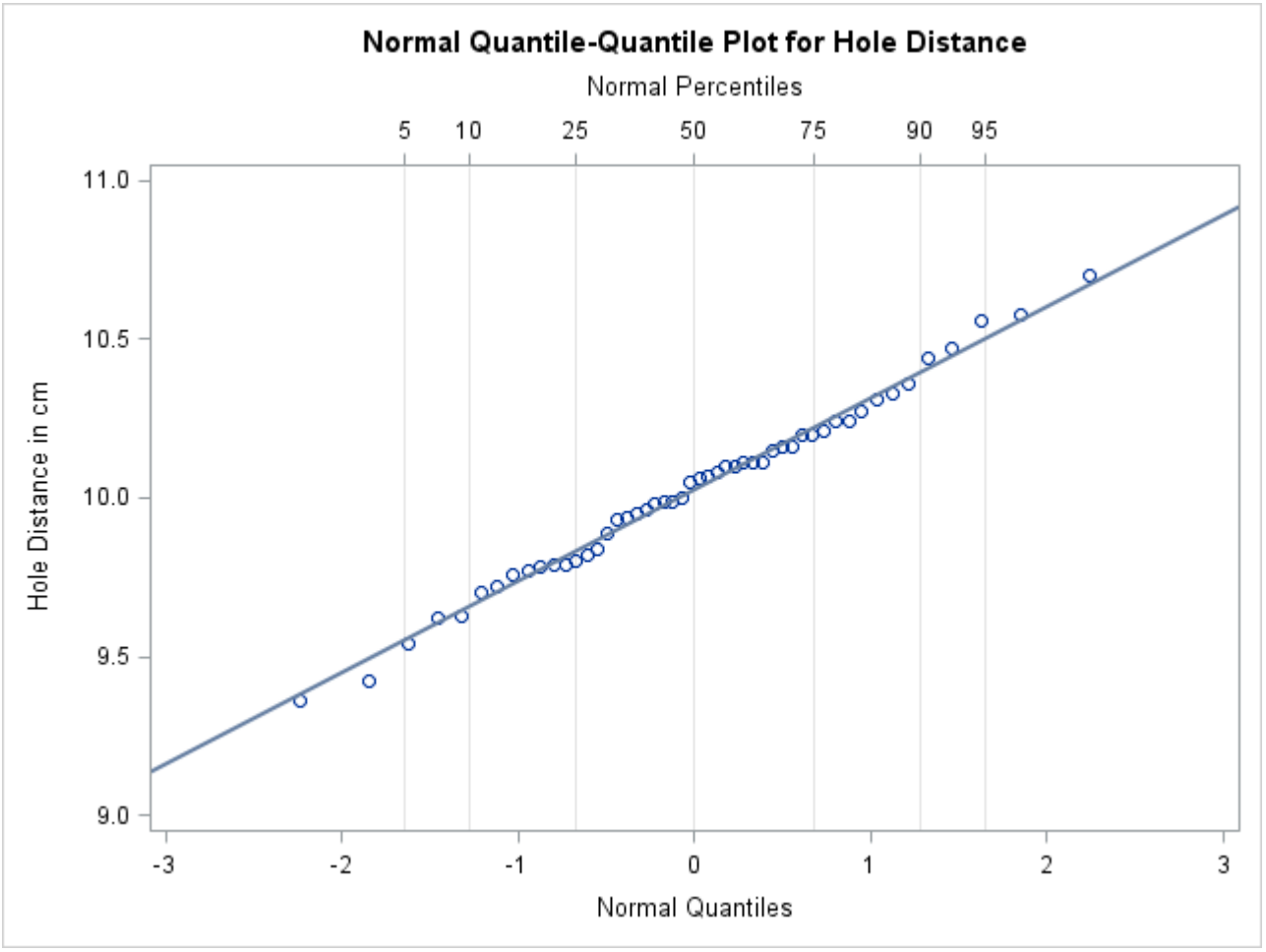
WGRID=*value*
specifies the thickness for grid lines associated with the percentile axis.

NOTE: See *Creating Normal Q-Q Plots* in the SAS/QC Sample Library.

For example, the following statements display the plot in Figure 6.41:

```
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=Sheets noprint;
   qqplot Distance / normal(mu=est sigma=est)
                     nolegend
                     pctlaxis(grid label='Normal Percentiles')
                     odstitle=title;
run;
```

Figure 6.41 Normal Q-Q Plot with Percentile Axis



PCTLSCALE

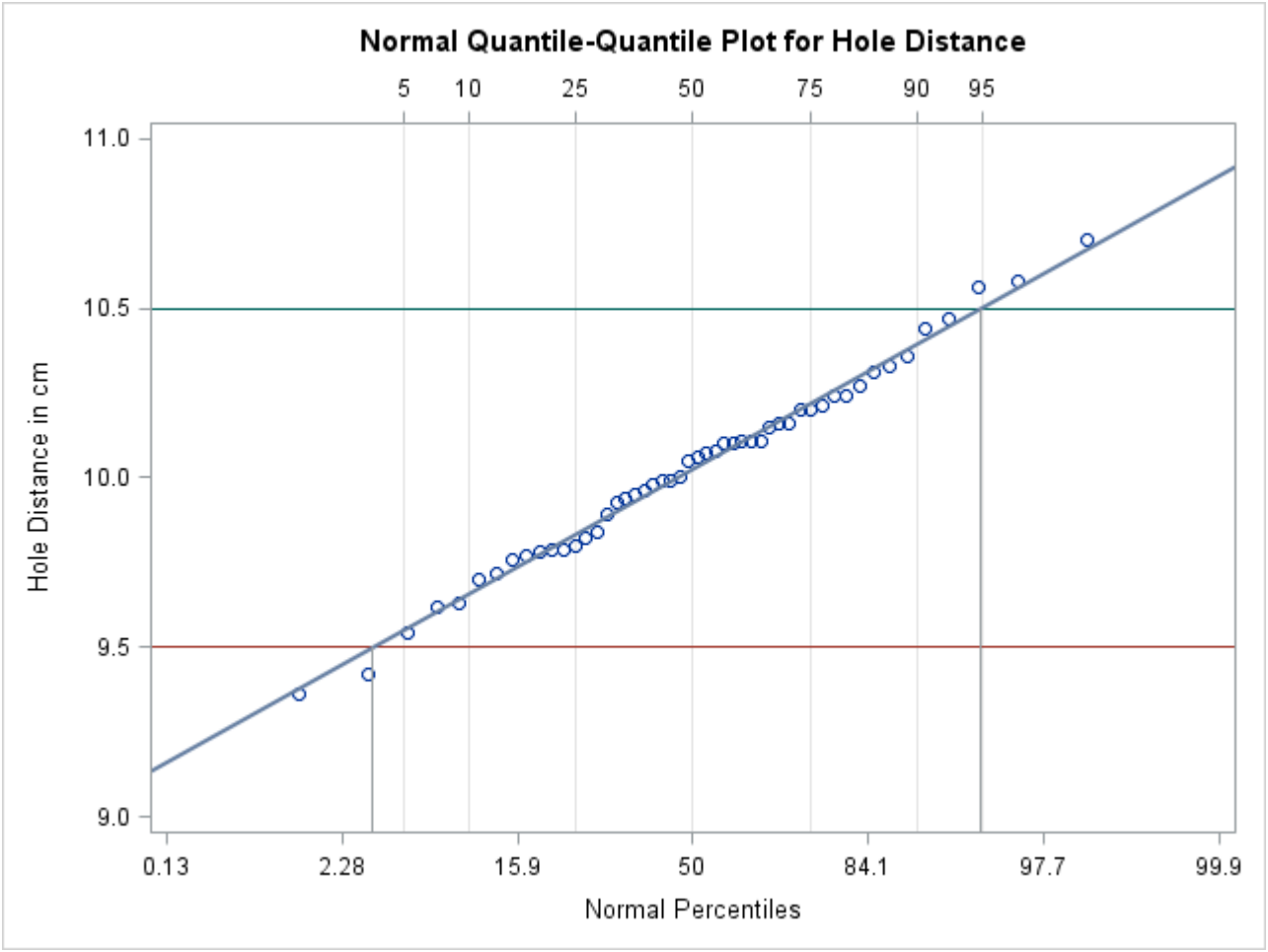
requests scale labels for the theoretical quantile axis in percentile units, resulting in a nonlinear axis scale. Tick marks are drawn uniformly across the axis based on the quantile scale. In all other respects, the plot remains the same, and you must specify HREF= values in quantile units. For a true nonlinear axis, use the PCTLAXIS option or use the PROBPLOT statement.

NOTE: See *Creating Normal Q-Q Plots* in the SAS/QC Sample Library.

For example, the following statements display the plot in Figure 6.42:

```
title 'Normal Quantile-Quantile Plot for Hole Distance';
proc capability data=Sheets noprint;
   spec lsl=9.5 usl=10.5;
   qqplot Distance / normal(mu=est sigma=est cpkref)
                     pctlaxis(grid lgrid=35)
                     nolegend pctlscale
                     odstitle=title;
run;
```

Figure 6.42 Normal Q-Q Plot for Reading Percentiles of Specification Limits



POWER(< *power-options* **>)**

creates a power function Q-Q plot for each value of the shape parameter α given by the mandatory ALPHA= option. If you specify ALPHA=EST, a plot is created based on a maximum likelihood estimate for α.

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $B^{-1}_{\alpha(1)}\left(\frac{i-0.375}{n+0.25}\right)$, where $B^{-1}_{\alpha(1)}(\cdot)$ is the inverse normalized incomplete beta function, n is the number of nonmissing observations, α is one shape parameter of the beta distribution, and the second shape parameter, $\beta = 1$.

The point pattern on the plot for ALPHA=α tends to be linear with intercept θ and slope σ if the data are power function distributed with the specific density function

$$p(x) = \begin{cases} \frac{\alpha}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1} & \text{for } \theta < x < \theta + \sigma \\ 0 & \text{for } x \le \theta \text{ or } x \ge \theta + \sigma \end{cases}$$

where

θ = threshold parameter
σ = scale parameter ($\sigma > 0$)
α = shape parameter ($\alpha > 0$)

To obtain a graphical estimate of α, specify a list of values for the ALPHA= option, and select the value that most nearly linearizes the point pattern.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *power-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *power-options* THETA=EST and SIGMA=EST. Specify these options in parentheses following the POWER option.

Agreement between the reference line and the point pattern indicates that the power function distribution with parameters α, θ_0, and σ_0 is a good fit.

RANKADJ=*value*

specifies the adjustment value added to the ranks in the calculation of theoretical quantiles. The default is $-\frac{3}{8}$, as described by Blom (1958). Also refer to Chambers et al. (1983) for additional information.

RAYLEIGH(< *Rayleigh-options* **>)**

creates a Rayleigh Q-Q plot. To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $\sqrt{-2\log\left(1-\frac{i-0.375}{n+0.25}\right)}$, where n is the number of nonmissing observations.

The point pattern on the plot tends to be linear with intercept θ and slope σ if the data are Rayleigh distributed with the specific density function

$$p(x) = \begin{cases} \frac{x-\theta}{\sigma^2}\exp(-(x-\theta)^2/(2\sigma^2)) & \text{for } x \ge \theta \\ 0 & \text{for } x < \theta \end{cases}$$

where θ is a threshold parameter, and σ is a positive scale parameter.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to θ_0 and σ_0 with the *Rayleigh-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Rayleigh-options* THETA=EST and SIGMA=EST. Specify these options in parentheses after the RAYLEIGH option.

Agreement between the reference line and the point pattern indicates that the Rayleigh distribution with parameters θ_0 and σ_0 is a good fit.

ROTATE

switches the horizontal and vertical axes so that the theoretical percentiles are plotted vertically while the data are plotted horizontally. Regardless of whether the plot has been rotated, horizontal axis options (such as HAXIS=) refer to the horizontal axis, and vertical axis options (such as VAXIS=) refer to the vertical axis. All other options that depend on axis placement adjust to the rotated axes.

SIGMA=*value-list*|**EST**

specifies the value of the distribution parameter σ, where $\sigma > 0$. Alternatively, you can specify SIGMA=EST to request a maximum likelihood estimate for σ_0. The use of the SIGMA= option depends on the distribution option specified, as indicated by the following table:

Distribution Option	Use of the SIGMA= Option
BETA EXPONENTIAL GAMMA PARETO POWER RAYLEIGH WEIBULL	THETA=θ_0 and SIGMA=σ_0 request a distribution reference line with intercept θ_0 and slope σ_0.
GUMBEL	MU=μ_0 and SIGMA=σ_0 request a distribution reference line corresponding to μ_0 and σ_0.
LOGNORMAL	SIGMA=$\sigma_1 \ldots \sigma_n$ requests n Q-Q plots with shape parameters $\sigma_1 \ldots \sigma_n$. The SIGMA= option is mandatory.
NORMAL	MU=μ_0 and SIGMA=σ_0 request a distribution reference line with intercept μ_0 and slope σ_0. SIGMA=EST requests a slope equal to the sample standard deviation.
WEIBULL2	SIGMA=σ_0 and C=c_0 request a distribution reference line with intercept $\log(\sigma_0)$ and slope $\frac{1}{c_0}$.

For an example using SIGMA=EST, see Output 6.24.1. For an example of lognormal plots using the SIGMA= option, see Example 6.22.

SLOPE=*value*|**EST**

specifies the slope for a distribution reference line requested with the LOGNORMAL and WEIBULL2 options.

When you use the SLOPE= option with the LOGNORMAL option, you must also specify a threshold parameter value θ_0 with the THETA= option. Specifying the SLOPE= option is an alternative to specifying ZETA=ζ_0, which requests a slope of $\exp(\zeta_0)$. See Output 6.22.4 for an example.

When you use the SLOPE= option with the WEIBULL2 option, you must also specify a scale parameter value σ_0 with the SIGMA= option. Specifying the SLOPE= option is an alternative to specifying C=c_0, which requests a slope of $\frac{1}{c_0}$.

For example, the first and second QQPLOT statements that follow produce plots identical to those produced by the third and fourth QQPLOT statements:

```
proc capability data=measures;
   qqplot width  / lognormal(sigma=2 theta=0 zeta=0);
   qqplot width  / weibull2(sigma=2 theta=0 c=0.25);
   qqplot width  / lognormal(sigma=2 theta=0 slope=1);
   qqplot width  / weibull2(sigma=2 theta=0 slope=4);
run;
```

For more information, see "Graphical Estimation" on page 519.

SQUARE

displays the Q-Q plot in a square frame. Compare Figure 6.39 with Figure 6.40. The default is a rectangular frame.

THETA=*value*|**EST**

THRESHOLD=*value*|**EST**

specifies the lower threshold parameter θ for Q-Q plots requested with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, PARETO, POWER, RAYLEIGH, WEIBULL, and WEIBULL2 options.

When used with the WEIBULL2 option, the THETA= option specifies the known lower threshold θ_0, for which the default is 0. See Output 6.23.2 for an example.

When used with the other distribution options, the THETA= option specifies θ_0 for a distribution reference line; alternatively in this situation, you can specify THETA=EST to request a maximum likelihood estimate for θ_0. To request the line, you must also specify a scale parameter. See Output 6.22.4 for an example of the THETA= option with a lognormal Q-Q plot.

WEIBULL(C=*value-list*|**EST < ***Weibull-options* **>)**

WEIB(C=*value-list* **< ***Weibull-options* **>)**

creates a three-parameter Weibull Q-Q plot for each value of the shape parameter c given by the mandatory C= option or its alias, the SHAPE= option. For example,

```
proc capability data=measures;
   qqplot width / weibull(c=1.8 to 2.4 by 0.2);
run;
```

To create the plot, the observations are ordered from smallest to largest, and the ith ordered observation is plotted against the quantile $\left(-\log\left(1-\frac{i-0.375}{n+0.25}\right)\right)^{\frac{1}{c}}$, where n is the number of nonmissing observations, and c is the Weibull distribution shape parameter.

The pattern on the plot for C=c tends to be linear with intercept θ and slope σ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right) & \text{for } x > \theta \\ 0 & \text{for } x \le \theta \end{cases}$$

where θ is the threshold parameter, σ is the scale parameter ($\sigma > 0$), and c is the shape parameter ($c > 0$).

To obtain a graphical estimate of c, specify a list of values for the C= option, and select the value that most nearly linearizes the point pattern. For an illustration, see Example 6.23. To assess the point pattern, you can add a diagonal distribution reference line with intercept θ_0 and slope σ_0 with the *Weibull-options* THETA=θ_0 and SIGMA=σ_0. Alternatively, you can add a line corresponding to estimated values of θ_0 and σ_0 with the *Weibull-options* THETA=EST and SIGMA=EST. Specify these options in parentheses, as in the following example:

```
proc capability data=measures;
   qqplot width / weibull(c=2 theta=3 sigma=4);
run;
```

Agreement between the reference line and the point pattern indicates that the Weibull distribution with parameters c, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the THRESHOLD= option as an alias for the THETA= option.

WEIBULL2<(*Weibull2-options*)>

W2<(*Weibull2-options*)>

creates a two-parameter Weibull Q-Q plot. You should use the WEIBULL2 option when your data have a *known* lower threshold θ_0. You can specify the threshold value θ_0 with the THETA= option or its alias, the THRESHOLD= option. If you are uncertain of the lower threshold value, you can estimate θ_0 graphically by specifying a list of values for the THETA= option. Select the value that most linearizes the point pattern. The default is $\theta_0 = 0$.

To create the plot, the observations are ordered from smallest to largest, and the log of the shifted ith ordered observation $x_{(i)}$, $\log(x_{(i)} - \theta_0)$, is plotted against the quantile $\log\left(-\log\left(1 - \frac{i-0.375}{n+0.25}\right)\right)$, where n is the number of nonmissing observations. Unlike the three-parameter Weibull quantile, the preceding expression is free of distribution parameters. This is why the C= shape parameter option is not mandatory with the WEIBULL2 option.

The pattern on the plot for THETA=θ_0 tends to be linear with intercept $\log(\sigma)$ and slope $\frac{1}{c}$ if the data are Weibull distributed with the specific density function

$$p(x) = \begin{cases} \frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1} \exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right) & \text{for } x > \theta_0 \\ 0 & \text{for } x \le \theta_0 \end{cases}$$

where θ_0 is a known lower threshold parameter, σ is a scale parameter ($\sigma > 0$), and c is a shape parameter ($c > 0$).

The advantage of a two-parameter Weibull plot over a three-parameter Weibull plot is that you can visually estimate the shape parameter c and the scale parameter σ from the slope and intercept of the point pattern; see Example 6.23 for an illustration of this method. The disadvantage is that the two-parameter Weibull distribution applies only in situations where the threshold parameter is known. See "Graphical Estimation" on page 519 for more information.

To assess the point pattern, you can add a diagonal distribution reference line corresponding to the scale parameter σ_0 and shape parameter c_0 with the *Weibull2-options* SIGMA=σ_0 and C=c_0. Alternatively, you can add a distribution reference line corresponding to estimated values of σ_0 and c_0 with the

Weibull2-options SIGMA=EST and C=EST. This line has intercept $\log(\sigma_0)$ and slope $\frac{1}{c_0}$. Agreement between the line and the point pattern indicates that the Weibull distribution with parameters c_0, θ_0, and σ_0 is a good fit. You can specify the SCALE= option as an alias for the SIGMA= option and the SHAPE= option as an alias for the C= option.

You can also display the reference line by specifying SIGMA=σ_0, and you can specify the slope with the SLOPE= option. For example, the following QQPLOT statements produce identical plots:

```
proc capability data=measures;
   qqplot width / weibull2(theta=3 sigma=4 c=2);
   qqplot width / weibull2(theta=3 sigma=4 slope=0.5);
run;
```

ZETA=*value*|**EST**
specifies a value for the scale parameter ζ for lognormal Q-Q plots requested with the LOGNORMAL option. Specify THETA=θ_0 and ZETA=ζ_0 to request a distribution reference line with intercept θ_0 and slope $\exp(\zeta_0)$.

Options for Traditional Graphics

You can specify the following options if you are producing traditional graphics:

CGRID=*color*
specifies the color for the grid lines associated with the quantile axis, requested by the GRID option.

LGRID=*linetype*
specifies the line type for the grid lines associated with the quantile axis, requested by the GRID option.

CPKREF
draws reference lines extending from the intersections of the specification limits with the distribution reference line to the quantile axis in plots requested with the NORMAL option. Specify CPKREF in parentheses after the NORMAL option. You can use the CPKREF option with the CPKSCALE option for graphical estimation of the capability indices *CPU*, *CPL*, and C_{pk}, as illustrated in Output 6.24.1.

PCTLMINOR
requests minor tick marks for the percentile axis displayed when you use the PCTLAXIS option. See the entry for the PCTLAXIS option for an example.

WGRID=*n*
specifies the width of the grid lines associated with the quantile axis, requested with the GRID option. If you use the WGRID= option, you do not need to specify the GRID option.

Options for Legacy Line Printer Plots

You can specify the following options if you are producing legacy line printer plots:

NOOBSLEGEND

NOOBSL
suppresses the legend that indicates the number of hidden observations.

QQSYMBOL=‘*character*’
specifies the character used to plot the Q-Q points in line printer plots. The default is the plus sign (+).

SYMBOL=‘*character*’
specifies the character used for a distribution reference line in a line printer plot. The default character is the first letter of the distribution option keyword.

Details: QQPLOT Statement

This section provides details on the following topics:

- construction of Q-Q plots
- interpretation of Q-Q plots
- distributions supported by the QQPLOT statement
- graphical estimation of shape parameters, location and scale parameters, theoretical percentiles, and capability indices
- SYMBOL statement options

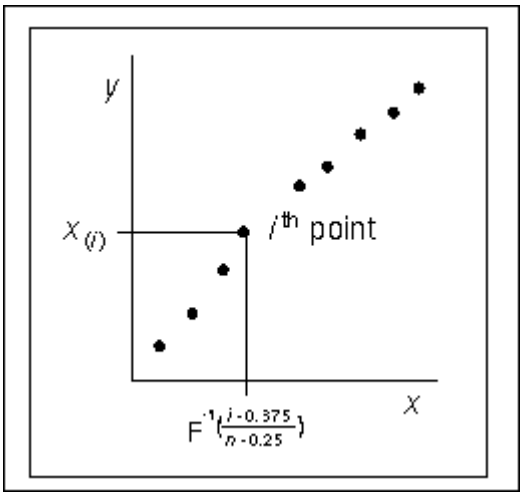
Construction of Quantile-Quantile and Probability Plots

Figure 6.43 illustrates how a Q-Q plot is constructed. First, the n nonmissing values of the variable are ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

Then the ith ordered value $x_{(i)}$ is represented on the plot by a point whose y-coordinate is $x_{(i)}$ and whose x-coordinate is $F^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $F(\cdot)$ is the theoretical distribution with zero location parameter and unit scale parameter.

Figure 6.43 Construction of a Q-Q Plot



You can modify the adjustment constants –0.375 and 0.25 with the RANKADJ= and NADJ= options. This default combination is recommended by Blom (1958). For additional information, refer to Chambers et al. (1983). Because $x_{(i)}$ is a quantile of the empirical cumulative distribution function (ecdf), a Q-Q plot compares quantiles of the ecdf with quantiles of a theoretical distribution. Probability plots (see "PROBPLOT Statement: CAPABILITY Procedure" on page 461) are constructed the same way, except that the x-axis is scaled nonlinearly in percentiles.

Interpretation of Quantile-Quantile and Probability Plots

The following properties of Q-Q plots and probability plots make them useful diagnostics of how well a specified theoretical distribution fits a set of measurements:

- If the quantiles of the theoretical and data distributions agree, the plotted points fall on or near the line $y = x$.
- If the theoretical and data distributions differ only in their location or scale, the points on the plot fall on or near the line $y = ax + b$. The slope a and intercept b are visual estimates of the scale and location parameters of the theoretical distribution.

Q-Q plots are more convenient than probability plots for graphical estimation of the location and scale parameters because the x-axis of a Q-Q plot is scaled linearly. On the other hand, probability plots are more convenient for estimating percentiles or probabilities.

There are many reasons why the point pattern in a Q-Q plot may not be linear. Chambers et al. (1983) and Fowlkes (1987) discuss the interpretations of commonly encountered departures from linearity, and these are summarized in the following table.

Table 6.70 Quantile-Quantile Plot Diagnostics

Description of Point Pattern	Possible Interpretation
All but a few points fall on a line	Outliers in the data
Left end of pattern is below the line; right end of pattern is above the line	Long tails at both ends of the data distribution
Left end of pattern is above the line; right end of pattern is below the line	Short tails at both ends of the data distribution
Curved pattern with slope increasing from left to right	Data distribution is skewed to the right
Curved pattern with slope decreasing from left to right	Data distribution is skewed to the left
Staircase pattern (plateaus and gaps)	Data have been rounded or are discrete

In some applications, a nonlinear pattern may be more revealing than a linear pattern. However, Chambers et al. (1983) note that departures from linearity can also be due to chance variation.

Summary of Theoretical Distributions

You can use the QQPLOT statement to request Q-Q plots based on the theoretical distributions summarized in Table 6.71.

Table 6.71 QQPLOT Statement Distribution Options

Distribution	Density Function $p(x)$	Range	Parameters		
			Location	Scale	Shape
Beta	$\frac{(x-\theta)^{\alpha-1}(\theta+\sigma-x)^{\beta-1}}{B(\alpha,\beta)\sigma^{(\alpha+\beta-1)}}$	$\theta < x < \theta+\sigma$	θ	σ	α, β
Exponential	$\frac{1}{\sigma}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x \geq \theta$	θ	σ	
Gamma	$\frac{1}{\sigma\Gamma(\alpha)}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}\exp\left(-\frac{x-\theta}{\sigma}\right)$	$x > \theta$	θ	σ	α
Gumbel	$\frac{e^{-(x-\mu)/\sigma}}{\sigma}\exp\left(-e^{-(x-\mu)/\sigma}\right)$	all x	μ	σ	
Lognormal (3-parameter)	$\frac{1}{\sigma\sqrt{2\pi}(x-\theta)}\exp\left(-\frac{(\log(x-\theta)-\zeta)^2}{2\sigma^2}\right)$	$x > \theta$	θ	ζ	σ
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	all x	μ	σ	
Generalized Pareto	$\alpha \neq 0$ $\frac{1}{\sigma}(1-\alpha(x-\theta)/\sigma)^{1/\alpha-1}$ $\alpha = 0$ $\frac{1}{\sigma}\exp(-(x-\theta)/\sigma)$	$x > \theta$	θ	σ	α
Power Function	$\frac{\alpha}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{\alpha-1}$	$x > \theta$	θ	σ	α
Rayleigh	$\frac{x-\theta}{\sigma^2}\exp(-(x-\theta)^2/(2\sigma^2))$	$x \geq \theta$	θ	σ	
Weibull (3-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta}{\sigma}\right)^{c}\right)$	$x > \theta$	θ	σ	c
Weibull (2-parameter)	$\frac{c}{\sigma}\left(\frac{x-\theta_0}{\sigma}\right)^{c-1}\exp\left(-\left(\frac{x-\theta_0}{\sigma}\right)^{c}\right)$	$x > \theta_0$ (known)	θ_0	σ	c

You can request these distributions with the BETA, EXPONENTIAL, GAMMA, LOGNORMAL, NORMAL, WEIBULL, and WEIBULL2 options, respectively. If you do not specify a distribution option, a normal Q-Q plot is created.

Graphical Estimation

You can use Q-Q plots to estimate shape, location, and scale parameters and to estimate percentiles. If you are working with a normal Q-Q plot, you can also estimate certain capability indices.

Shape Parameters

Some distribution options in the QQPLOT statement require that you specify one or two shape parameters in parentheses after the distribution keyword. These are summarized in Table 6.72.

You can visually estimate a shape parameter by specifying a list of values for the shape parameter option. A separate plot is displayed for each value, and you can then select the value that linearizes the point pattern. Alternatively, you can request that the plot be created using an estimated shape parameter. See the entries for the distribution options in the section "Dictionary of Options" on page 502. for details on specification of shape parameters. Example 6.22 and Example 6.23 illustrate shape parameter estimation with lognormal and Weibull Q-Q plots.

Note that for Q-Q plots requested with the WEIBULL2 option, you can estimate the shape parameter c from a linear pattern using the fact that the slope of the pattern is $\frac{1}{c}$. For an illustration, see Example 6.23.

Table 6.72 Shape Parameter Options for the QQPLOT Statement

Distribution Keyword	Mandatory Shape Parameter Option	Range
BETA	ALPHA=α, BETA=β	$\alpha > 0, \beta > 0$
EXPONENTIAL	None	
GAMMA	ALPHA=α	$\alpha > 0$
GUMBEL	None	
LOGNORMAL	SIGMA=σ	$\sigma > 0$
NORMAL	None	
PARETO	ALPHA=α	$\alpha > 0$
POWER	ALPHA=α	$\alpha > 0$
RAYLEIGH	None	
WEIBULL	C=c	$a > 0$
WEIBULL2	None	

Location and Scale Parameters

When the point pattern on a Q-Q plot is linear, its intercept and slope provide estimates of the location and scale parameters. (An exception to this rule is the two-parameter Weibull distribution, for which the intercept and slope are related to the scale and shape parameters.) Table 6.73 shows how the intercept and slope are related to the parameters for each distribution supported by the QQPLOT statement.

Table 6.73 Intercept and Slope of Linear Q-Q Plots

Distribution	Parameters			Linear Pattern	
	Location	Scale	Shape	Intercept	Slope
Beta	θ	σ	α , β	θ	σ
Exponential	θ	σ		θ	σ
Gamma	θ	σ	α	θ	σ
Gumbel	μ	σ		μ	σ
Lognormal	θ	ζ	σ	θ	$\exp(\zeta)$
Normal	μ	σ		μ	σ
Generalized Pareto	θ	σ	α	θ	σ
Power Function	θ	σ	α	θ	σ
Rayleigh	θ	σ		θ	σ
Weibull (3-parameter)	θ	σ	c	θ	σ
Weibull (2-parameter)	θ_0 (known)	σ	c	$\log(\sigma)$	$\frac{1}{c}$

You can enhance a Q-Q plot with a diagonal *distribution reference line* by specifying the parameters that determine the slope and intercept of the line; alternatively, you can request estimates for these parameters. This line is an aid to checking the linearity of the point pattern, and it facilitates parameter estimation. For instance, specifying MU=3 and SIGMA=2 with the NORMAL option requests a line with intercept 3 and slope 2. Specifying SIGMA=1 and C=2 with the WEIBULL2 option requests a line with intercept $\log(1) = 0$ and slope $\frac{1}{2}$.

With the LOGNORMAL and WEIBULL2 options, you can specify the slope directly with the SLOPE= option. That is, for the LOGNORMAL option, specifying THETA=θ_0 and SLOPE=$\exp(\zeta_0)$ gives the same reference line as specifying THETA=θ_0 and ZETA=ζ_0. For the WEIBULL2 option, specifying SIGMA=σ_0 and SLOPE=$\frac{1}{c_0}$ gives the same reference line as specifying SIGMA=σ_0 and C=c_0.

For an example of parameter estimation using a normal Q-Q plot, see "Adding a Distribution Reference Line" on page 495. Example 6.22 illustrates parameter estimation using a lognormal plot, and Example 6.23 illustrates estimation using two-parameter and three-parameter Weibull plots.

Theoretical Percentiles

There are two ways to estimate percentiles from a Q-Q plot:

- Specify the PCTLAXIS option, which adds a percentile axis opposite the theoretical quantile axis. The scale for the percentile axis ranges between 0 and 100 with tick marks at percentile values such as 1, 5, 10, 25, 50, 75, 90, 95, and 99. See Figure 6.41 for an example.
- Specify the PCTLSCALE option, which relabels the horizontal axis tick marks with their percentile equivalents but does not alter their spacing. For example, on a normal Q-Q plot, the tick mark labeled "0" is relabeled as "50" because the 50th percentile corresponds to the zero quantile. See Figure 6.42 for an example.

You can also estimate percentiles using probability plots created with the PROBPLOT statement. See Output 6.20.1 for an example.

Capability Indices

When the point pattern on a normal Q-Q plot is linear, you can estimate the capability indices CPU, CPL, and C_{pk} from the plot, as explained by Rodriguez (1992). This method exploits the fact that the horizontal axis of a Q-Q plot indicates the distance in standard deviation units (multiple of σ) between a measurement or specification limit and the process average.

In particular, one-third the standardized distance between an upper specification limit and the mean is the one-sided capability index *CPU*.

$$CPU = \frac{USL - \mu}{3\sigma}$$

Likewise, one-third the standardized distance between a lower specification limit and the mean is the one-sided capability index *CPL*.

$$CPL = \frac{\mu - LSL}{3\sigma}$$

Consequently, if you *rescale* the quantile axis of a normal Q-Q plot by a factor of three, you can read *CPU* and *CPL* from the horizontal coordinates of the points at which the upper and lower specification lines intersect the point pattern. Because C_{pk} is defined as the minimum of *CPU* and *CPL*, this method also provides a graphical estimate of C_{pk}. For an illustration, see Example 6.24.

SYMBOL Statement Options

In earlier releases of SAS/QC software, graphical features of lower and upper specification lines and diagonal distribution reference lines were controlled with options in the SYMBOL2, SYMBOL3, and SYMBOL4 statements, respectively. These options are still supported, although they have been superseded by options in the QQPLOT and SPEC statements. Table 6.74 summarizes the two sets of options. **NOTE:** These statements have no effect on ODS Graphics output.

Table 6.74 SYMBOL Statement Options

Feature	Statement and Options	Alternative Statement and Options
Symbol markers	**SYMBOL1 Statement**	
character	VALUE=*special-symbol*	
color	COLOR=*color*	
font	FONT=*font*	
height	HEIGHT=*value*	
Lower specification line	**SPEC Statement**	**SYMBOL2 Statement**
position	LSL=*value*	
color	CLSL=*color*	COLOR=*color*
line type	LLSL=*linetype*	LINE=*linetype*
width	WLSL=*value*	WIDTH=*value*
Upper specification line	**SPEC Statement**	**SYMBOL3 Statement**
position	USL=*value*	
color	CUSL=*color*	COLOR=*color*
line type	LUSL=*linetype*	LINE=*linetype*
width	WUSL=*value*	WIDTH=*value*
Target reference line	**SPEC Statement**	
position	TARGET=*value*	
color	CTARGET=*color*	
line type	LTARGET=*linetype*	
width	WTARGET=*value*	
Distribution reference line	**QQPLOT Statement**	**SYMBOL4 Statement**
color	COLOR=*color*	COLOR=*color*
line type	LINE=*linetype*	LINE=*linetype*
width	WIDTH=*value*	WIDTH=*value*

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. QQPLOT options used to control the appearance of traditional graphics are ignored for ODS Graphics output.

When ODS Graphics is in effect, the QQPLOT statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 6.75.

Table 6.75 ODS Graphics Produced by the QQPLOT Statement

ODS Graph Name	Plot Description
QQPlot	Q-Q plot

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Examples: QQPLOT Statement

This section provides advanced examples of the QQPLOT statement.

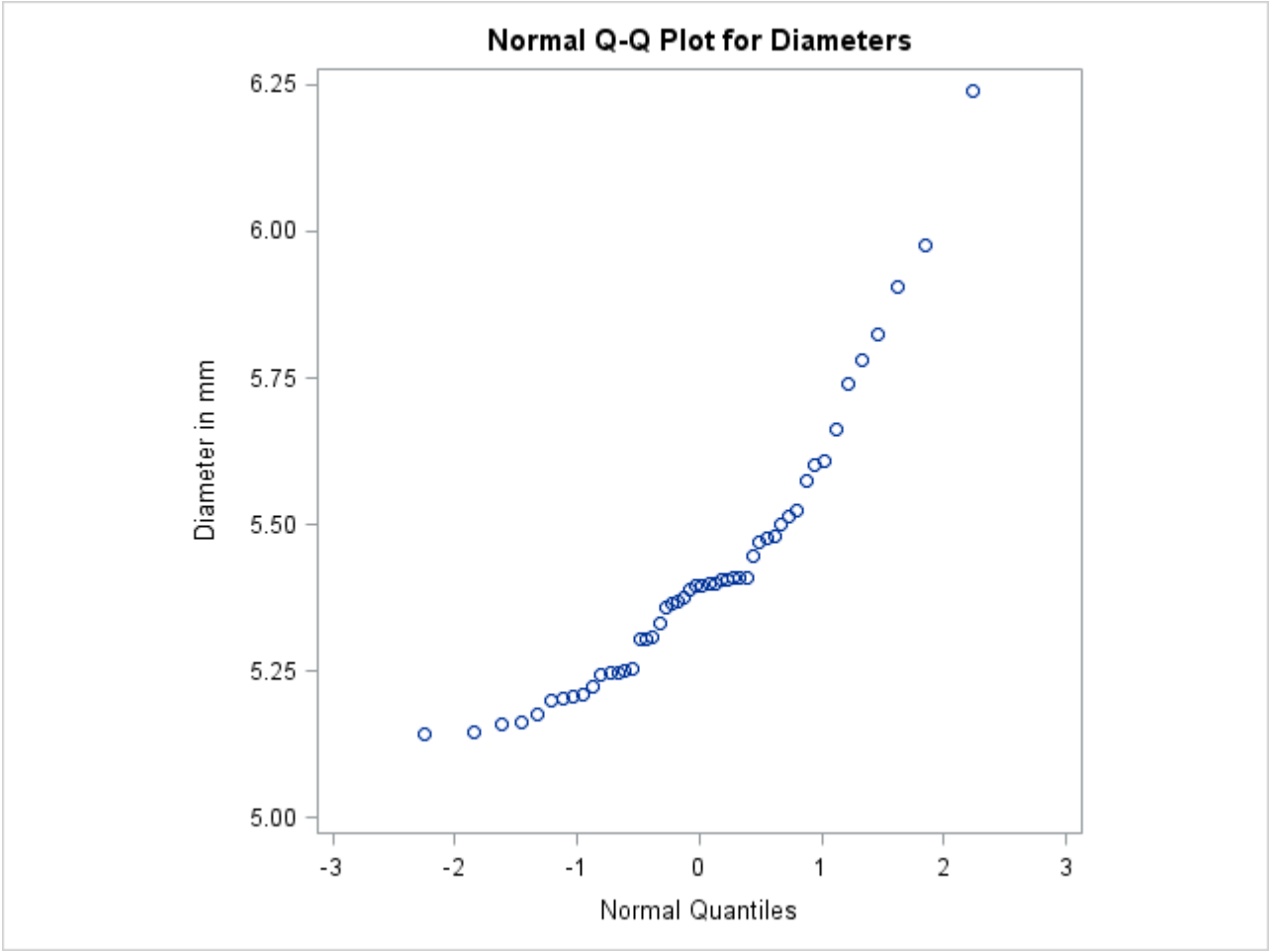
Example 6.21: Interpreting a Normal Q-Q Plot of Nonnormal Data

NOTE: See *Creating Lognormal Q-Q Plots* in the SAS/QC Sample Library.

The following statements produce the normal Q-Q plot in Output 6.21.1:

```
data Measures;
   input Diameter @@;
   label Diameter='Diameter in mm';
   datalines;
5.501  5.251  5.404  5.366  5.445  5.576  5.607
5.200  5.977  5.177  5.332  5.399  5.661  5.512
5.252  5.404  5.739  5.525  5.160  5.410  5.823
5.376  5.202  5.470  5.410  5.394  5.146  5.244
5.309  5.480  5.388  5.399  5.360  5.368  5.394
5.248  5.409  5.304  6.239  5.781  5.247  5.907
5.208  5.143  5.304  5.603  5.164  5.209  5.475
5.223
 ;

title 'Normal Q-Q Plot for Diameters';
proc capability data=Measures noprint;
   qqplot Diameter / normal square odstitle=title;
run;
```

Output 6.21.1 Normal Quantile-Quantile Plot of Nonnormal Data



The nonlinearity of the points in Output 6.21.1 indicates a departure from normality. Because the point pattern is curved with slope increasing from left to right, a theoretical distribution that is skewed to the right, such as a lognormal distribution, should provide a better fit than the normal distribution. The mild curvature suggests that you should examine the data with a series of lognormal Q-Q plots for small values of the shape parameter, as illustrated in the next example.

Example 6.22: Estimating Parameters from Lognormal Plots

This example, which is a continuation of Example 6.21, demonstrates techniques for estimating the shape parameter, location and scale parameters, and theoretical percentiles for a lognormal distribution.

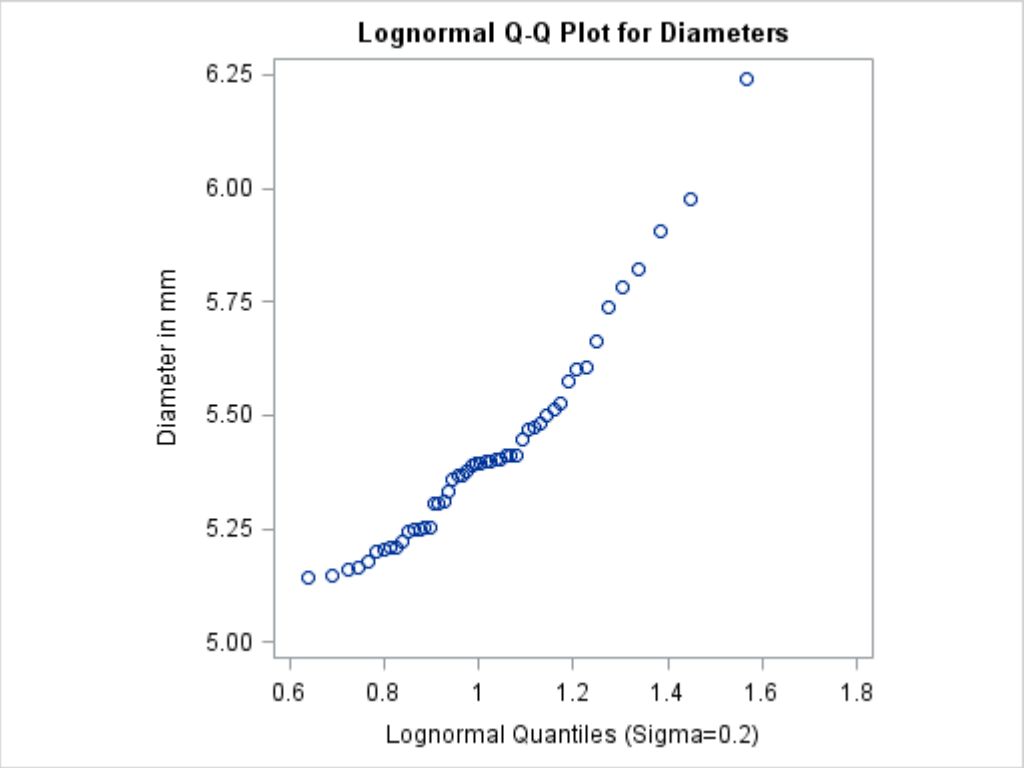
Three-Parameter Lognormal Plots

NOTE: See *Creating Lognormal Q-Q Plots* in the SAS/QC Sample Library.

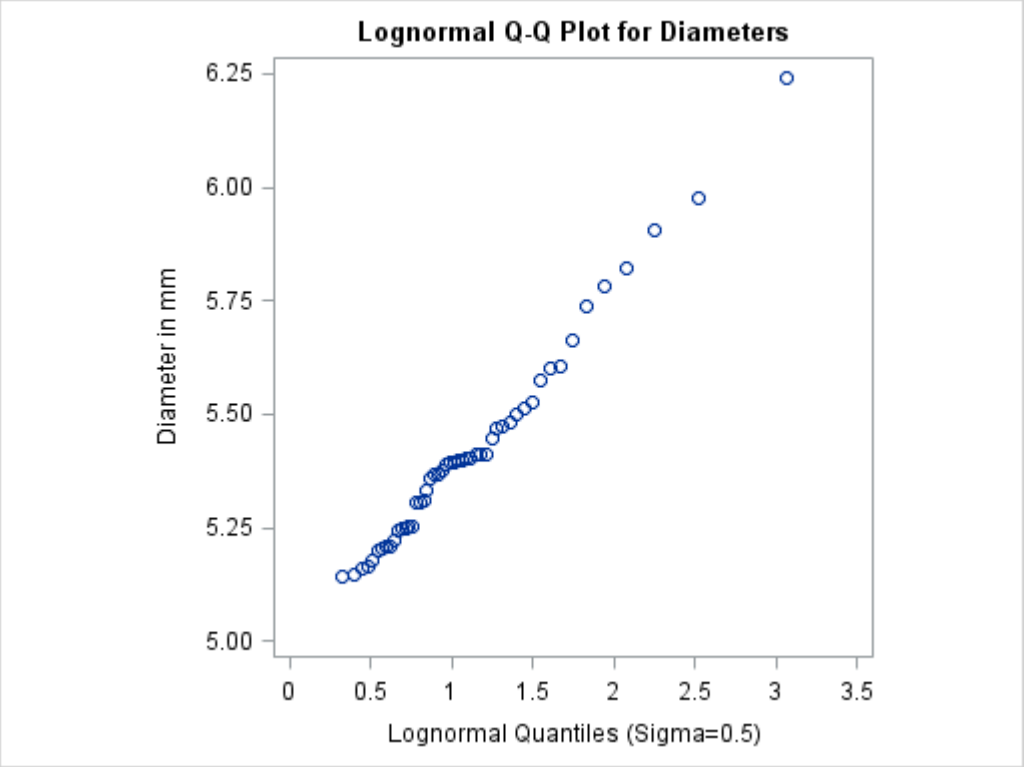
The three-parameter lognormal distribution depends on a threshold parameter θ, a scale parameter ζ, and a shape parameter σ. You can estimate σ from a series of lognormal Q-Q plots with different values of σ. The estimate is the value of σ that linearizes the point pattern. You can then estimate the threshold and scale parameters from the intercept and slope of the point pattern. The following statements create the series of plots in Output 6.22.1 through Output 6.22.3 for σ values of 0.2, 0.5, and 0.8:

```
title 'Lognormal Q-Q Plot for Diameters';
proc capability data=Measures noprint;
   qqplot Diameter / lognormal(sigma=0.2 0.5 0.8)
                     square
                     odstitle=title;
run;
```

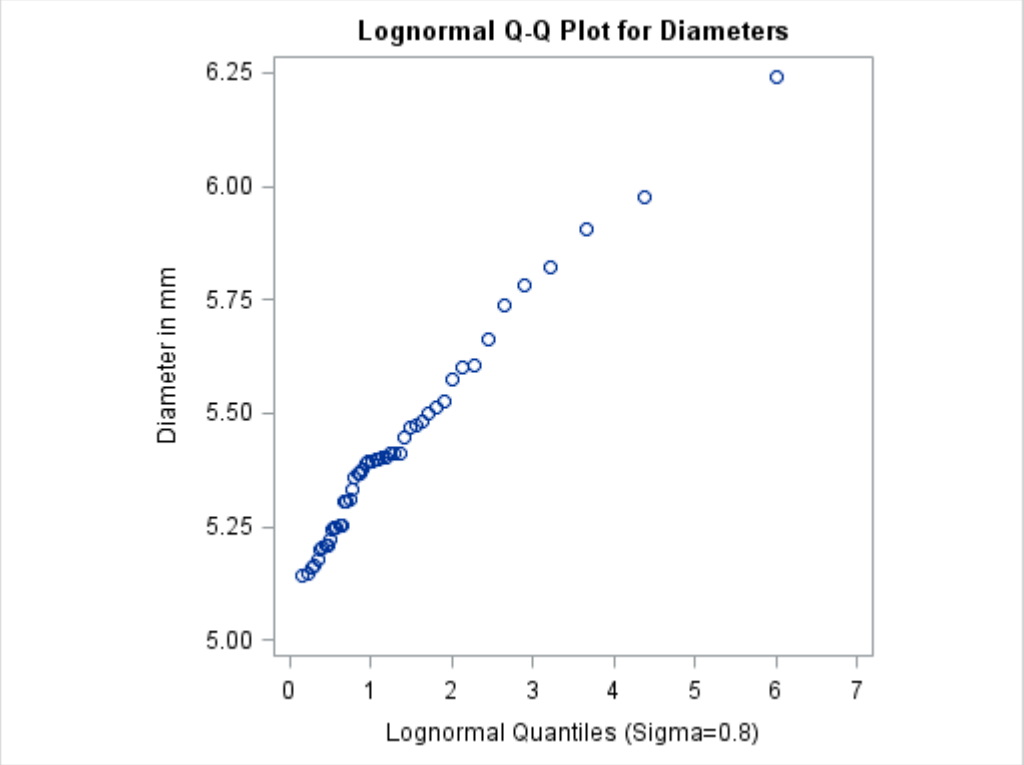
Output 6.22.1 Lognormal Quantile-Quantile Plot (σ =0.2)



Output 6.22.2 Lognormal Quantile-Quantile Plot (σ =0.5)



Output 6.22.3 Lognormal Quantile-Quantile Plot (σ =0.8)



NOTE: You must specify a value for the shape parameter σ for a lognormal Q-Q plot with the SIGMA= option or its alias, the SHAPE= option.

The plot in Output 6.22.2 displays the most linear point pattern, indicating that the lognormal distribution with $\sigma = 0.5$ provides a reasonable fit for the data distribution.

Data with this particular lognormal distribution have the density function

$$p(x) = \begin{cases} \frac{\sqrt{2}}{\sqrt{\pi}(x-\theta)} \exp\left(-2(\log(x-\theta)-\zeta)^2\right) & \text{for } x > \theta \\ 0 & \text{for } x \leq \theta \end{cases}$$

The points in the plot fall on or near the line with intercept θ and slope $\exp(\zeta)$. Based on Output 6.22.2, $\theta \approx 5$ and $\exp(\zeta) \approx \frac{1.2}{3} = 0.4$, giving $\zeta \approx \log(0.4) \approx -0.92$.

Estimating Percentiles

NOTE: See *Creating Lognormal Q-Q Plots* in the SAS/QC Sample Library.

You can use a Q-Q plot to estimate percentiles such as the 95th percentile of the lognormal distribution.[7]

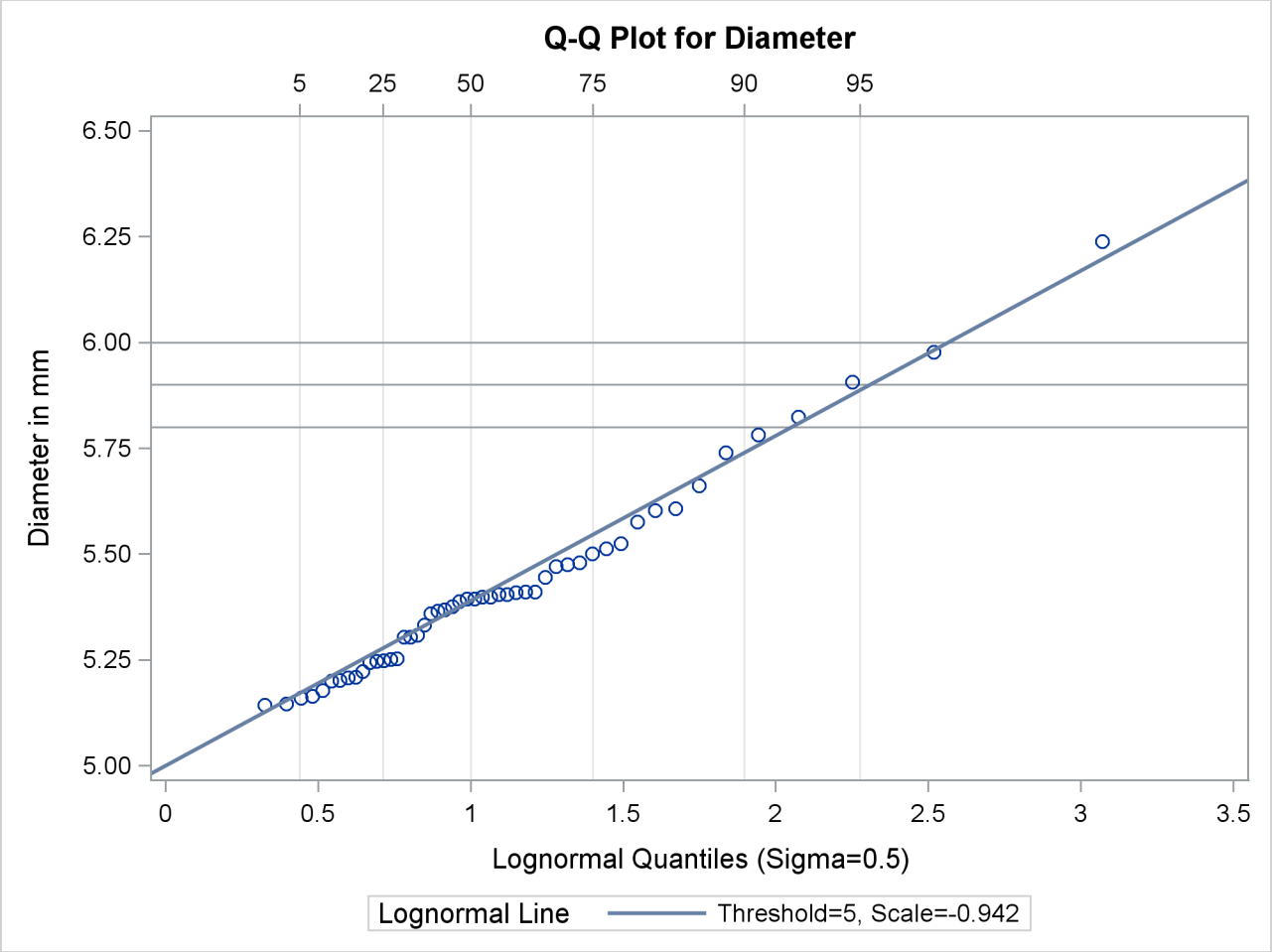
The point pattern in Output 6.22.2 has a slope of approximately 0.39 and an intercept of 5. The following statements reproduce this plot, adding a lognormal reference line with this slope and intercept.

```
title 'Lognormal Q-Q Plot for Diameters';
proc capability data=Measures noprint;
   qqplot Diameter / lognormal(sigma=0.5 theta=5 slope=0.39)
                     pctlaxis(grid)
                     vref  = 5.8 5.9 6.0;
run;
```

The ODS GRAPHICS ON statement specified before the PROC CAPABILITY statement enables ODS Graphics, so the Q-Q plot is created using ODS Graphics instead of traditional graphics. The result is shown in Output 6.22.4.

[7] You can also use a probability plot for this purpose. See Output 6.20.1.

Output 6.22.4 Lognormal Q-Q Plot Identifying Percentiles



The PCTLAXIS option labels the major percentiles, and the GRID option draws percentile axis reference lines. The 95th percentile is 5.9, because the intersection of the distribution reference line and the 95th reference line occurs at this value on the vertical axis.

Alternatively, you can compute this percentile from the estimated lognormal parameters. The 100αth percentile of the lognormal distribution is

$$P_\alpha = \exp(\sigma\Phi^{-1}(\alpha) + \zeta) + \theta$$

where $\Phi^{-1}(\cdot)$ is the inverse cumulative standard normal distribution. Consequently,

$$P_{0.95} \approx \exp\left(\frac{1}{2}\Phi^{-1}(0.95) + \log(0.39)\right) + 5 \approx \exp\left(\frac{1}{2} \times 1.645 - 0.94\right) + 5 \approx 5.89$$

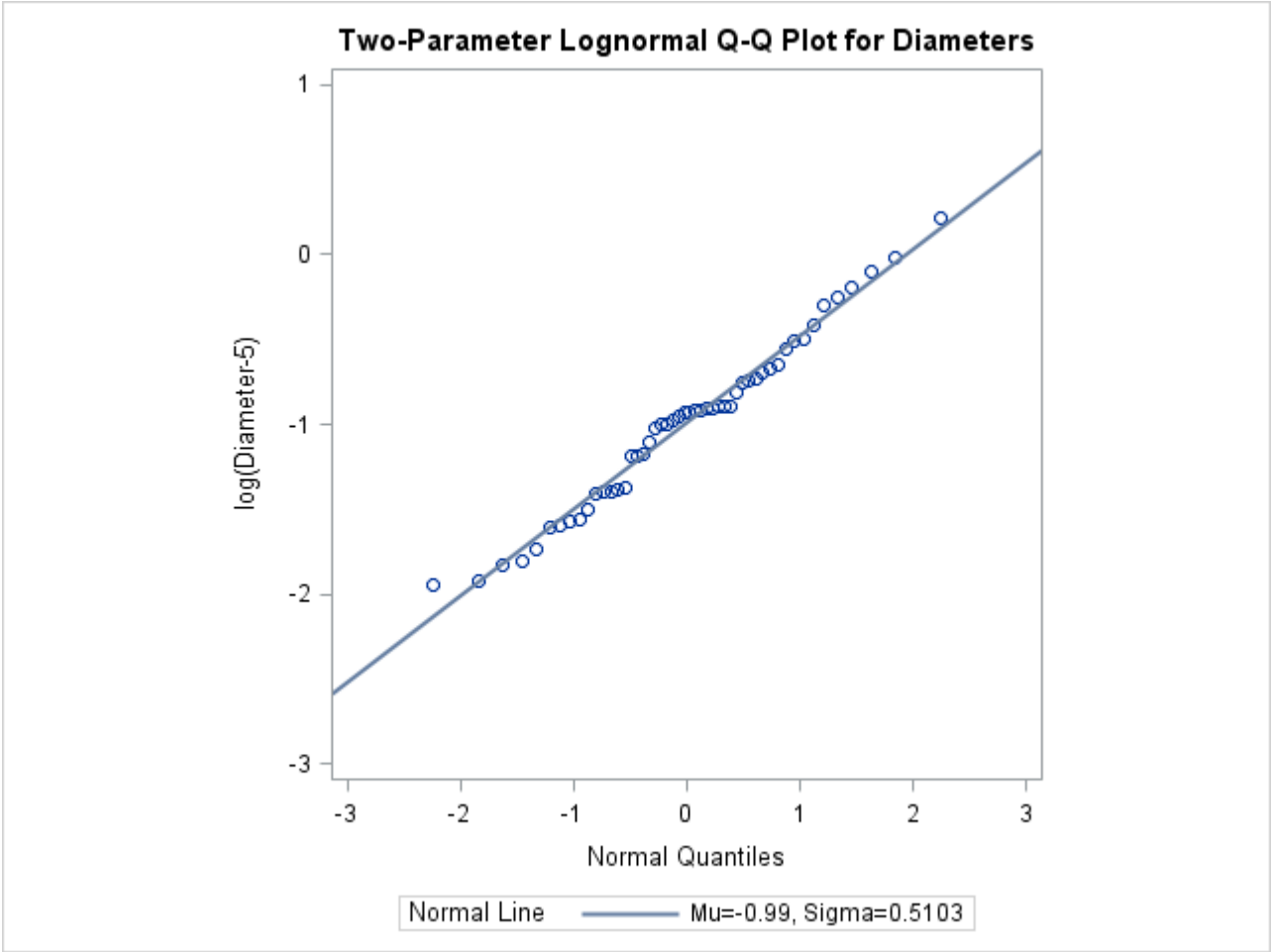
Two-Parameter Lognormal Plots

NOTE: See *Creating Lognormal Q-Q Plots* in the SAS/QC Sample Library.

If a known threshold parameter is available, you can construct a two-parameter lognormal Q-Q plot by subtracting the threshold from the data and requesting a normal Q-Q plot. The following statements create this plot for Diameter, assuming a known threshold of five:

```
data Measures;
   set Measures;
   Logdiam=log(Diameter-5);
   label Logdiam='log(Diameter-5)';
run;

title 'Two-Parameter Lognormal Q-Q Plot for Diameters';
proc capability data=Measures noprint;
   qqplot Logdiam / normal(mu=est sigma=est)
                    square
                    odstitle=title;
run;
```

Output 6.22.5 Two-Parameter Lognormal Q-Q Plot for Diameters



Because the point pattern in Output 6.22.5 is linear, you can estimate the lognormal parameters ζ and σ as the normal plot estimates of μ and σ, which are –0.99 and 0.51. These values correspond to the previous estimates of –0.92 for ζ and 0.5 for σ.

Example 6.23: Comparing Weibull Q-Q Plots

NOTE: See *Creating Weibull Q-Q Plots* in the SAS/QC Sample Library.

This example compares the use of three-parameter and two-parameter Weibull Q-Q plots for the failure times in months for 48 integrated circuits. The times are assumed to follow a Weibull distribution.

```
data Failures;
   input Time @@;
   label Time='Time in Months';
   datalines;
29.42 32.14 30.58 27.50 26.08 29.06 25.10 31.34
29.14 33.96 30.64 27.32 29.86 26.28 29.68 33.76
29.32 30.82 27.26 27.92 30.92 24.64 32.90 35.46
30.28 28.36 25.86 31.36 25.26 36.32 28.58 28.88
26.72 27.42 29.02 27.54 31.60 33.46 26.78 27.82
29.18 27.94 27.66 26.42 31.00 26.64 31.44 32.52
 ;
```

Three-Parameter Weibull Plots

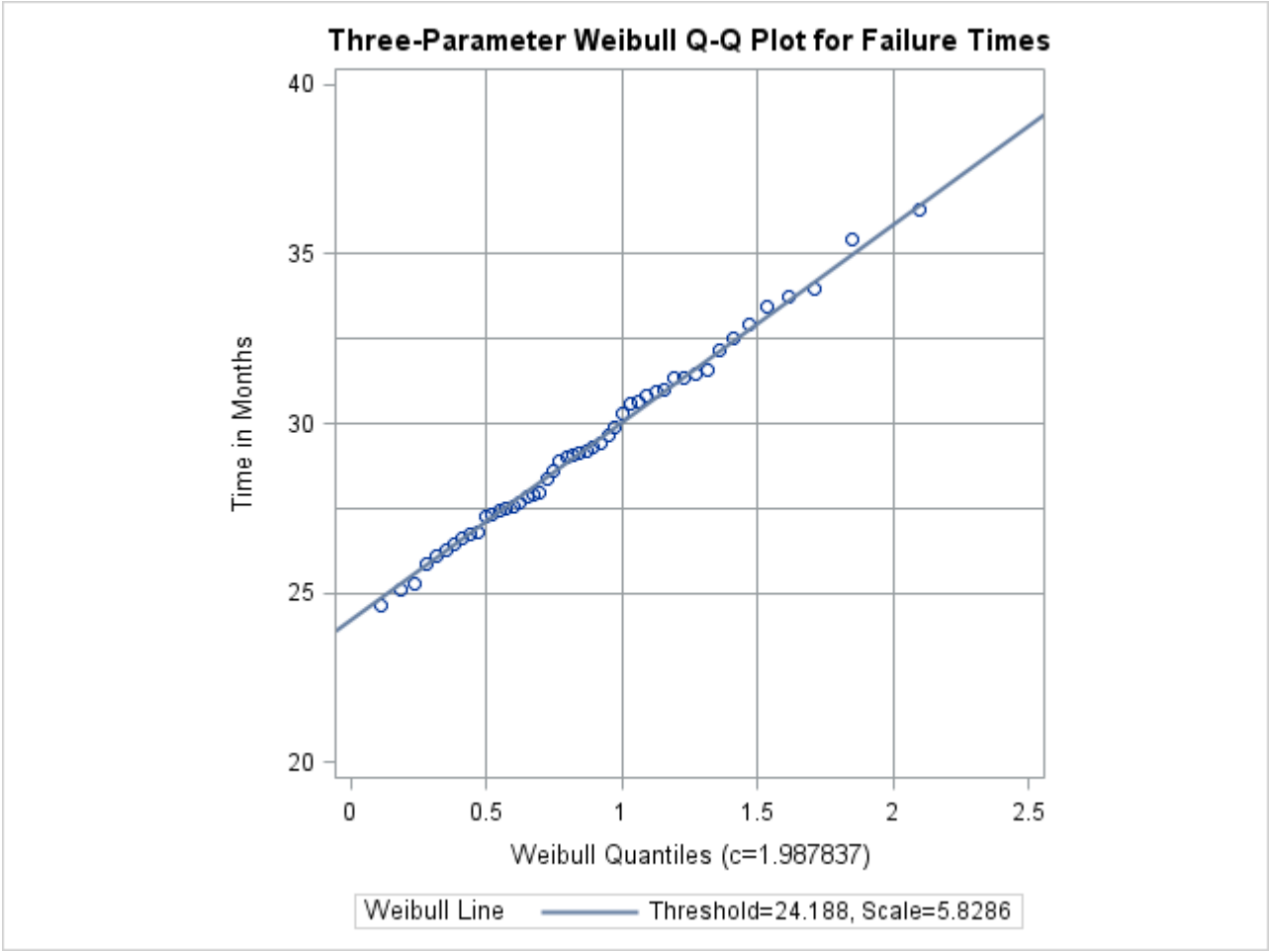
If no assumption is made about the parameters of this distribution, you can use the WEIBULL option to request a three-parameter Weibull plot. As in the previous example, you can visually estimate the shape parameter c by requesting plots for different values of c and choosing the value of c that linearizes the point pattern. Alternatively, you can request a maximum likelihood estimate for c, as illustrated in the following statements produce Weibull plots for c = 1, 2 and 3:

```
title 'Three-Parameter Weibull Q-Q Plot for Failure Times';
proc capability data=Failures noprint;
   qqplot Time / weibull(c=est theta=est sigma=est)
                 square
                 href=0.5 1 1.5 2
                 vref=25 27.5 30 32.5 35
                 odstitle=title;
run;
```

NOTE: When using the WEIBULL option, you must either specify a list of values for the Weibull shape parameter c with the C= option, or you must specify C=EST.

Output 6.23.1 displays the plot for the estimated value c = 1.99. The reference line corresponds to the estimated values for the threshold and scale parameters of ($\hat{\theta}_0$=24.19 and $\hat{\sigma}_0$=5.83, respectively.

Output 6.23.1 Three-Parameter Weibull Q-Q Plot for c = 2



Two-Parameter Weibull Plots

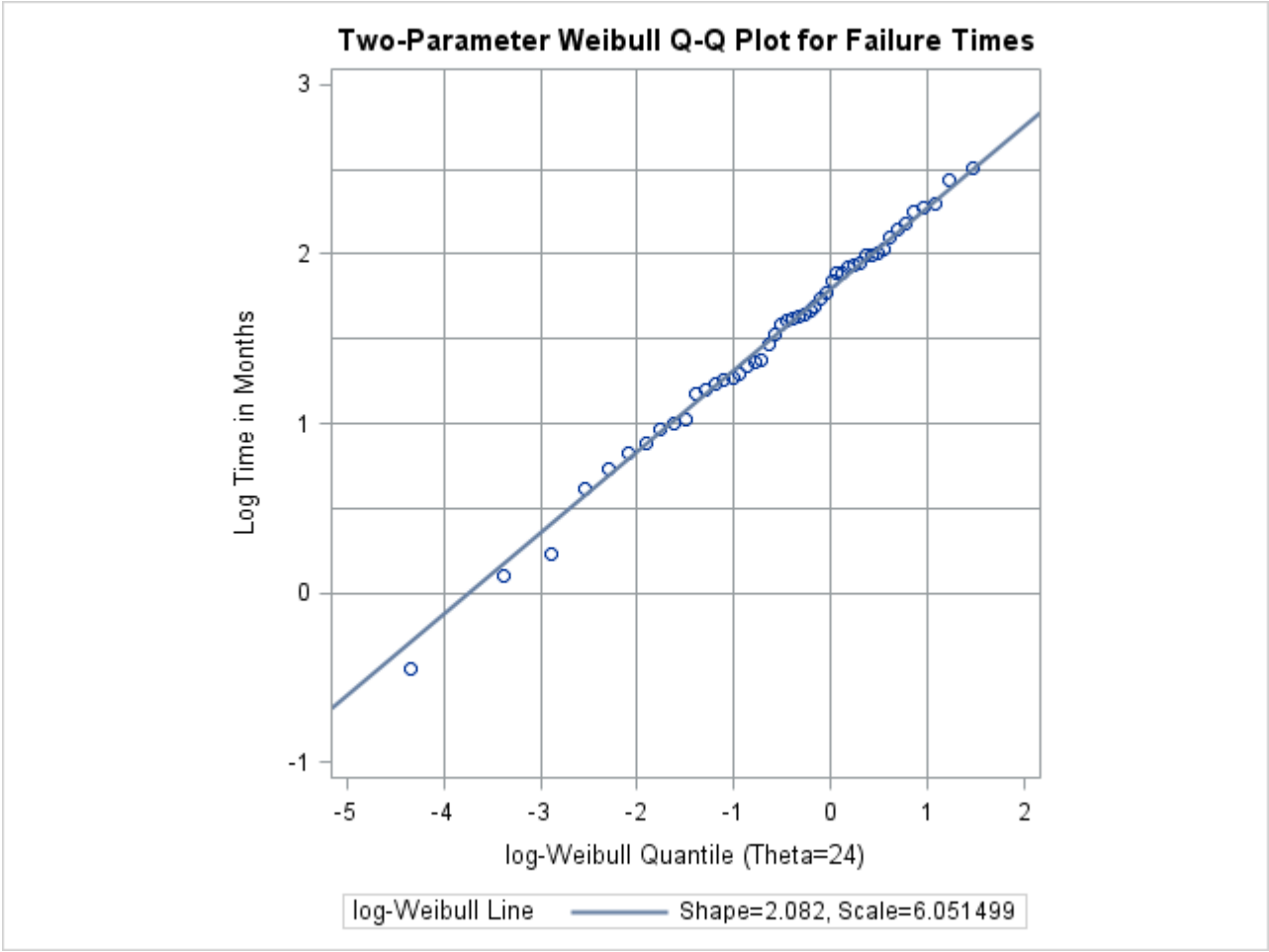
NOTE: See *Creating Weibull Q-Q Plots* in the SAS/QC Sample Library.

Now, suppose it is known that the circuit lifetime is at least 24 months. The following statements use the threshold value $\theta_0 = 24$ to produce the two-parameter Weibull Q-Q plot shown in Output 6.23.2:

```
title 'Two-Parameter Weibull Q-Q Plot for Failure Times';
proc capability data=Failures noprint;
   qqplot Time / weibull2(theta=24 c=est sigma=est) square
                 href= -4 to 1
                 vref= 0 to 2.5 by 0.5
                 odstitle=title;
run;
```

The reference line is based on maximum likelihood estimates $\hat{c}$=2.08 and $\hat{\sigma}$=6.05. These estimates agree with those of the previous example.

Output 6.23.2 Two-Parameter Weibull Q-Q Plot for $\theta_0 = 24$



Example 6.24: Estimating Cpk from a Normal Q-Q Plot

NOTE: See *Creating Normal Q-Q Plots* in the SAS/QC Sample Library.

This example illustrates how you can use a normal Q-Q plot to estimate the capability index C_{pk}. The data used here are the distance measurements provided in the section "Creating a Normal Quantile-Quantile Plot" on page 494.

The linearity of the point pattern in Figure 6.40 indicates that the measurements are normally distributed (recall that normality should be checked when process capability indices are reported). Furthermore, Figure 6.40 shows that the upper specification limit is about 1.7 standard deviation units above the mean, and the lower specification limit is about 1.8 standard deviation units below the mean. Because *CPU* is defined as

$$CPU = \frac{USL - \mu}{3\sigma}$$

and *CPL* is defined as

$$CPL = \frac{\mu - LSL}{3\sigma}$$

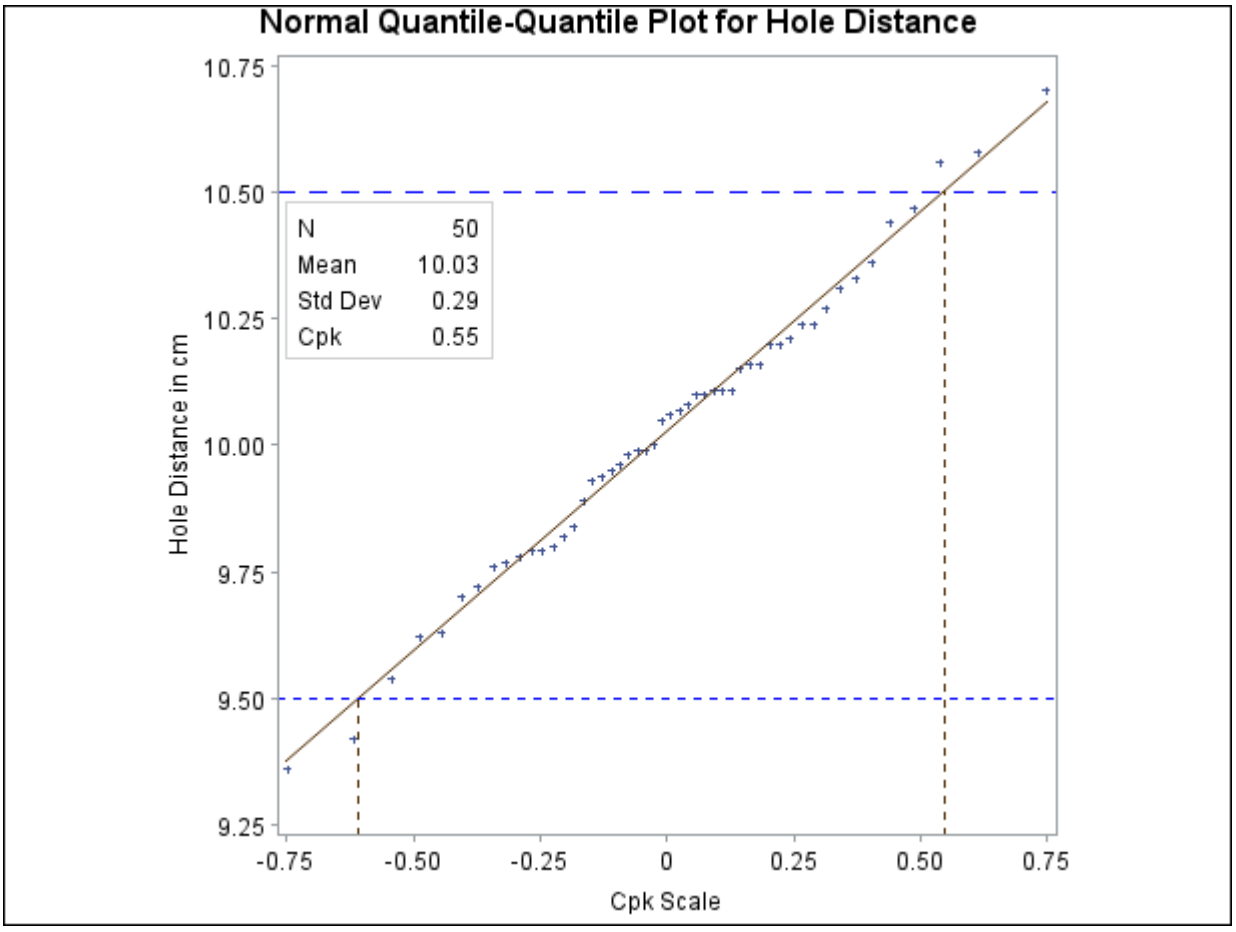
it follows that an estimate of *CPU* is $1.7/3 = 0.57$, and an estimate of *CPL* is $1.8/3 = 0.6$. Thus, except for a factor of three, you can estimate *CPU* and *CPL* from the points of intersection between the specification lines and the point pattern.

The following statements facilitate this type of estimation by creating a Q-Q plot, displayed in Output 6.24.1, in which the horizontal axis is rescaled by a factor of three:

```
ods graphics off;
symbol v=plus;
title "Normal Quantile-Quantile Plot for Hole Distance";
proc capability data=Sheets noprint;
   spec  lsl=9.5   llsl=2   clsl=blue
         usl=10.5  lusl=20  cusl=blue;
   qqplot Distance / normal(mu=est sigma=est cpkscale cpkref)
                     nolegend
                     square;
   inset n mean (5.2) std="Std Dev" (4.2) cpk (4.2) /
         pos=(-0.75,10.48) data refpoint=tl;
run;
```

The CPKSCALE option rescales the horizontal axis, and the CPKREF option adds reference lines indicating the intersections of the distribution reference line and the specification limits.

Output 6.24.1 Normal Q-Q Plot With C_{pk} Scaling



Using this display, you can estimate *CPU* and *CPL* directly from the horizontal axis as 0.55 and 0.60, respectively (the negative sign for –0.60 is ignored). The minimum of these values (0.55) is an estimate of C_{pk}. Note that this estimate agrees with the numerically obtained estimate for C_{pk} that is displayed on the plot with the INSET statement.

See Rodriguez (1992) for further discussion concerning the use of Q-Q plots in process capability analysis.

Dictionary of Common Options: CAPABILITY Procedure

This chapter provides detailed descriptions of options that you can specify in the following chart statements:

- CDFPLOT
- COMPHISTOGRAM
- HISTOGRAM
- PPPLOT
- PROBPLOT
- QQPLOT

As noted, some options are applicable only to comparative plots produced by the COMPHISTOGRAM statement or by another plot statement in conjunction with a CLASS statement.

General Options

ALPHADELTA=*value*

specifies the change in successive estimates of $\hat{\alpha}$ at which iteration terminates in the Newton-Raphson approximation of the maximum likelihood estimate of α for gamma distributions requested with the GAMMA option. Enclose the ALPHADELTA= option in parentheses after the GAMMA keyword. Iteration continues until the change in α is less than the value specified or the number of iterations exceeds the value of the MAXITER= option. The default value is 0.00001.

ALPHAINITIAL=*value*

specifies the initial value for $\hat{\alpha}$ in the Newton-Raphson approximation of the maximum likelihood estimate of α for gamma distributions requested with the GAMMA option. Enclose the ALPHAINITIAL= option in parentheses after the GAMMA keyword. The default value is Thom's approximation of the estimate of α. See Johnson, Kotz, and Balakrishnan (1995).

CDELTA=*value*

specifies the change in successive estimates of c at which iterations terminate in the Newton-Raphson approximation of the maximum likelihood estimate of c for Weibull distributions requested by the WEIBULL option. Enclose the CDELTA= option in parentheses after the WEIBULL keyword. Iteration continues until the change in c between consecutive steps is less than the *value* specified or until the number of iterations exceeds the value of the MAXITER= option. The default value is 0.00001.

CINITIAL=*value*

specifies the initial value for $\hat{c}$ in the Newton-Raphson approximation of the maximum likelihood estimate of c for Weibull distributions requested with the WEIBULL or WEIBULL2 option. The default value is 1.8. See Johnson, Kotz, and Balakrishnan (1995).

CONTENTS='*string*'

specifies the table of contents grouping entry for output produced by the plot statement. You can specify CONTENTS='' to suppress the grouping entry.

CPROP

CPROP=*color* | **EMPTY**

specifies the color for a horizontal bar whose length (relative to the width of the tile) indicates the proportion of the total frequency that is represented by the corresponding cell in a comparative plot. By default, no proportion bars are displayed. You can specify the keyword EMPTY to display empty bars.

For traditional graphics with the GSTYLE system option in effect, you can specify CPROP with no argument to produce proportion bars using an appropriate color from the ODS style. The CPROP option is not available with ODS Graphics.

HAXIS=*value*

specifies the name of an AXIS statement describing the horizontal axis.

HREF=*values*

draws reference lines that are perpendicular to the horizontal axis at the values that you specify. Also see the CHREF= and LHREF= options.

HREFLABELS='*label1*' **...** '*labeln*'

HREFLABEL='*label1*' **...** '*labeln*'

HREFLAB='*label1*' **...** '*labeln*'

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can have up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of HREFLABELS= labels, as described in the following table.

n	Position
1	along top of plot
2	staggered from top to bottom of plot
3	along bottom of plot
4	staggered from bottom to top of plot

By default, HREFLABPOS=1. **NOTE:** HREFLABPOS=2 and HREFLABPOS=4 are not supported for ODS Graphics output.

INTERTILE=*value*

specifies the distance in horizontal percentage screen units between the framed areas, called *tiles*, of a comparative plot. By default, INTERTILE=0.75 percentage screen units. You can specify INTERTILE=0 to create contiguous tiles.

MAXITER=*n*

specifies the maximum number of iterations in the Newton-Raphson approximation of the maximum likelihood estimate of α for gamma distributions requested with the GAMMA option and c for Weibull distributions requested with the WEIBULL and WEIBULL2 options. Enclose the MAXITER= option in parentheses after the GAMMA, WEIBULL, or WEIBULL2 keywords. The default value of n is 20.

NCOLS=*n*

NCOL=*n*

specifies the number of columns per panel in a comparative plot. By default, NCOLS=1 if you specify only one CLASS variable, and NCOLS=2 if you specify two CLASS variables. If you specify two CLASS variables, you can use the NCOLS= option with the NROWS= option.

NOHLABEL

suppresses the label for the horizontal axis. You can use this option to reduce clutter.

NOVLABEL

suppresses the label for the vertical axis. You can use this option to reduce clutter.

NOVTICK

suppresses the tick marks and tick mark labels for the vertical axis. This option also suppresses the label for the vertical axis.

NROWS=*n*

NROW=*n*

specifies the number of rows per panel in a comparative plot. By default, NROWS=2. If you specify two CLASS variables, you can use the NCOLS= option with the NROWS= option.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string***'**

adds a footnote to ODS Graphics output. If you specify the FOOTNOTE (or FOOTNOTE1) keyword, the value of SAS FOOTNOTE statement is used as the graph footnote. If you specify a quoted string, that is used as the footnote. The quoted string can contain either of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	analysis variable name
\l	analysis variable label (or name if the analysis variable has no label)

ODSFOOTNOTE2=FOOTNOTE2 | '*string***'**

adds a secondary footnote to ODS Graphics output. If you specify the FOOTNOTE2 keyword, the value of SAS FOOTNOTE2 statement is used as the secondary graph footnote. If you specify a quoted string, that is used as the secondary footnote. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	analysis variable name
\l	analysis variable label (or name if the analysis variable has no label)

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | LABELFMT | '*string*'
specifies a title for ODS Graphics output.

TITLE (or TITLE1)	uses the value of SAS TITLE statement as the graph title.
NONE	suppresses all titles from the graph.
DEFAULT	uses the default ODS Graphics title (a descriptive title consisting of the plot type and the analysis variable name.)
LABELFMT	uses the default ODS Graphics title with the variable label instead of the variable name.

If you specify a quoted string, that is used as the graph title. The quoted string can contain the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	analysis variable name
\l	analysis variable label (or name if the analysis variable has no label)

ODSTITLE2=TITLE2 | '*string*'
specifies a secondary title for ODS Graphics output. If you specify the TITLE2 keyword, the value of SAS TITLE2 statement is used as the secondary graph title. If you specify a quoted string, that is used as the secondary title. The quoted string can contain the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	analysis variable name
\l	analysis variable label (or name if the analysis variable has no label)

OVERLAY
specifies that plots associated with different levels of a CLASS variable be overlaid onto a single plot, rather than displayed as separate cells in a comparative plot. If you specify the OVERLAY option with one CLASS variable, the output associated with each level of the CLASS variable is overlaid on a single plot. If you specify the OVERLAY option with two CLASS variables, a comparative plot based on the first CLASS variable's levels is produced. Each cell in this comparative plot contains overlaid output associated with the levels of the second CLASS variable.

The OVERLAY option applies only to ODS Graphics output. It is not available in the COMPHISTOGRAM statement.

SCALE=*value*
is an alias for the SIGMA= option for distributions requested by the BETA, EXPONENTIAL, GAMMA, SB, SU, WEIBULL, and WEIBULL2 options and for the ZETA= option for distributions requested by the LOGNORMAL option.

SHAPE=*value*
is an alias for the ALPHA= option for distributions requested by the GAMMA option, for the SIGMA= option for distributions requested by the LOGNORMAL option, and for the C= option for distributions requested by the WEIBULL and WEIBULL2 options.

STATREF=*keyword-list*

draws reference lines at the values of the statistics requested in the *keyword-list*. These reference lines are perpendicular to the horizontal axis in a histogram or cdf plot, and perpendicular to the vertical axis in a probability or Q-Q plot (unless the ROTATE option is specified). The STATREF= option does not apply to the PPPLOT statement.

Valid keywords are listed in the following table.

Keyword	Statistic
MAX	largest value
MEAN	sample mean
MEDIAN | Q2	median (50th percentile)
MIN	smallest value
MODE	most frequent value
P *pctl*	*pctl*th percentile
Q1	lower quartile (25th percentile)
Q3	upper quartile (75th percentile)
factor STD	*factor* standard deviations from the mean

Note that the *factor* specified with the STD keyword can be positive (which puts a reference line above the mean) or negative (below the mean).

Also see the CSTATREF=, LSTATREF=, STATREFLABELS=, and STATREFSUBCHAR= options.

STATREFLABELS='*label1*' **...** '*labeln*'

STATREFLABEL='*label1*' **...** '*labeln*'

STATREFLAB='*label1*' **...** '*labeln*'

specifies labels for the lines requested by the STATREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can have up to 16 characters.

STATREFSUBCHAR='*keyword-list*'

specifies a substitution character (such as #) for labels specified with the STATREFLABELS= option. When the labels are displayed on a graph, the first occurrence of the specified character in each label is replaced with the value of the corresponding STATREF= statistic.

VAXIS=*name*

VAXIS=*value-list*

specifies the name of an AXIS statement describing the vertical axis. In a COMPHISTOGRAM or HISTOGRAM statement, you can alternatively specify a *value-list* for the vertical axis.

VAXISLABEL='*label*'

specifies a label for the vertical axis. Labels can have up to 40 characters.

VREF=*value-list*

draws reference lines perpendicular to the vertical axis at the values specified. Also see the CVREF= and LVREF= options.

VREFLABELS='*label1*'**...**'*labeln*'

VREFLABEL='*label1*'**...**'*labeln*'

VREFLAB='*label1*'**...**'*labeln*'

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can have up to 16 characters.

VREFLABPOS=*n*

specifies the horizontal position of VREFLABELS= labels. If you specify VREFLABPOS=1, the labels are positioned at the left of the plot. If you specify VREFLABPOS=2, the labels are positioned at the right of the plot. By default, VREFLABPOS=1 for traditional graphics and 2 for ODS Graphics.

Options for Traditional Graphics

ANNOKEY

applies the annotation requested with the ANNOTATE= option only to the key cell of a comparative plot. By default, the procedure applies annotation to all of the cells. You can use the KEYLEVEL= option in the CLASS statement or the CLASSKEY= option in the COMPHISTOGRAM statement to specify the key cell.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains annotate variables, as described in *SAS/GRAPH: Reference*, for annotating traditional graphics. The ANNOTATE= data set you specify in the plot statement is used for all plots created by the statement. You can also specify an ANNOTATE= data set in the PROC CAPABILITY statement to enhance all plots created by the procedure (see "ANNOTATE= Data Sets" on page 221).

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement.

CFRAME=*color*

specifies the color for the area that is enclosed by the axes and frame.

CFRAMESIDE=*color*

specifies the color to fill the frame area for the row labels that display along the left side of a comparative plot. This color also fills the frame area for the label of the corresponding CLASS variable, if you associate a label with the variable.

CFRAMETOP=*color*

specifies the color to fill the frame area for the column labels that display across the top of a comparative plot. This color also fills the frame area for the label of the corresponding CLASS variable, if you associate a label with the variable.

CHREF=*color* | *(color-list)*

CH=*color* | *(color-list***)**

specifies the colors for horizontal axis reference lines requested by the HREF= option. If you specify a single color, it is used for all HREF= lines. Otherwise, if there are fewer colors specified than reference lines requested, the remaining lines are displayed with the default reference line color. You can also specify the value *_default* in the color list to request the default color.

COLOR=*color*

COLOR=*color-list*

specifies the color of the curve or reference line associated with a distribution or kernel density estimate. Enclose the COLOR= option in parentheses after a distribution option or the KERNEL option. In a HISTOGRAM statement, you can specify a list of colors in parentheses for multiple density curves.

CSTATREF=*color* | *(color-list)*

specifies the colors for reference lines requested by the STATREF= option. If you specify a single color, it is used for all STATREF= lines. Otherwise, if there are fewer colors specified than reference lines requested, the remaining lines are displayed with the default reference line color. You can also specify the value *_default* in the color list to request the default color.

CTEXT=*color*

CT=*color*

specifies the color for tick mark values and axis labels.

CTEXTSIDE=*color*

specifies the color for the row labels that display along the left side of a comparative plot. If you do not specify the CTEXTSIDE= option, the color specified with the CTEXT= option is used. You can specify the CFRAMESIDE= option to change the background color for the row labels.

CTEXTTOP=*color*

specifies the color for the column labels that display along the left side of a comparative plot. If you do not specify the CTEXTTOP= option, the color specified with the CTEXT= option is used. You can use the CFRAMETOP= option to change the background color for the column labels.

CVREF=*color* | *(color-list)*

CV=*color* | *(color-list)*

specifies the colors for lines requested with the VREF= option. If you specify a single color, it is used for all VREF= lines. Otherwise, if there are fewer colors specified than reference lines requested, the remaining lines are displayed with the default reference line color. You can also specify the value *_default* in the color list to request the default color.

DESCRIPTION='*string*'

DES='*string*'

specifies a description, up to 256 characters long, for the GRSEG catalog entry for a traditional graphics chart. The default value is the analysis variable name.

FONT=*font*

specifies a font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= option takes precedence over the FTEXT= font specified in the GOPTIONS statement. For a list of software fonts, see *SAS/GRAPH: Reference*.

HEIGHT=*value*

specifies the height, in percentage screen units, of text for axis labels, tick mark labels, and legends. This option takes precedence over the HTEXT= option in the GOPTIONS statement.

HMINOR=*n*

HM=*n*

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. By default, HMINOR=0.

INFONT=*font*

specifies a font to use for text inside the framed areas of the plot. The INFONT= option takes precedence over the FTEXT= option in the GOPTIONS statement. For a list of software fonts, see *SAS/GRAPH: Reference*.

INHEIGHT=*value*

specifies the height, in percentage screen units, of text used inside the framed areas of the plot. If you do not specify the INHEIGHT= option, the height specified with the HEIGHT= option is used.

L=*linetype*

L=*linetype-list*

specifies the line type of the curve or reference line associated with a distribution or kernel density estimate. Enclose the L= option in parentheses after the distribution option or the KERNEL option. In a HISTOGRAM statement, you can specify a list of line types in parentheses for multiple density curves.

LHREF=*linetype* | *linetype-list*

LH=*linetype* | *linetype-list*

specifies the line types for the reference lines that you request with the HREF= option. If you specify a single line type, it is used for all HREF= lines. Otherwise, if there are fewer line types specified than reference lines requested, the remaining lines are displayed with the default reference line type. You can also specify line type 0 to request the default color.

LSTATREF=*linetype* | *linetype-list*

specifies the line types for the reference lines that you request with the STATREF= option. If you specify a single line type, it is used for all STATREF= lines. Otherwise, if there are fewer line types specified than reference lines requested, the remaining lines are displayed with the default reference line type. You can also specify line type 0 to request the default color.

LVREF=*linetype* | *linetype-list*

LV=*linetype* | *linetype-list*

specifies the line types for lines requested with the VREF= option. If you specify a single line type, it is used for all VREF= lines. Otherwise, if there are fewer line types specified than reference lines requested, the remaining lines are displayed with the default reference line type. You can also specify line type 0 to request the default color.

NAME='*string*'

specifies the name of the GRSEG catalog entry for a traditional graphics plot, and the name of the graphics output file if one is created. The name can be up to 256 characters long, but the GRSEG name is truncated to eight characters. The default value is 'CAPABILI'.

NOFRAME
suppresses the frame around the subplot area.

TURNVLABELS
TURNVLABEL
turns the characters in the vertical axis labels so that they display vertically.

VMINOR=*n*
VM=*n*
specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. The default is zero.

W=*value*
W=*value-list*
specifies the width in pixels of the curve or reference line associated with a distribution or kernel density estimate. Enclose the W= option in parentheses after the distribution option or the KERNEL option. In a HISTOGRAM statement, you can specify a list of widths in parentheses for multiple density curves.

WAXIS=*n*
specifies the line thickness, in pixels, for the axes and frame.

Options for Legacy Line Printer Charts

HREFCHAR='*character***'**
specifies the character used to form the lines requested by the HREF= option for a line printer chart. The default is the vertical bar (|).

VREFCHAR='*character***'**
VREF= option for a line printer chart. specifies the character used to form the lines requested by the The default is the hyphen (-).

References

Bai, D. S., and Choi, I. S. (1997). "Process Capability Indices for Skewed Populations." Unpublished manuscript, Korean Advanced Institute of Science and Technology, Taejon, Korea.

Bissell, A. F. (1990). "How Reliable Is Your Capability Index?" *Journal of the Royal Statistical Society, Series C* 39:331–340.

Blom, G. (1958). *Statistical Estimates and Transformed Beta Variables*. New York: John Wiley & Sons.

Bowman, K. O., and Shenton, L. R. (1983). "Johnson's System of Distributions." In *Encyclopedia of Statistical Sciences*, vol. 4, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons.

Boyles, R. A. (1991). "The Taguchi Capability Index." *Journal of Quality Technology* 23:107–126.

Boyles, R. A. (1992). *Cpm for Asymmetrical Tolerances*. Technical report, Precision Castparts Corp., Portland, OR.

Boyles, R. A. (1994). "Process Capability with Asymmetric Tolerances." *Communications in Statistics—Simulation and Computation* 23:615–643.

Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. (1983). *Graphical Methods for Data Analysis*. Belmont, CA: Wadsworth International Group.

Chen, H. F., and Kotz, S. (1996). "An Asymptotic Distribution of Wright's Process Capability Index Sensitive to Skewness." *Journal of Statistical Computation and Simulation* 55:147–158.

Chen, K. S. (1998). "Incapability Index with Asymmetric Tolerances." *Statistica Sinica* 8:253–262.

Chou, Y., Owen, D. B., and Borrego, S. A. (1990). "Lower Confidence Limits on Process Capability Indices." *Journal of Quality Technology* 22:223–229; corrigenda, 24, 251.

Cohen, A. C. (1951). "Estimating Parameters of Logarithmic-Normal Distributions by Maximum Likelihood." *Journal of the American Statistical Association* 46:206–212.

Croux, C., and Rousseeuw, P. J. (1992). "Time-Efficient Algorithms for Two Highly Robust Estimators of Scale." *Computational Statistics* 1:411–428.

D'Agostino, R. B., and Stephens, M., eds. (1986). *Goodness-of-Fit Techniques*. New York: Marcel Dekker.

Dixon, W. J., and Tukey, J. W. (1968). "Approximate Behavior of the Distribution of Winsorized *t* (Trimming/Winsorization 2)." *Technometrics* 10:83–98.

Ekvall, D. N., and Juran, J. M. (1974). "Manufacturing Planning." In *Quality Control Handbook*, 3rd ed., edited by J. M. Juran. New York: McGraw-Hill.

Elandt, R. C. (1961). "The Folded Normal Distribution: Two Methods of Estimating Parameters from Moments." *Technometrics* 3:551–562.

Fowlkes, E. B. (1987). *A Folio of Distributions: A Collection of Theoretical Quantile-Quantile Plots*. New York: Marcel Dekker.

Gnanadesikan, R. (1997). *Statistical Data Analysis of Multivariate Observations*. New York: John Wiley & Sons.

Gupta, A. K., and Kotz, S. (1997). "A New Process Capability Index." *Metrika* 45:213–224.

Hahn, G. J. (1969). "Factors for Calculating Two-Sided Prediction Intervals for Samples from a Normal Distribution." *Journal of the American Statistical Association* 64:878–898.

Hahn, G. J. (1970a). "Additional Factors for Calculating Prediction Intervals for Samples from a Normal Distribution." *Journal of the American Statistical Association* 65:1668–1676.

Hahn, G. J. (1970b). "Statistical Intervals for a Normal Population, Part 2: Formulas, Assumptions, Some Derivations." *Journal of Quality Technology* 2:195–206.

Hahn, G. J., and Meeker, W. Q. (1991). *Statistical Intervals: A Guide for Practitioners*. New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994). *Continuous Univariate Distributions.* 2nd ed. Vol. 1. New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1995). *Continuous Univariate Distributions.* 2nd ed. Vol. 2. New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Pearn, W. L. (1994). "Flexible Process Capability Indices." *Pakistan Journal of Statistics* 10:23–31.

Kane, V. E. (1986). "Process Capability Indices." *Journal of Quality Technology* 1:41–52.

Kotz, S., and Johnson, N. L. (1993). *Process Capability Indices*. London: Chapman & Hall.

Kotz, S., and Lovelace, C. R. (1998). *Process Capability Indices in Theory and Practice*. London: Edward Arnold.

Krishnamoorthy, K., and Mathew, T. (2009). *Statistical Tolerance Regions: Theory, Applications, and Computation*. Hoboken, NJ: John Wiley & Sons.

Kushler, R. H., and Hurley, P. (1992). "Confidence Bounds for Capability Indices." *Journal of Quality Technology* 24:188–195.

Lehmann, E. L., and D'Abrera, H. J. M. (1975). *Nonparametrics: Statistical Methods Based on Ranks*. San Francisco: Holden-Day.

Luceño, A. (1996). "A Process Capability Index with Reliable Confidence Intervals." *Communications in Statistics—Simulation and Computation* 25:235–245.

Marcucci, M. O., and Beazley, C. F. (1988). "Capability Indices: Process Performance Measures." *Transactions of ASQC Congress* 42:516–523.

Montgomery, D. C. (1996). *Introduction to Statistical Quality Control*. 3rd ed. New York: John Wiley & Sons.

Odeh, R. E., and Owen, D. B. (1980). *Tables for Normal Tolerance Limits, Sampling Plans, and Screening*. New York: Marcel Dekker.

Owen, D. B., and Hua, T. A. (1977). "Tables of Confidence Limits on the Tail Area of the Normal Distribution." *Communications in Statistics—Simulation and Computation* 6:285–311.

Pearn, W. L., Kotz, S., and Johnson, N. L. (1992). "Distributional and Inferential Properties of Process Capability Indices." *Journal of Quality Technology* 24:216–231.

Rodriguez, R. N. (1992). "Recent Developments in Process Capability Analysis." *Journal of Quality Technology* 24:176–187.

Rodriguez, R. N., and Bynum, R. A. (1992). "Examples of Short Run Process Control Methods with the SHEWHART Procedure in SAS/QC Software." Unpublished manuscript available from the authors.

Rousseeuw, P. J., and Croux, C. (1993). "Alternatives to the Median Absolute Deviation." *Journal of the American Statistical Association* 88:1273–1283.

Royston, J. P. (1992). "Approximating the Shapiro-Wilk W Test for Nonnormality." *Statistics and Computing* 2:117–119.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. New York: Chapman & Hall.

Slifker, J. F., and Shapiro, S. S. (1980). "The Johnson System: Selection and Parameter Estimation." *Technometrics* 22:239–246.

Terrell, G. R., and Scott, D. W. (1985). "Oversmoothed Nonparametric Density Estimates." *Journal of the American Statistical Association* 80:209–214.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Tukey, J. W., and McLaughlin, D. H. (1963). "Less Vulnerable Confidence and Significance Procedures for Location Based on a Single Sample: Trimming/Winsorization 1." *Sankhyā, Series A* 25:331–352.

Vännmann, K. (1995). "A Unified Approach to Capability Indices." *Statistica Sinica* 5:805–820.

Vännmann, K. (1997). "A General Class of Capability Indices in the Case of Asymmetric Tolerances." *Communications in Statistics—Theory and Methods* 26:2049–2072.

Velleman, P. F., and Hoaglin, D. C. (1981). *Applications, Basics, and Computing of Exploratory Data Analysis*. Boston: Duxbury Press.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986). *Modern Methods for Quality Control and Improvement*. New York: John Wiley & Sons.

Wainer, H. (1974). "The Suspended Rootogram and Other Visual Displays: An Empirical Validation." *American Statistician* 28:143–145.

Wilk, M. B., and Gnanadesikan, R. (1968). "Probability Plotting Methods for the Analysis of Data." *Biometrika* 49:525–545.

Wright, P. A. (1995). "A Process Capability Index Sensitive to Skewness." *Journal of Statistical Computation and Simulation* 52:195–203.

Zhang, N. F., Stenback, G. A., and Wardrop, D. M. (1990). "Interval Estimation of Process Capability Index Cpk." *Communications in Statistics—Theory and Methods* 19:4455–4470.

Chapter 7
The CUSUM Procedure

Contents

Introduction: CUSUM Procedure **548**
Learning about the CUSUM Procedure 549
PROC CUSUM Statement **549**
Overview: PROC CUSUM Statement 549
Syntax: PROC CUSUM Statement 550
BY Statement 552
Input and Output Data Sets: CUSUM Procedure 553
XCHART Statement: CUSUM Procedure **554**
Overview: XCHART Statement 554
Getting Started: XCHART Statement 555
Creating a V-Mask Cusum Chart from Raw Data 555
Creating a V-Mask Cusum Chart from Subgroup Summary Data 558
Saving Summary Statistics 560
Creating a One-Sided Cusum Chart with a Decision Interval 561
Saving Cusum Scheme Parameters 565
Reading Cusum Scheme Parameters 567
Syntax: XCHART Statement 569
Summary of Options 570
Dictionary of Special Options 579
Details: XCHART Statement 585
Basic Notation for Cusum Charts 585
Formulas for Cumulative Sums 585
Defining the Decision Interval for a One-Sided Cusum Scheme 588
Defining the V-Mask for a Two-Sided Cusum Scheme 588
Designing a Cusum Scheme 590
Cusum Charts Compared with Shewhart Charts 593
Methods for Estimating the Standard Deviation 594
Output Data Sets 596
ODS Tables 599
ODS Graphics 599
Input Data Sets 600
Missing Values 602
Examples: XCHART Statement 602
Example 7.1: Cusum and Standard Deviation Charts 602
Example 7.2: Upper and Lower One-Sided Cusum Charts 605
Example 7.3: Combined Shewhart–Cusum Scheme 607

INSET Statement: CUSUM Procedure **610**
Overview: INSET Statement 610
Getting Started: INSET Statement 610
Syntax: INSET Statement 612
References **614**

Introduction: CUSUM Procedure

The CUSUM procedure creates cumulative sum control charts, also known as *cusum charts*, which display cumulative sums of the deviations of measurements or subgroup means from a target value. Cusum charts are used to decide whether a process is in statistical control by detecting a shift in the process mean.

You can use the CUSUM procedure to

- apply a *one-sided cusum scheme*, also referred to as a *decision interval scheme*, which detects a shift in one direction from the target mean. You can specify the scheme with the decision interval h and the reference value k.
- apply a *two-sided cusum scheme* with a V-mask, which detects a shift in either direction from the target mean. You can specify the scheme with geometric parameters (h and k) for the V-mask or with error probabilities (α and β).
- implement cusum schemes graphically or computationally
- specify the shift to be detected as a multiple of standard error or in data units
- estimate the process standard deviation σ using a variety of methods
- compute average run lengths (ARLs)
- read raw data (actual measurements) or summarized data (subgroup means and standard deviations)
- analyze multiple process variables. If used with a BY statement, PROC CUSUM produces charts separately for groups of observations.
- save cusums and cusum scheme parameters in output data sets
- tabulate the information displayed on the chart
- read cusum scheme parameters from an input data set
- read numeric- or character-valued subgroup variables
- display subgroups with date and time formats
- enhance cusum charts with special legends and symbol markers that indicate the levels of stratification variables
- superimpose plotted points with stars (polygons) whose vertices indicate the values of multivariate data related to the process

- display a trend chart below the cusum chart that plots a systematic or fitted trend in the data
- produce charts as traditional graphics, ODS Graphics output, or legacy line printer charts. Line printer charts can use special formatting characters that improve the appearance of the chart. Traditional graphics can be annotated, saved, and replayed.

Learning about the CUSUM Procedure

If you are using the CUSUM procedure for the first time, begin by reading "PROC CUSUM Statement" on page 549 to learn about input data sets. Then turn to "Getting Started: XCHART Statement" on page 555 in "XCHART Statement: CUSUM Procedure" on page 554. This chapter also provides syntax information and advanced examples.

If you are not familiar with cusum charts, read "Formulas for Cumulative Sums" on page 585 "Defining the Decision Interval for a One-Sided Cusum Scheme" on page 588 and "Defining the V-Mask for a Two-Sided Cusum Scheme" on page 588 in the section "Details: XCHART Statement" on page 585. References lists articles and textbooks that provide more detailed information on cusum charts. The expository articles by Lucas (1976) and Goel (1982) and the textbooks by Montgomery (1996) and Ryan (1989) are recommended introductory reading.

PROC CUSUM Statement

Overview: PROC CUSUM Statement

The PROC CUSUM statement starts the CUSUM procedure and it identifies input data sets.

After the PROC CUSUM statement, you provide an XCHART statement that specifies the cusum chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request a one-sided (decision interval) cusum chart:

```
proc cusum data=values;
   xchart weight*lot / scheme = onesided
                       mu0    = 8.100
                       sigma0 = 0.050
                       delta  = 1
                       h      = 2.2
                       k      = 0.5;
run;
```

In this example, the DATA= option specifies an input data set (values) that contains the *process* measurement variable weight and the *subgroup-variable* lot.

You can use options in the PROC CUSUM statement to do the following:

- specify input data sets containing variables to be analyzed, parameters for cusum schemes, or annotation information

- specify a graphics catalog for saving traditional graphics output
- specify that line printer charts are to be produced
- define characters used for features on line printer charts

In addition to the XCHART statement, you can provide BY statements, ID statements, TITLE statements, and FOOTNOTE statements. If you are producing traditional graphics, you can also provide graphics enhancement statements, such as SYMBOL*n* statements, which are described in *SAS/GRAPH: Reference*.

See Chapter 4, "SAS/QC Graphics," for a detailed discussion of the alternatives available for producing charts with SAS/QC procedures.

NOTE: If you are using the CUSUM procedure for the first time, you should read both this chapter and the section "Getting Started: XCHART Statement" on page 555 in "XCHART Statement: CUSUM Procedure" on page 554.

Syntax: PROC CUSUM Statement

The syntax for the PROC CUSUM statement is as follows:

PROC CUSUM < *options* > **;**

The PROC CUSUM statement starts the CUSUM procedure, and it optionally identifies various data sets. You can specify the following options in the PROC CUSUM statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains appropriate annotate variables, as described in *SAS/GRAPH: Reference*. The ANNOTATE= option enables you to add features to a cusum chart (for example, labels that explain out-of-control points). The ANNOTATE= data set is used only when the chart is created as traditional graphics; it is ignored when the LINEPRINTER option is specified or ODS Graphics is enabled. The data set specified with the ANNOTATE= option in the PROC CUSUM statement is a "global" annotate data set in the sense that the information in this data set is displayed on every chart produced in the current run of the CUSUM procedure.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an input data set that contains appropriate annotate variables that add features to the trend chart (secondary chart) produced with the TRENDVAR= option in the XCHART statement. This option applies only when you produce traditional graphics.

DATA=*SAS-data-set*

names an input data set that contains raw data (measurements) as observations. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). The DATA= data set can contain more than one observation for each value of the *subgroup-variable*.

You cannot use a DATA= data set with a HISTORY= data set. If you do not specify a DATA= or HISTORY= data set, PROC CUSUM uses the most recently created data set as a DATA= data set. For more information, see "DATA= Data Set" on page 600

FORMCHAR(*index***)='***string***'**

defines characters used for features on legacy line printer charts, where *index* is a list of numbers ranging from 1 to 17 and *string* is a character or hexadecimal string. This option applies only if you also specify the LINEPRINTER option.

The *index* identifies which features are controlled with the *string* characters, as described in Table 7.1. If you specify the FORMCHAR= option and omit the *index*, the *string* controls all 17 features.

Table 7.1 FORMCHAR= Features

Value of *index*	Description of Character	Chart Feature
1	Vertical bar	Frame
2	Horizontal bar	Frame, central line
3	Box character (upper left)	Frame
4	Box character (upper middle)	Serifs, tick (horizontal axis)
5	Box character (upper right)	Frame
6	Box character (middle left)	Not used
7	Box character (middle middle)	Serifs
8	Box character (middle right)	Tick (vertical axis)
9	Box character (lower left)	Frame
10	Box character (lower middle)	Serifs
11	Box character (lower right)	Frame
12	Vertical bar	Control limits
13	Horizontal bar	Control limits
14	Box character (upper right)	Control limits
15	Box character (lower left)	Control limits
16	Box character (lower right)	Control limits
17	Box character (upper left)	Control limits

Not all printers can produce the characters in the preceding list. By default, the form character list specified by the SAS system option FORMCHAR= is used; otherwise, the default is FORMCHAR='|—-|+|—|====='. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that you can use the FORMCHAR= option to temporarily override the values of the SAS system FORMCHAR= option. The values of the SAS system FORMCHAR= option are not altered by the FORMCHAR= option in the PROC CUSUM statement.

GOUT=*graphics-catalog*

specifies the graphics catalog for traditional graphics output from PROC CUSUM. This is useful if you want to save the output. The GOUT= option is used only when the chart is created using traditional graphics; it is ignored when the LINEPRINTER option is specified or ODS Graphics is enabled.

HISTORY=*SAS-data-set*

HIST=*SAS-data-set*

names an input data set that contains subgroup summary statistics (means, standard deviations, and sample sizes). Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC CUSUM or PROC SHEWHART, but it can also be created with a SAS summarization procedure such as PROC MEANS.

If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). A HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set together with a DATA= data set. If you do not specify a HISTORY= or DATA= data set, PROC CUSUM uses the most recently created data set as a DATA= data set. For more information on HISTORY= data sets, see "HISTORY= Data Set" on page 601.

LIMITS=*SAS-data-set*

names an input data set that contains a set of decision interval or V-mask parameters. Each observation in a LIMITS= data set contains the parameters for a *process*.

If you are using SAS 6.09 or an earlier release of SAS/QC software, you must specify the options READLIMITS or READINDEX= in the XCHART statement to read the parameters from the LIMITS= data set. In SAS 6.10 and later releases, these options are not needed.

For details about the variables needed in a LIMITS= data set, see "LIMITS= Data Set" on page 600. If you do not provide a LIMITS= data set, you must specify the parameters with options in the XCHART statement.

LINEPRINTER

requests that legacy line printer charts be produced.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CUSUM to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

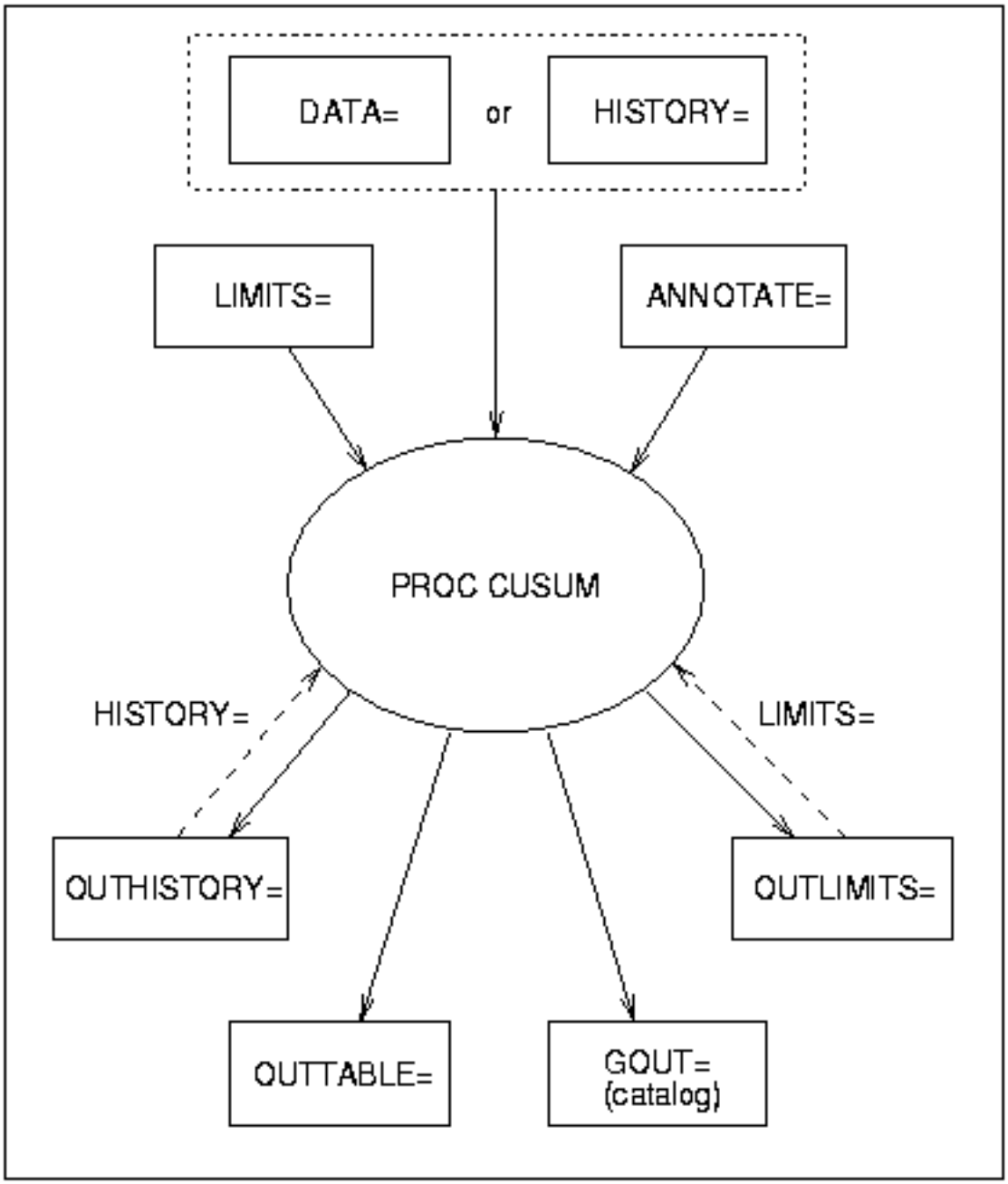
- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the CUSUM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

Input and Output Data Sets: CUSUM Procedure

Figure 7.1 summarizes the data sets used with the CUSUM procedure.

Figure 7.1 Input and Output Data Sets in the CUSUM Procedure



XCHART Statement: CUSUM Procedure

Overview: XCHART Statement

The XCHART statement creates cumulative sum control charts from subgroup means or individual measurements. You can create these charts for one-sided cusum (decision interval) schemes or for two-sided (V-mask) schemes. A one-sided scheme is designed to detect either a positive or a negative shift from the target mean, and a two-sided scheme is designed to detect positive and negative shifts from the target mean.

You can use options in the XCHART statement to

- specify parameters for a decision interval or V-mask
- specify the shift δ to be detected
- specify the target mean μ_0
- specify a known (standard) value σ_0 for the process standard deviation or estimate the standard deviation from the data using various methods
- tabulate the information displayed on the chart
- save the information displayed on the chart in an output data set
- read parameters for the cusum scheme from a data set
- display a secondary chart that plots a time trend that has been removed from the data
- add block legends and special symbol markers to reveal stratification in process data
- superimpose stars at each point to represent related multivariate factors
- display vertical and horizontal reference lines
- modify the axis values and labels
- modify the chart layout and appearance

You have three alternatives for producing cumulative sum control charts with the XCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: XCHART Statement

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the section "Syntax: XCHART Statement" on page 569, and advanced examples are given in the section "Examples: XCHART Statement" on page 602.

Creating a V-Mask Cusum Chart from Raw Data

NOTE: See *Two-sided Cusum Chart with V-Mask* in the SAS/QC Sample Library.

A machine fills eight-ounce cans of two-cycle engine oil additive. The filling process is believed to be in statistical control, and the process is set so that the average weight of a filled can is $\mu_0 = 8.100$ ounces. Previous analysis shows that the standard deviation of fill weights is $\sigma_0 = 0.050$ ounces. A two-sided cusum chart is used to detect shifts of at least one standard deviation in either the positive or negative direction from the target mean of 8.100 ounces.

Subgroup samples of four cans are selected every hour for twelve hours. The cans are weighed, and their weights are saved in a SAS data set named Oil.

```
data Oil;
   label Hour = 'Hour';
   input Hour @;
   do i=1 to 4;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
 1  8.024  8.135  8.151  8.065
 2  7.971  8.165  8.077  8.157
 3  8.125  8.031  8.198  8.050
 4  8.123  8.107  8.154  8.095
 5  8.068  8.093  8.116  8.128
 6  8.177  8.011  8.102  8.030
 7  8.129  8.060  8.125  8.144
 8  8.072  8.010  8.097  8.153
 9  8.066  8.067  8.055  8.059
10  8.089  8.064  8.170  8.086
11  8.058  8.098  8.114  8.156
12  8.147  8.116  8.116  8.018
;
```

The data set Oil is partially listed in Figure 7.2.

Figure 7.2 Partial Listing of the Data Set Oil

Hour	Weight
1	8.024
1	8.135
1	8.151
1	8.065
2	7.971
2	8.165
2	8.077
2	8.157
3	8.125
3	8.031
3	8.198
3	8.050
4	8.123
4	8.107

Each observation contains one value of Weight along with its associated value of Hour, and the values of Hour are in increasing order. The CUSUM procedure assumes that DATA= input data sets are sorted in this "strung-out" form.

The following statements request a two-sided cusum chart with a V-mask for the average weights:

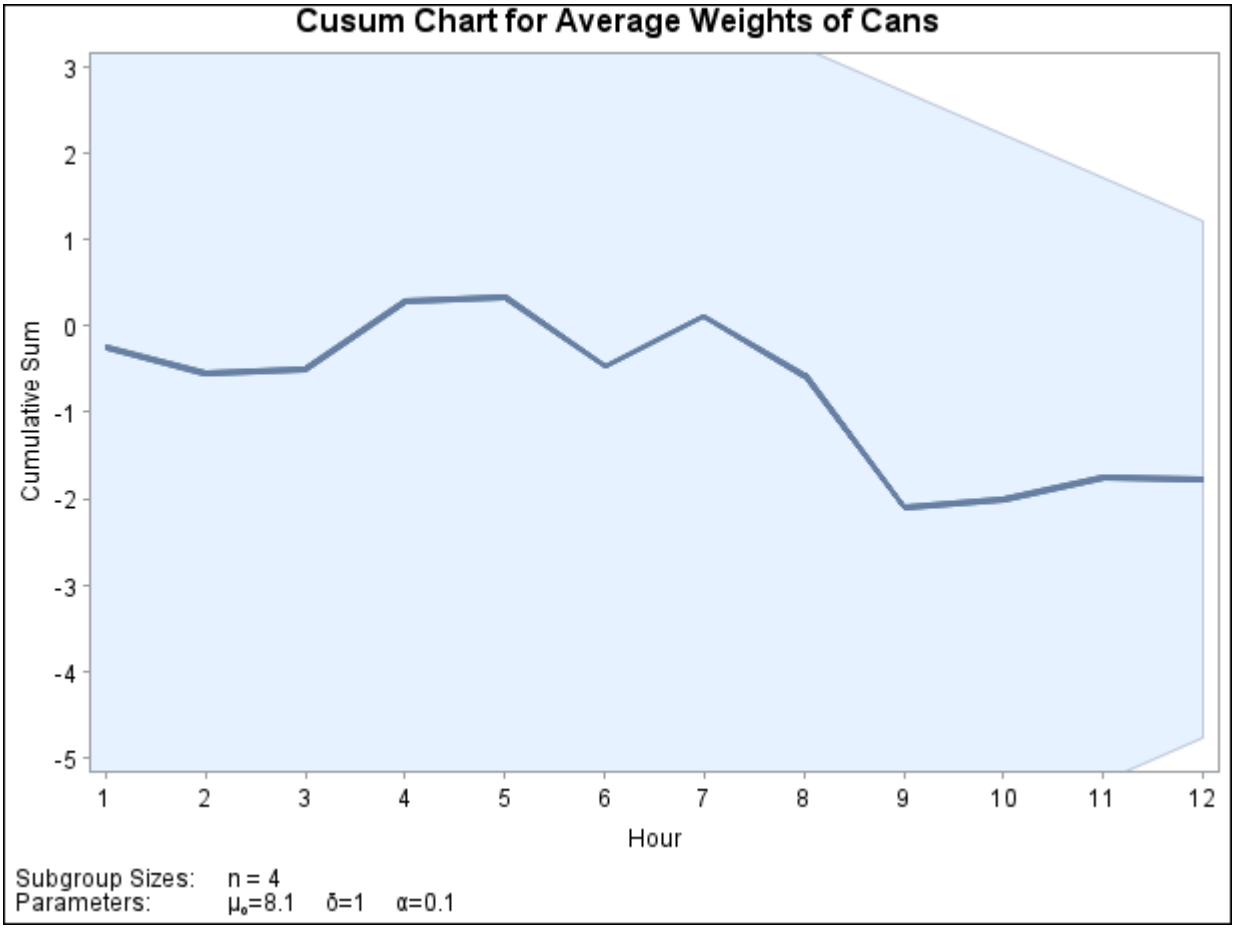
```
ods graphics off;
title 'Cusum Chart for Average Weights of Cans';
proc cusum data=Oil;
   xchart Weight*Hour /
      mu0    = 8.100          /* Target mean for process  */
      sigma0 = 0.050          /* Known standard deviation */
      delta  = 1              /* Shift to be detected     */
      alpha  = 0.10           /* Type I error probability */
      vaxis  = -5 to 3 ;
   label Weight = 'Cumulative Sum';
run;
```

The CUSUM procedure is invoked with the PROC CUSUM statement. The DATA= option in the PROC CUSUM statement specifies that the SAS data set Oil is to be read. The variables to be analyzed are specified in the XCHART statement. The process measurement variable (Weight) is specified before the asterisk (this variable is referred to more generally as a *process*). The time variable (Hour) is specified after the asterisk (this variable is referred to more generally as a *subgroup-variable* because it determines how the measurements are classified into rational subgroups).

The option ALPHA=0.10 specifies the probability of a Type 1 error for the cusum scheme (the probability of detecting a shift when none occurs).

The cusum chart is shown in Figure 7.3.

Figure 7.3 Two-Sided Cusum Chart with V-Mask



The cusum S_1 plotted at Hour=1 is simply the standardized deviation of the first subgroup mean from the target mean.

$$S_1 = \frac{8.09375 - 8.100}{0.050/\sqrt{4}} = -0.250$$

The cusum S_2 plotted at Hour=2 is S_1 plus the standardized deviation of the second subgroup mean from the target mean.

$$S_2 = S_1 + \frac{8.0925 - 8.100}{0.050/\sqrt{4}} = -0.550$$

In general, the cusum plotted at Hour=t is S_{t-1} plus the standardized deviation of the tth subgroup mean from the target mean.

$$S_t = S_{t-1} + \frac{\bar{X}_t - \mu_0}{\sigma_0/\sqrt{n}}$$

For further details, see "Two-Sided Cusum Schemes" on page 587.

You can interpret the chart by comparing the points with the V-mask whose right edge is centered at the most recent point (Hour=12). Since none of the points cross the arms of the V-mask, there is no evidence

that a shift has occurred, and the fluctuations in the cusums can be attributed to chance variation. In general, crossing the lower arm is evidence of an increase in the process mean, whereas crossing the upper arm is evidence of a decrease in the mean.

Creating a V-Mask Cusum Chart from Subgroup Summary Data

NOTE: See *Two-sided Cusum Chart with V-Mask* in the SAS/QC Sample Library.

The previous example illustrates how you can create a cusum chart using raw process measurements read from a DATA= data set. In many applications, however, the data are provided in *summarized form* as subgroup means. This example illustrates the use of the XCHART statement when the input data set is a HISTORY= data set.

The following data set provides the subgroup means, standard deviations, and sample sizes corresponding to the variable Weight in the data set Oil (see the section "Creating a V-Mask Cusum Chart from Raw Data" on page 555:

```
data Oilstat;
   label Hour = 'Hour';
   input Hour WeightX WeightS WeightN;
   datalines;
 1  8.0938  0.0596  4
 2  8.0925  0.0902  4
 3  8.1010  0.0763  4
 4  8.1198  0.0256  4
 5  8.1013  0.0265  4
 6  8.0800  0.0756  4
 7  8.1145  0.0372  4
 8  8.0830  0.0593  4
 9  8.0618  0.0057  4
10  8.1023  0.0465  4
11  8.1065  0.0405  4
12  8.0993  0.0561  4
;
```

The data set Oilstat is listed in Figure 7.4.

Figure 7.4 Listing of the Data Set Oilstat

Obs	Hour	WeightX	WeightS	WeightN
1	1	8.0938	0.0596	4
2	2	8.0925	0.0902	4
3	3	8.1010	0.0763	4
4	4	8.1198	0.0256	4
5	5	8.1013	0.0265	4
6	6	8.0800	0.0756	4
7	7	8.1145	0.0372	4
8	8	8.0830	0.0593	4
9	9	8.0618	0.0057	4
10	10	8.1023	0.0465	4
11	11	8.1065	0.0405	4
12	12	8.0993	0.0561	4

Since the data set contains a subgroup variable, a mean variable, a standard deviation variable, and a sample size variable, it can be read as a HISTORY= data set. Note that the names WeightX, WeightS, and WeightN satisfy the naming conventions for summary variables since they begin with a common prefix (Weight) and end with the suffix letters *X*, *S*, and *N*.

The following statements create the cusum chart:

```
title 'Cusum Chart for Average Weights of Cans';
proc cusum history=Oilstat;
   xchart Weight*Hour /
      mu0    = 8.100          /* target mean              */
      sigma0 = 0.050          /* known standard deviation */
      delta  = 1              /* shift to be detected     */
      alpha  = 0.10           /* Type 1 error probability */
      vaxis  = -5 to 3 ;
   label WeightX = 'Cumulative Sum';
run;
```

Note that the *process* Weight specified in the XCHART statement is the prefix of the summary variable names in Oilstat. Also note that the vertical axis label is specified by associating a variable label with the subgroup mean variable (WeightX). The chart (not shown here) is identical to the one in Figure 7.2.

In general, a HISTORY= input data set used with the XRCHART statement must contain the following four variables:

- subgroup variable
- subgroup mean variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of subgroup mean, standard deviation, and sample size variables must begin with the prefix *process* specified in the XRCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively.

Note that the interpretation of *process* depends on the input data set specified in the PROC CUSUM statement.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names containing the summary statistics.

For more information, see “DATA= Data Set” on page 600 and “HISTORY= Data Set” on page 601.

Saving Summary Statistics

NOTE: See *Two-sided Cusum Chart with V-Mask* in the SAS/QC Sample Library.

In this example, the CUSUM procedure is used to save summary statistics and cusums in an output data set. The summary statistics can subsequently be analyzed by the CUSUM procedure (as in the preceding example). The following statements read the raw measurements from the data set Oil (see "Creating a V-Mask Cusum Chart from Raw Data" on page 555) and create a summary data set named Oilhist:

```
title 'Cusum Chart for Average Weights of Cans';
proc cusum data=Oil;
   xchart Weight*Hour /
      nochart
      outhistory = Oilhist
      mu0        = 8.100        /* Target mean for process  */
      sigma0     = 0.050        /* Known standard deviation */
      delta      = 1            /* Shift to be detected     */
      alpha      = 0.10         /* Type I error probability */
      vaxis      = -5 to 3 ;
   label Weight = 'Cumulative Sum';
run;
```

The OUTHISTORY= option names the SAS data set containing the summary information, and the NOCHART option suppresses the display of the charts (since the purpose here is simply to create an output data set). Figure 7.5 lists the data set Oilhist.

Figure 7.5 Listing of the Data Set Oilhist

Cusum Chart for Average Weights of Cans

Obs	Hour	WeightX	WeightS	WeightC	WeightN
1	1	8.0938	0.0596	-.2500	4
2	2	8.0925	0.0902	-.5500	4
3	3	8.1010	0.0763	-.5100	4
4	4	8.1198	0.0256	0.2800	4
5	5	8.1013	0.0265	0.3300	4
6	6	8.0800	0.0756	-.4700	4
7	7	8.1145	0.0372	0.1100	4
8	8	8.0830	0.0593	-.5700	4
9	9	8.0618	0.0057	-2.100	4
10	10	8.1023	0.0465	-2.010	4
11	11	8.1065	0.0405	-1.750	4
12	12	8.0993	0.0561	-1.780	4

There are five variables in the data set.

- Hour contains the subgroup index
- WeightX contains the subgroup means
- WeightS contains the subgroup standard deviations
- WeightC contains the cumulative sums
- WeightN contains the subgroup sample sizes

Note that the variables in the OUTHISTORY= data set are named by adding the suffix characters *X*, *S*, *N*, and *C* to the *process* Weight specified in the XCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 597.

Creating a One-Sided Cusum Chart with a Decision Interval

NOTE: See *One-sided Cusum Chart* in the SAS/QC Sample Library.

An alternative to the V-mask cusum chart is the one-sided cusum chart with a decision interval, which is sometimes referred to as the "computational form of the cusum chart." This example illustrates how you can create a one-sided cusum chart for individual measurements.

A can of oil is selected every hour for fifteen hours. The cans are weighed, and their weights are saved in a SAS data set named Cans:[1]

```
data Cans;
   length comment $16;
   label Hour = 'Hour';
   input Hour Weight comment $16. ;
   datalines;
 1  8.024
 2  7.971
 3  8.125
 4  8.123
 5  8.068
 6  8.177  Pump Adjusted
 7  8.229  Pump Adjusted
 8  8.072
 9  8.066
10  8.089
11  8.058
12  8.147
13  8.141
14  8.047
15  8.125
;
```

[1] This data set is used by later examples in this chapter.

Suppose the problem is to detect a *positive* shift in the process mean of one standard deviation ($\delta = 1$) from the target of 8.100 ounces. Furthermore, suppose that

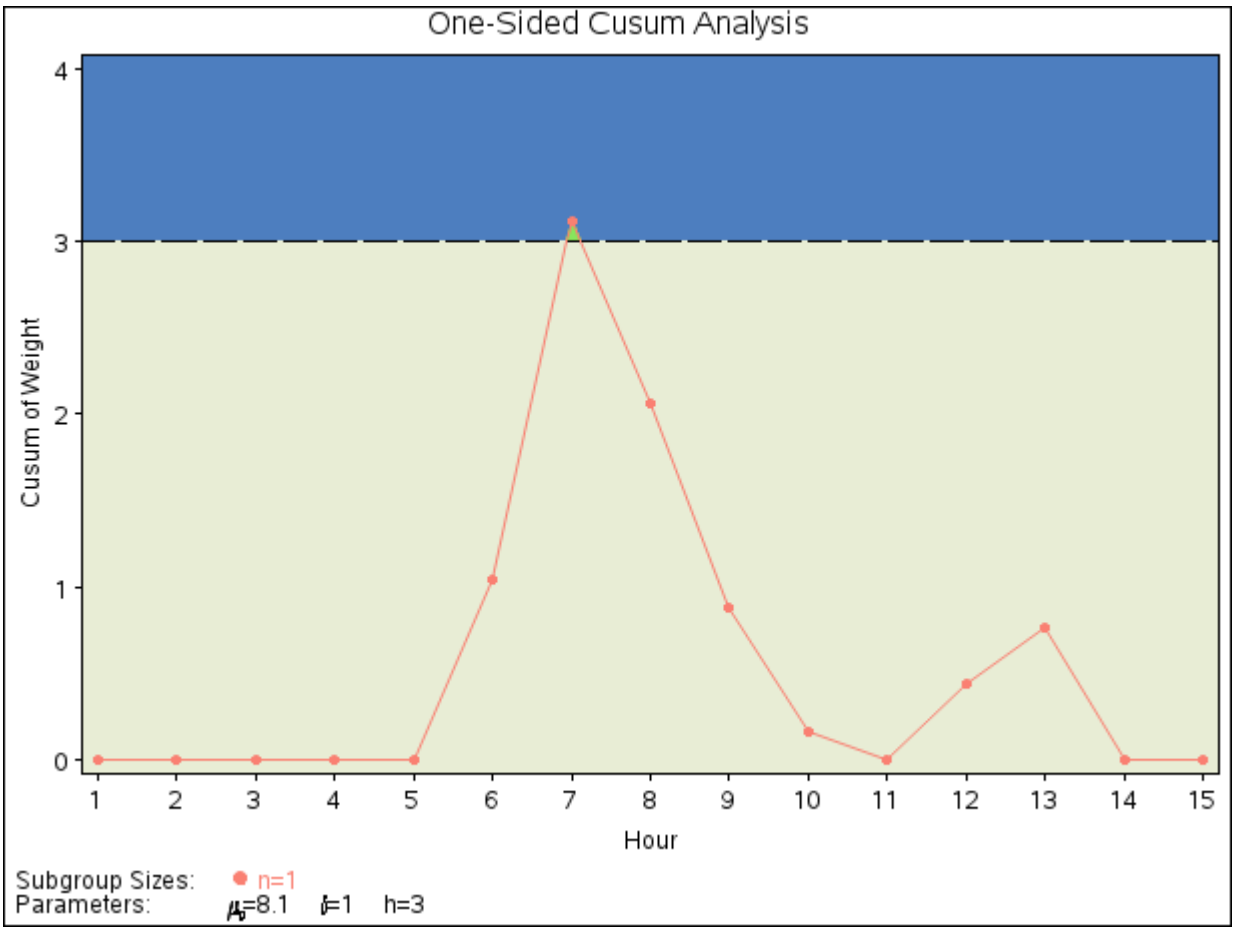
- a known value $\sigma_0 = 0.050$ is available for the process standard deviation
- an in-control average run length (ARL) of approximately 100 is required
- an ARL of approximately five is appropriate for detecting the shift

Table 7.5 indicates that these ARLs can be achieved with the decision interval $h = 3$ and the reference value $k = 0.5$. The following statements use these parameters to create the chart and tabulate the cusum scheme:

```
options nogstyle;
goptions ftext='albany amt';
symbol v=dot color=salmon h=1.8 pct;
title "One-Sided Cusum Analysis";
proc cusum data=Cans;
   xchart Weight*Hour /
      mu0       = 8.100     /* target mean for process     */
      sigma0    = 0.050     /* known standard deviation    */
      delta     = 1         /* shift to be detected        */
      h         = 3         /* cusum parameter h           */
      k         = 0.5       /* cusum parameter k           */
      scheme    = onesided  /* one-sided decision interval */
      tableall              /* table                       */
      cinfill   = ywh
      cframe    = bigb
      cout      = salmon
      cconnect  = salmon
      climits   = black
      coutfill  = bilg;
   label Weight = 'Cusum of Weight';
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement, GOPTIONS, and XCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The chart is shown in Figure 7.6.

Figure 7.6 One-Sided Cusum Chart with Decision Interval



The cusum plotted at Hour=t is

$$S_t = \max(0, S_{t-1} + (z_t - k))$$

where $S_0 = 0$, and z_t is the standardized deviation of the tth measurement from the target.

$$z_t = \frac{x_t - \mu_0}{\sigma_0}$$

The cusum S_t is referred to as an *upper cumulative sum*. A shift is signaled at the seventh hour since S_7 exceeds h. For further details, see "One-Sided Cusum Schemes" on page 585.

The option TABLEALL requests the tables shown in Figure 7.7, Figure 7.8, and Figure 7.9. The table in Figure 7.7 summarizes the cusum scheme, and it confirms that an in-control ARL of 117.6 and an ARL of 6.4 at $\delta = 1$ are achieved with the specified h and k.

Figure 7.7 Summary Table

Cusum Parameters	
Process Variable	Weight (Cusum of Weight)
Subgroup Variable	Hour (Hour)
Scheme	One-Sided
Target Mean (Mu0)	8.1
Sigma0	0.05
Delta	1
Nominal Sample Size	1
h	3
k	0.5
Average Run Length (Delta)	6.40390895
Average Run Length (0)	117.595692

The table in Figure 7.8 tabulates the information displayed in Figure 7.6.

Figure 7.8 Tabulation of One-Sided Chart

One-Sided Cusum Analysis

The CUSUM Procedure

Cumulative Sum Chart Summary for Weight					
Hour	Subgroup Sample Size	Individual Value	Cusum	Decision Interval	Decision Interval Exceeded
1	1	8.0240000	0.0000000	3.0000	
2	1	7.9710000	0.0000000	3.0000	
3	1	8.1250000	0.0000000	3.0000	
4	1	8.1230000	0.0000000	3.0000	
5	1	8.0680000	0.0000000	3.0000	
6	1	8.1770000	1.0400000	3.0000	
7	1	8.2290000	3.1200000	3.0000	Upper
8	1	8.0720000	2.0600000	3.0000	
9	1	8.0660000	0.8800000	3.0000	
10	1	8.0890000	0.1600000	3.0000	
11	1	8.0580000	0.0000000	3.0000	
12	1	8.1470000	0.4400000	3.0000	
13	1	8.1410000	0.7600000	3.0000	
14	1	8.0470000	0.0000000	3.0000	
15	1	8.1250000	0.0000000	3.0000	

The table in Figure 7.9 presents the computational form of the cusum scheme described by Lucas (1976).

Figure 7.9 Computational Form of Cusum Scheme

One-Sided Cusum Analysis

The CUSUM Procedure

Computational Cumulative Sum for Weight				
Hour	Subgroup Sample Size	Individual Value	Upper Cusum	Number of Consecutive Upper Sums > 0
1	1	8.0240000	0.0000000	0
2	1	7.9710000	0.0000000	0
3	1	8.1250000	0.0000000	0
4	1	8.1230000	0.0000000	0
5	1	8.0680000	0.0000000	0
6	1	8.1770000	1.0400000	1
7	1	8.2290000	3.1200000	2
8	1	8.0720000	2.0600000	3
9	1	8.0660000	0.8800000	4
10	1	8.0890000	0.1600000	5
11	1	8.0580000	0.0000000	0
12	1	8.1470000	0.4400000	1
13	1	8.1410000	0.7600000	2
14	1	8.0470000	0.0000000	0
15	1	8.1250000	0.0000000	0

Following the method of Lucas (1976), the process average at the out-of-control point (Hour=7) can be estimated as

$$\mu_0 + \sigma_0 \frac{(N_7 k + S_7)}{(N_7 \sqrt{n})} = 8.10 + 0.05(2(0.5) + 3.12)/2$$
$$= 8.203 \text{ ounces}$$

where $S_7 = 3.12$ is the upper sum at Hour=7, and $N_7 = 2$ is the number of successive positive upper sums at Hour=7.

Saving Cusum Scheme Parameters

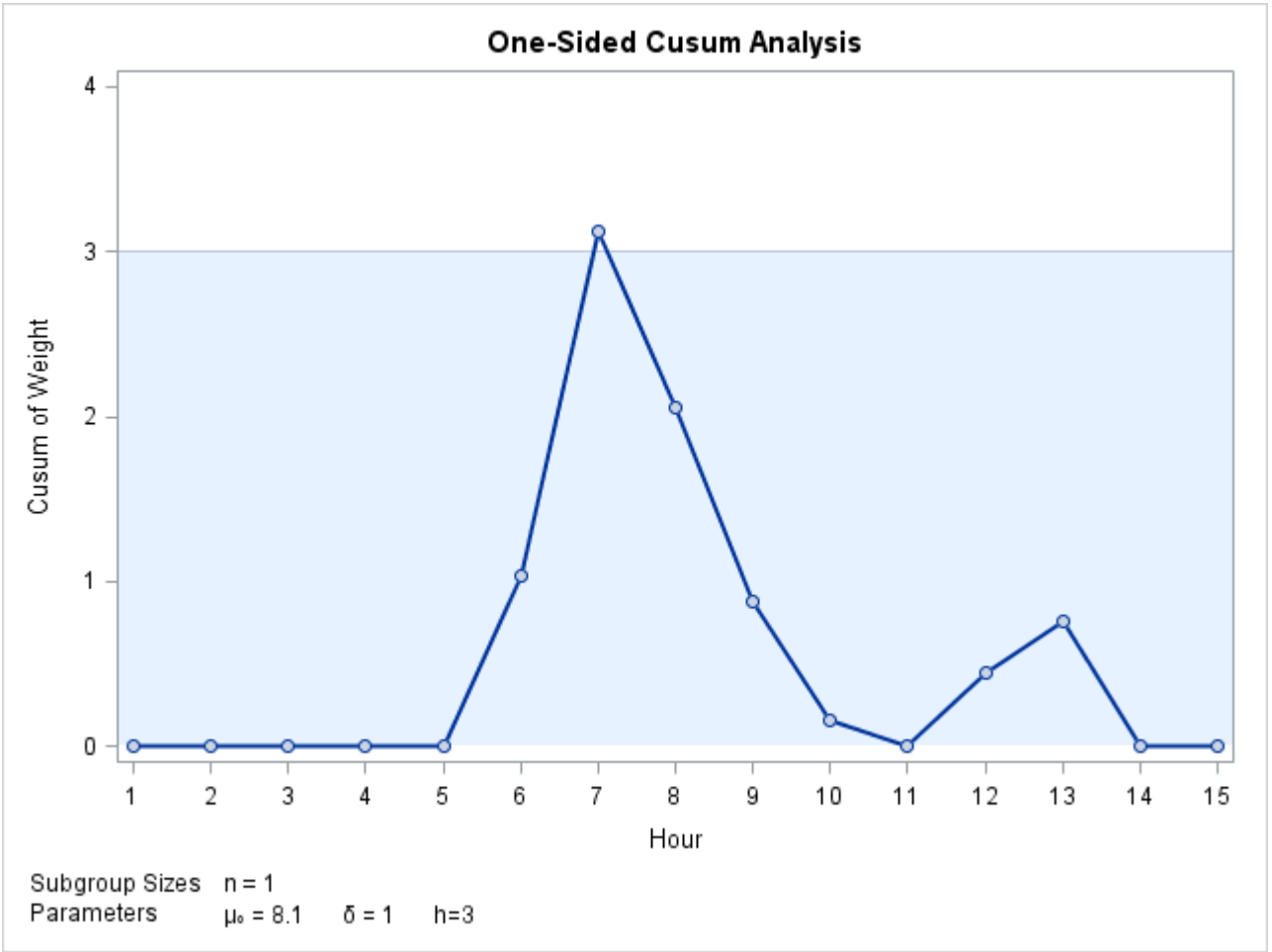
NOTE: See *One-sided Cusum Chart* in the SAS/QC Sample Library.

This example is a continuation of the previous example that illustrates how to save cusum scheme parameters in a data set specified with the OUTLIMITS= option. This enables you to apply the parameters to future data or to subsequently modify the parameters with a DATA step program.

```
ods graphics on;
title 'One-Sided Cusum Analysis';
proc cusum data=Cans;
   xchart Weight*Hour /
      mu0       = 8.100     /* target mean for process     */
      sigma0    = 0.050     /* known standard deviation    */
      delta     = 1         /* shift to be detected        */
      h         = 3         /* cusum parameter h           */
      k         = 0.5       /* cusum parameter k           */
      scheme    = onesided  /* one-sided decision interval */
      outlimits = cusparm
      odstitle  = title
      markers;
   label Weight = 'Cusum of Weight';
run;
```

The chart, shown in Figure 7.10, is similar to the one in Figure 7.6 but is created by using ODS Graphics because the ODS GRAPHICS ON statement is specified before the PROC CUSUM statement.

Figure 7.10 One-Sided Cusum Scheme with Decision Interval



The OUTLIMITS= data set is listed in Figure 7.11.

Figure 7.11 Listing of the OUTLIMITS= Data Set cusparm

One-Sided Cusum Analysis

Obs	_VAR_	_SUBGRP_	_TYPE_	_LIMITN_	_H_	_K_	_SCHEME_	_MU0_	_DELTA_
1	Weight	Hour	STANDARD	1	3	0.5	ONESIDED	8.1	1

Obs	_MEAN_	_STDDEV_	_ARLIN_	_ARLOUT_
1	8.09747	0.05	117.596	6.40391

The data set contains one observation with the parameters for *process* Weight. The variables _TYPE_, _H_, _K_, _MU0_, _DELTA_, and _STDDEV_ save the parameters specified with the options SCHEME=, H=, K=, MU0=, DELTA=, and SIGMA0=, respectively. The variable _MEAN_ saves an estimate of the process mean, and the variable _LIMITN_ saves the nominal sample size. The variables _ARLIN_ and _ARLOUT_ save the average run lengths for $\delta = 0$ and for $\delta = 1$.

The variables _VAR_ and _SUBGRP_ save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _STDDEV_ is an estimate or a standard value.

For more information, see "OUTLIMITS= Data Set" on page 596.

Reading Cusum Scheme Parameters

NOTE: See *One-sided Cusum Chart* in the SAS/QC Sample Library.

This example shows how the cusum parameters saved in the previous example can be applied to new measurements saved in a data set named Cans2:

```
data Cans2;
   length pump $ 8;
   label Hour = 'Hour';
   input Hour Weight pump $ 8. ;
   datalines;
16  8.1765  Pump 3
17  8.0949  Pump 3
18  8.1393  Pump 3
19  8.1491  Pump 3
20  8.0473  Pump 1
21  8.1602  Pump 1
22  8.0633  Pump 1
23  8.0921  Pump 1
24  8.1573  Pump 1
25  8.1304  Pump 1
26  8.0979  Pump 1
27  8.2407  Pump 1
28  8.0730  Pump 1
29  8.0986  Pump 2
30  8.0785  Pump 2
31  8.2308  Pump 2
32  8.0986  Pump 2
33  8.0782  Pump 2
34  8.1435  Pump 2
35  8.0666  Pump 2
;
```

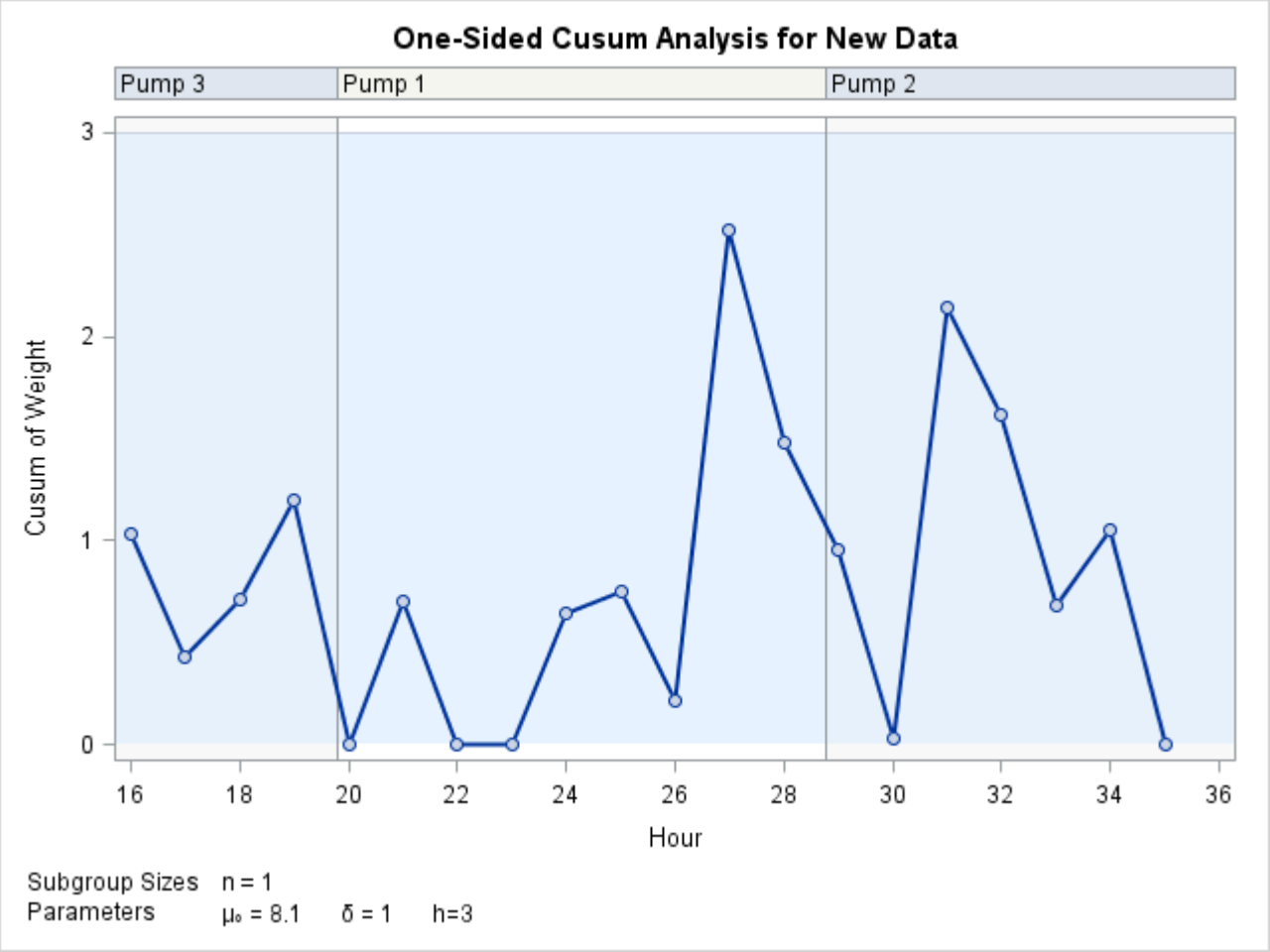
The following statements create a one-sided cusum chart for the measurements in Cans2 using the parameters in cusparm:

```
title "One-Sided Cusum Analysis for New Data";
proc cusum data=Cans2 limits=cusparm;
   xchart Weight*Hour( pump ) / odstitle=title markers;
   label Weight = 'Cusum of Weight';
run;
```

ODS Graphics remains enabled until it is disabled with the ODS GRAPHICS OFF statement, so this cusum chart is also created using ODS Graphics.

The LIMITS= option in the PROC CUSUM statement specifies the data set containing preestablished cusum parameters. The chart, shown in Figure 7.12, indicates that the process is in control. Levels of the variable pump (referred to as a *block-variable*) do not enter into the analysis but are displayed in a block legend across the top of the chart. See Block Variable Legend Options.

Figure 7.12 Cusum Chart with Decision Interval for New Data



In general, the parameters for a specified *process* and *subgroup-variable* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* (in this case, Weight)
- the value of _SUBGRP_ matches the *subgroup-variable* name (in this case, Hour)

If you are maintaining more than one set of cusum parameters for a particular *process*, you will find it convenient to include a special identifier variable named _INDEX_ in the LIMITS= data set. This must be a character variable of length 16. Then, if you specify READINDEX='*value*' in the XCHART statement, the parameters for a specified *process* and *subgroup-variable* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable* name
- the value of _INDEX_ matches *value*

In this example, the LIMITS= data set was created in a previous run of the CUSUM procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 600 for details concerning the variables that you must provide.

Syntax: XCHART Statement

The basic syntax for a *one-sided (decision interval) scheme* using the XCHART statement is as follows:

XCHART *process* * *subgroup-variable* / **SCHEME=ONESIDED MU0=***target* **DELTA=***shift* **H=***h* < *options* > **;**

The general form of this syntax is as follows:

XCHART *processes* * *subgroup-variable* < *(block-variables)* >
< **=***symbol-variable* **='***character***'** > / **SCHEME=ONESIDED MU0=***target* **DELTA=***shift* **H=***h* < *options* > **;**

Note that the options SCHEME=ONESIDED, MU0=, DELTA=, and H= are required unless their values are read from a LIMITS= data set.

The basic syntax for a *two-sided (V-mask) scheme* is as follows:

XCHART *process* * *subgroup-variable* / **MU0=***target* **DELTA=***shift* **ALPHA=***alpha* **H=***h* < *options* > **;**

The general form of this syntax is as follows:

XCHART *processes* * *subgroup-variable* < *(block-variables)* >
< **=***symbol-variable* | **=** **'***character***'** > / **MU0=***target* **DELTA=***shift* **ALPHA=***alpha* **H=***h* < *options* > **;**

Note that the options MU0=, DELTA=, and either ALPHA= or H= are required unless their values are read from a LIMITS= data set.

You can use any number of XCHART statements in the CUSUM procedure. The components of the XCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC CUSUM statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating a V-Mask Cusum Chart from Raw Data" on page 555.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating a V-Mask Cusum Chart from Subgroup Summary Data" on page 558.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. The parameters specified in the XCHART statement are applied to all of the *processes*.[2]

subgroup-variable

is the variable that classifies the data into subgroups. The *subgroup-variable* is required. In the examples "Creating a V-Mask Cusum Chart from Raw Data" on page 555 and "Creating a V-Mask Cusum Chart from Subgroup Summary Data" on page 558, Hour is the subgroup variable.

block-variables

are optionally specified variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See Figure 7.12 for an example.

symbol-variable

is an optionally specified variable whose levels (unique values) determine the plotting character or symbol marker used to plot the cusums.

- If you produce a line printer chart, an 'A' marks points corresponding to the first level of the *symbol-variable*, a 'B' marks points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements.

character

specifies a plotting character for line printer charts. See Figure 7.10 for an example.

options

specify optional cusum parameters, enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function.

Summary of Options

The following tables list the XCHART statement options by function. Options unique to the CUSUM procedure are listed in Table 7.2, and are described in detail in the section "Dictionary of Special Options" on page 579. Options that are common to both the CUSUM and SHEWHART procedures are listed in Table 7.3. They are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

[2] For this reason, it may be preferable to read distinct cusum parameters for each *process* from a LIMITS= data set.

Table 7.2 XCHART Statement Special Options

Option	Description
Options for Specifying a One-Sided (Decision Interval) Cusum Scheme	
DELTA=	Specifies shift to be detected as a multiple of standard error
H=	Specifies decision interval h ($h > 0$) as a multiple of standard error
HEADSTART=	Specifies headstart value S_0 as a multiple of standard error
K=	Specifies reference value k ($k > 0$)
LIMITN=	Specifies fixed nominal sample size for cusum scheme
LIMITN=	Specifies that cusums are to be computed for all subgroups regardless of sample size
MU0=	Specifies target μ_0 for mean
NOREADLIMITS	Specifies that cusum parameters are not to be read from LIMITS= data set (SAS 6.10 and later releases)
READINDEX=	Reads cusum scheme parameters from a LIMITS= data set
READLIMITS	Specifies that cusum parameters are to be read from LIMITS= data set (SAS 6.09 and earlier releases)
SCHEME=ONESIDED	Specifies a one-sided scheme
SHIFT=	Specifies shift to be detected in data units
SIGMA0=	Specifies standard (known) value σ_0 for process standard deviation
Options for Specifying a Two-Sided (V-Mask) Cusum Scheme	
ALPHA=	Specifies probability of Type 1 error
BETA=	Specifies probability of Type 2 error
H=	Specifies vertical distance between V-mask origin and upper (or lower) arm
K=	Specifies slope of lower arm of V-mask
LIMITN=	Specifies fixed nominal sample size for cusum scheme
LIMITN=	Specifies that cusums are to be computed for all subgroups regardless of sample size
NOREADLIMITS	Specifies that cusum parameters are not to be read from LIMITS= data set (SAS 6.10 and later releases)
READINDEX=	Reads cusum scheme parameters from a LIMITS= data set
READLIMITS	Specifies that cusum parameters are to be read from LIMITS= data set (SAS 6.09 and earlier releases)
READSIGMAS	Reads _SIGMAS_ instead of _ALPHA_ from LIMITS= data set when both variables are available
SIGMAS=	Specifies probability of Type 1 error as probability that standard normally distributed variable exceeds *value* in absolute value

Table 7.2 *continued*

Option	Description
Options for Estimating Process Standard Deviation	
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies whether _STDDEV_ in OUTLIMITS= data set is an estimate or standard, and specifies value of _TYPE_ in OUTLIMITS= data set
Options for Displaying Decision Interval or V-Mask	
CINFILL=	Specifies color for area under decision interval line or inside V-mask
CLIMITS=	Specifies color of decision interval line
CMASK=	Specifies color of V-mask outline
LLIMITS=	Specifies line type for decision interval
LMASK=	Specifies line type for V-mask arms
NOMASK	Suppresses display of V-mask
ORIGIN=	Specifies value of *subgroup-variable* locating origin of V-mask
WLIMITS=	Specifies line width for decision interval
WMASK=	Specifies line width for V-mask
Tabulation Options	
TABLEALL	Specifies the options TABLECHART, TABLECOMP, TABLEID, TABLEOUT, and TABLESUMMARY
TABLECHART	Tabulates the information displayed in the cusum chart
TABLECOMP	Tabulates the computational form of the cusum scheme as described by Lucas (1976) and Lucas and Crosier (1982)
TABLEID	Augments TABLECHART and TABLECOMP tables with columns for ID variables
TABLEOUT	Augments TABLECHART table with a column indicating if the decision interval or V-mask was exceeded
TABLESUMMARY	Tabulates the parameters for the cusum scheme and the average run lengths corresponding to shifts of zero and δ

Table 7.3 XCHART Statement General Options

Option	Description
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on cusum chart
ALLLABEL2=	Labels every point on trend chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
COUT=	Specifies color for portions of line segments that connect points outside control limits

Table 7.3 *continued*

Option	Description
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels
LABELHEIGHT=	Specifies height of labels
NOCONNECT	Suppresses line segments that connect points on chart
NOTRENDCONNECT	Suppresses line segments that connect points on trend chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of cusum chart

Table 7.3 *continued*

Option	Description
VAXIS2=	Specifies major tick mark values for vertical axis of trend chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	Specifies the natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	Specifies the maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
TRENDVAR=	Specifies list of trend variables
YPCT1=	Specifies length of vertical axis on cusum chart as a percentage of sum of lengths of vertical axes for cusum and trend charts
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options

Table 7.3 *continued*

Option	Description
HREF=	Specifies position of reference lines perpendicular to horizontal axis on cusum chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on cusum chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on cusum chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	Specifies the position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to cusum chart
ANNOTATE2=	Specifies annotate data set that adds features to trend chart
DESCRIPTION=	Specifies description of cusum chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts

Table 7.3 *continued*

Option	Description
NAME=	Specifies name of cusum chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
WTREND=	Specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	Draws portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote

Table 7.3 *continued*

Option	Description
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend

Table 7.3 *continued*

Option	Description
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on cusum chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars

Table 7.3 *continued*

Option	Description
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
VREFCHAR=	Specifies line character for VREF= and VREF2= lines

Dictionary of Special Options

General Options

You can specify the following *options* when you use either ODS Graphics or traditional graphics:

ALPHA=*value*

specifies the probability α of incorrectly deciding that a shift has occurred when the process mean is equal to the target mean. This is known as the probability of a Type 1 error. The *value* must be between zero and one, and it is typically set at 0.05 or 0.10. If you specify the ALPHA= option, the error probability approach is used to determine the V-mask. For details, see "Defining the V-Mask for a Two-Sided Cusum Scheme" on page 588.

The ALPHA= option is applicable only with two-sided cusum schemes. As an alternative to the ALPHA= *value*, you can specify the percentile $z_{1-\alpha/2}$ from a standard normal distribution with the SIGMAS= option. As a second alternative, you can specify the geometric parameter h for the V-mask (in standard error units) with the H= option.

In addition to the ALPHA= option, you can optionally specify the probability of a Type 2 error with the BETA= option.

BETA=*value*

specifies the probability β of failing to discover that the specified shift has occurred. This is known as the probability of a Type 2 error. The *value* must be between zero and one. The BETA= option is used in conjunction with either the ALPHA= option or the SIGMAS= option.

The interpretation of β is based on the analogy between cusum charts and sequential probability ratio tests, and it is inexact since the cusum chart does not provide an acceptance region. Refer to Johnson (1961) and Van Dobben de Bruyn (1968) for further details.

DATAUNITS

computes cumulative sums without standardizing the subgroup means or individual measurements. As a result, the vertical axis of the cusum chart is scaled in the same units as the data.

The DATAUNITS option requires constant subgroup sample sizes. If your data do not have constant subgroup sample sizes, you need to specify a constant nominal sample size n for the V-mask or decision interval with the LIMITN= option or with the variable _LIMITN_ in the LIMITS= data set.

DELTA=*value*

specifies the absolute value of the smallest shift to be detected as a multiple δ of the process standard deviation σ or the standard error $\sigma_{\bar{X}}$, depending on whether δ is viewed as a shift in the population mean or a shift in the sampling distribution of the subgroup mean $\bar{X}$, respectively.

If you specify SCHEME=ONESIDED (see the SCHEME= option later in this list) and the *value* is positive, a shift above the process mean is to be detected, whereas if the *value* is negative, a shift below the process mean is to be detected.

As an alternative to specifying the DELTA= option, you can specify the shift in the same units as the data with the SHIFT= option.

H=*value*

specifies the decision interval h for a one-sided cusum scheme. This type of scheme is completely specified by the parameters h and k (see the K= option later in this list). You can also specify the H= option as an alternative to the ALPHA= or SIGMAS= options for a two-sided cusum scheme with a V-mask. In this case, the H= option specifies the vertical distance h between the origin for the V-mask and the upper or lower arm of the V-mask. In either case, the H=*value* must be positive and must be expressed as a multiple of standard error.

You can use a table of average run lengths to choose h (this is typically between zero and 10). See Table 7.5 and Table 7.6

HEADSTART=*value*

HSTART=*value*

specifies a headstart value S_0 for a one-sided cusum scheme. The value must be expressed as a multiple of standard error. See the section "Headstart Values" on page 586,and refer to Lucas and Crosier (1982), Ryan (1989), and Montgomery (1996).

K=*value*

specifies the reference value k for a one-sided (decision interval) cusum scheme. This type of scheme is completely specified by the parameters k and h (see the H= option earlier in this list). You can also specify the K= and H= options as geometric parameters for a two-sided cusum scheme with a V-mask . In this case, the K= option specifies the slope of the lower arm of the V-mask, and the K= and H= options together are alternatives to the error probability options ALPHA=, SIGMAS=, and BETA=. In either case, the K= *value* must be positive and must be expressed as a multiple of standard error.

You can use a table of average run lengths to choose k and h (k is typically between zero and two). See Table 7.5 and Table 7.6.

For a one-sided scheme, the default K= *value* is $\delta/2$, which is referred to as the *central reference value*. For a two-sided scheme where the V-mask is specified geometrically with the H= option, the default K= *value* is $\delta/2$. If, however, the V-mask is specified by an error probability with the ALPHA= option, then the K= option should not be specified.

CAUTION: The interpretation of the K= *value* depends on the *subgroup-variable* and the interval between subgroups that is specified with the INTERVAL= option. For a two-sided scheme, the *value* is the increase in the lower V-mask arm per unit change on the subgroup axis, so the *value* depends on how the *subgroup-variable* is scaled.

- If integer values are assigned to the *subgroup-variable*, then a unit change is defined as one.
- If the *subgroup-variable* has character values, then a unit change is defined as the increment between adjacent values of the *subgroup-variable*.
- If the *subgroup-variable* is numeric and is formatted with a SAS date or time format, then a unit change is defined as the default value for the INTERVAL= option. For example, if a DATE7. format is associated with the *subgroup-variable*, then a unit change is defined as one day.

You can use the INTERVAL= option to modify the definition of a unit change. For example, if a DATE7. format is associated with the *subgroup-variable* but subgroups are collected hourly, then INTERVAL=HOUR defines a unit change as one hour rather than one day.

LIMITN=*n*

LIMITN=VARYING

specifies either a fixed or varying nominal sample size for the control limits. If you specify LIMITN=*n*, cusums are calculated and displayed only for those subgroups with a sample size equal to *n*, although you can specify the ALLN option to force all cusums to be plotted. If you specify LIMITN=VARYING, cusums are calculated and displayed for all subgroups, regardless of sample size.

MU0=*value*

specifies the target mean μ_0 for the process. The target mean must be scaled in the same units as the data.

NOARL

suppresses calculation of average run lengths. By default, this calculation is performed if you specify the TABLESUMMARY option or an OUTLIMITS= data set.

NOMASK

suppresses the display of the V-mask on charts for two-sided schemes. This option does not affect computations of cusums or V-mask parameters.

NOREADLIMITS

specifies that the cusum scheme parameters for each *process* listed in the chart statement are *not* to be read from the LIMITS= data set specified in the PROC CUSUM statement. The NOREADLIMITS option is available only in SAS 6.10 and later releases. See the READLIMITS option later in this list.

ORIGIN=*value*

specifies the origin of the V-mask, which is defined as the horizontal coordinate of the right edge of the V-mask. If a date, time, or datetime format is associated with the *subgroup-variable*, you must specify the *value* as a date, time, or datetime constant, respectively. If the subgroup variable is character, you must specify the *value* as a quoted string. The default *value* is the last (most recent) value of the *subgroup-variable*.

Note that estimates for the process mean and standard deviation are calculated only from subgroups up to and including the origin subgroup.

READINDEX='*value*'
reads cusum scheme parameters from a LIMITS= data set (specified in the PROC CUSUM statement) for each *process* listed in the chart statement. The *i*th set of control limits for a particular *process* is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable*
- the value of _INDEX_ matches *value*

The *value* can be up to 16 characters and must be enclosed in quotes.

READLIMITS
specifies that cusum scheme parameters are to be read from a LIMITS= data set specified in the PROC CUSUM statement. The parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on which release of SAS/QC software you are using.

- **In SAS 6.10 and later releases, the READLIMITS option is not necessary.** To read cusum scheme parameters as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases.
- **In SAS 6.09 and earlier releases, you must specify the READLIMITS option to read cusum scheme parameters as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the cusum scheme parameters are computed from the data.

READSIGMAS
specifies that the variable _SIGMAS_ (instead of _ALPHA_) is to be read from a LIMITS= data set that contains both variables. The variables _SIGMAS_ and _ALPHA_ provide the same parameters as the SIGMAS= and ALPHA= options. By default, _ALPHA_ is read from the LIMITS= data set.

SCHEME=ONESIDED
SCHEME=TWOSIDED
indicates whether the cusum scheme is a one-sided (decision interval) scheme or a two-sided scheme with a V-mask. By default, SCHEME=TWOSIDED.

SHIFT=*value*
specifies the shift to be detected in the same units as the data. The *value* is interpreted as the shift in the mean of the sampling distribution of the subgroup mean. The SHIFT= option is an alternative to the DELTA= option. To specify the SHIFT= option, one of the following must be true:

- The subgroup sample sizes are constant.
- A constant nominal sample size n is provided for the cusum scheme with the LIMITN= option or the _LIMITN_ variable in a LIMITS= data set.

The relationship between the SHIFT= *value* (denoted by Δ) and the DELTA= value (denoted by δ) is $\delta = \Delta/(\sigma/\sqrt{n})$, where σ is the process standard deviation.

SIGMA0=*value*
specifies a known standard deviation σ_0 for the process standard deviation σ. The *value* must be positive. By default, PROC CUSUM estimates σ from the data using the formulas given in "Methods for Estimating the Standard Deviation" on page 594. You can use the variable _STDDEV_ in a LIMITS= data set as an alternative to the SIGMA0= option.

SIGMAS=*value*
specifies the probability α of false detection for a two-sided cusum scheme with a V-mask as the probability that the absolute value of a standard normally distributed variable is greater than the *value*. For example, SIGMAS=3 corresponds to the probability α =0.0027. The *value* must be positive. The SIGMAS= option is an alternative to the ALPHA= and H= options, and only one of these three options can be specified.

The SIGMAS= option is useful for defining cusum charts that correspond to Shewhart charts whose control limits are defined with the same *value* as the multiple of σ. Refer to Johnson and Leone (1962, 1974).

SMETHOD=NOWEIGHT | MVLUE | RMSDF
specifies a method for estimating the process standard deviation from subgroup observations, σ, as summarized by the following table.

Keyword	Method for Estimating Standard Deviation
NOWEIGHT	Estimates σ as an unweighted average of unbiased subgroup estimates of σ
MVLUE	Calculates a minimum variance linear unbiased estimate for σ
RMSDF	Calculates a root-mean square estimate for σ

For formulas, see "Methods for Estimating the Standard Deviation" on page 594.

TABLEALL
requests all the tables specified by the options TABLECHART, TABLECOMP, TABLEID, TABLEOUT, and TABLESUMMARY.

TABLECHART < (EXCEPTIONS) >
creates a table of the subgroup variable, the subgroup sample sizes, the subgroup means, the cumulative sums, and the decision interval or V-mask limits. A table is produced for each *process* specified in the XCHART statement. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the decision interval or V-mask values are exceeded.

TABLECOMP
tabulates the computational form of the cusum scheme as described by Lucas (1976) and Lucas and Crosier (1982). Upper or lower cumulative sums (or both) are tabulated for each *process* given in the XCHART statement. See "Formulas for Cumulative Sums" on page 585 for more information.

TABLEID
augments the tables specified by the TABLECHART and TABLECOMP options with a column for each of the ID variables.

TABLEOUT
augments the table specified by the TABLECHART option with a column indicating whether the decision interval or V-mask values are exceeded.

TABLESUMMARY
produces a table that summarizes the cusum scheme. The table lists the parameters of the scheme and the average run lengths corresponding to shifts of zero and δ. The average run lengths are computed using the method of Goel and Wu (1971). A table is produced for each *process*. You can save the summary in a data set by specifying the OUTLIMITS= option. See "OUTLIMITS= Data Set" on page 596 for details.

TYPE=ESTIMATE

TYPE=STANDARD
specifies the value of _TYPE_ in an OUTLIMITS= data set. The variable _TYPE_ indicates whether the variable _STDDEV_ in the OUTLIMITS= data set represents an estimate or a standard (known) value. The default is 'STANDARD' if the SIGMA0= option is specified; otherwise, the default is 'ESTIMATE'.

Options for Traditional Graphics

You can specify the following *options* when you produce traditional graphics:

CINFILL=*color*
specifies the color for the area under the decision interval or inside the V-mask arms. See also the COUTFILL= option.

CLIMITS=*color*
specifies the color for the decision interval line.

CMASK=*color*
specifies the color for the V-mask arms.

LLIMITS=*linetype*
specifies the line type for the decision interval.

LMASK=*linetype*
specifies the line type for the V-mask arms.

WLIMITS=*linetype*
specifies the width (in pixels) of the decision interval line.

WMASK=*linetype*
specifies the width (in pixels) of the V-mask arms.

Details: XCHART Statement

Basic Notation for Cusum Charts

The following notation is used in this chapter:

μ denotes the mean of the population, also referred to as the *process mean* or the *process level*.

μ_0 denotes the target mean (goal) for the population. Goel and Wu (1971) refer to μ_0 as the "acceptable quality level" and use the symbol μ_a instead. The symbol $\bar{X}_0$ is used for μ_0 in *Glossary and Tables for Statistical Quality Control*. You can provide μ_0 with the MU0= option or with the variable _MU0_ in a LIMITS= data set.

σ denotes the population standard deviation. You can provide σ with the variable _STDDEV_ in a LIMITS= data set (where _TYPE_='STANDARD').

σ_0 denotes a known standard deviation. You can provide σ_0 with the SIGMA0= option or the variable _STDDEV_ in a LIMITS= data set.

$\hat{\sigma}$ denotes an estimate of σ. You can provide $\hat{\sigma}$ with the SIGMA0= option or the variable _STDDEV_ in a LIMITS= data set. To identify this value as an estimate, specify TYPE=ESTIMATE or assign the value 'ESTIMATE' to the variable _TYPE_ in a LIMITS= data set.

n denotes the nominal sample size for the cusum scheme. You can provide n with the LIMITN= option or the variable _LIMITN_ in a LIMITS= data set.

δ denotes the shift in μ to be detected, expressed as a multiple of the standard deviation. You can provide δ with the DELTA= option or the variable _DELTA_ in a LIMITS= data set.

Δ denotes the shift in μ to be detected, expressed in data units. If the sample size n is constant across subgroups, then $\Delta = \delta\sigma_{\bar{X}} = (\delta\sigma)/\sqrt{n}$.
Some authors use the symbol D instead of Δ; for example, refer to Johnson and Leone (1962, 1974) and Wadsworth, Stephens, and Godfrey (1986). You can provide Δ with the SHIFT= option. Although it may be more natural to specify the shift in data units, it is preferable to specify the shift as δ, since this generalizes to data with unequal subgroup sample sizes.

Formulas for Cumulative Sums

One-Sided Cusum Schemes

Positive Shifts If the shift δ to be detected is positive, the cusum computed for the tth subgroup is

$$S_t = \max(0, S_{t-1} + (z_t - k))$$

for t=1, 2, . . . , n, where S_0=0, z_t is defined as for two-sided schemes, and the parameter k, termed the *reference value*, is positive. The cusum S_t is referred to as an *upper cumulative sum*. Since S_t can be written as

$$\max\left(0, S_{t-1} + \frac{\bar{X}_i - (\mu_0 + k\sigma_{\bar{X}_i})}{\sigma_{\bar{X}_i}}\right)$$

the sequence S_t cumulates deviations in the subgroup means greater than k standard errors from μ_0. If S_t exceeds a positive value h (referred to as the *decision interval*), a shift or out-of-control condition is signaled. This formulation follows that of Lucas (1976), Lucas and Crosier (1982), and Montgomery (1996).

Negative Shifts If the shift δ to be detected is negative, the cusum computed for the tth subgroup is

$$S_t = \max(0, S_{t-1} - (z_t + k))$$

for t=1, 2, . . . , n, where S_0=0, z_t is defined as for two-sided cusum schemes, and the parameter k, termed the *reference value*, is positive. The cusum S_t is referred to as a *lower cumulative sum*. Since S_t can be written as

$$\max\left(0, S_{t-1} - \frac{\bar{X}_i - (\mu_0 - k\sigma_{\bar{X}_i})}{\sigma_{\bar{X}_i}}\right)$$

the sequence S_t cumulates the absolute value of deviations in the subgroup means less than k standard errors from μ_0. If S_t exceeds a positive value h (referred to as the *decision interval*), a shift or out-of-control condition is signaled.

This formulation follows that of Lucas (1976), Lucas and Crosier (1982), and Montgomery (1996). Note that S_t is always positive and h is always positive, regardless of whether δ is positive or negative. For schemes designed to detect a negative shift, some authors, including Van Dobben de Bruyn (1968) and Wadsworth, Stephens, and Godfrey (1986), define a reflected version of S_t for which a shift is signaled when S_t is less than a negative limit.

Headstart Values Lucas and Crosier (1982) describe the properties of a fast initial response (FIR) feature for cusum schemes in which the initial cusum S_0 is set to a "headstart" value. Average run length calculations given by Lucas and Crosier (1982) show that the FIR feature has little effect when the process is in control and that it leads to a faster response to an initial out-of-control condition than a standard cusum scheme. You can provide headstart value S_0 with the HEADSTART= option or the variable _HSTART_ in a LIMITS= data set.

Constant Sample Sizes When the subgroup sample sizes are constant (= n), it may be preferable to compute cusums that are scaled in the same units as the data. Refer to Montgomery (1996) and Wadsworth, Stephens, and Godfrey (1986). To request this, specify the DATAUNITS option. Cusums are then computed as

$$S_t = \max(0, S_{t-1} + (\bar{X}_i - (\mu_0 + k\sigma/\sqrt{n})))$$

for δ >0 and the equation

$$S_t = \max(0, S_{t-1} - (\bar{X}_i - (\mu_0 - k\sigma/\sqrt{n})))$$

for $\delta < 0$. In either case, a shift is signaled if S_t exceeds $h' = h\sigma/\sqrt{n}$. Wadsworth, Stephens, and Godfrey (1986) use the symbol H for h'.

If the subgroup sample sizes are not constant, you can specify a constant nominal sample size n with the LIMITN= option or the variable _LIMITN_ in a LIMITS= data set. In this case, only those subgroups with sample size n are analyzed unless you also specify the option ALLN. You can further specify the option NMARKERS to request special symbol markers for points corresponding to sample sizes not equal to n.

Two-Sided Cusum Schemes

If the cusum scheme is two-sided, the cumulative sum S_t plotted for the tth subgroup is

$$S_t = S_{t-1} + z_t$$

for t=1, 2, . . . , n. Here S_0=0, and the term z_t is calculated as

$$z_t = (\bar{X}_t - \mu_0)/(\sigma/\sqrt{n_t})$$

where $\bar{X}_t$ is the tth subgroup average, and n_t is the tth subgroup sample size. If the subgroup samples consist of individual measurements x_t, the term z_t simplifies to

$$z_t = (x_t - \mu_0)/\sigma$$

Since the first equation can be rewritten as

$$S_t = \sum_{i=1}^{t} z_i = \sum_{i=1}^{t} (\bar{X}_i - \mu_0)/\sigma_{\bar{X}_i}$$

the sequence S_t cumulates standardized deviations of the subgroup averages from the target mean μ_0.

In many applications, the subgroup sample sizes n_i are constant ($n_i = n$), and the equation for S_t can be simplified.

$$S_t = (1/\sigma_{\bar{X}}) \sum_{i=1}^{t} (\bar{X}_i - \mu_0) = (\sqrt{n}/\sigma) \sum_{i=1}^{t} (\bar{X}_i - \mu_0)$$

In some applications, it may be preferable to compute S_t as

$$S_t = \sum_{i=1}^{t} (\bar{X}_i - \mu_0)$$

which is scaled in the same units as the data. Refer to Montgomery (1996), Wadsworth, Stephens, and Godfrey (1986), and American Society for Quality Control (1983). If the subgroup sample sizes are constant (= n) and if you specify the DATAUNITS option in the XCHART statement, the CUSUM procedure computes cusums using the final equation above. In this case, the procedure rescales the V-mask parameters h and k to $h' = h\sigma/\sqrt{n}$ and $k' = k\sigma/\sqrt{n}$, respectively. Wadsworth, Stephens, and Godfrey (1986) use the symbols F for k' and H for h'.

If the subgroup sample sizes are not constant, you can specify a constant nominal sample size n with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set. In this case, only those subgroups

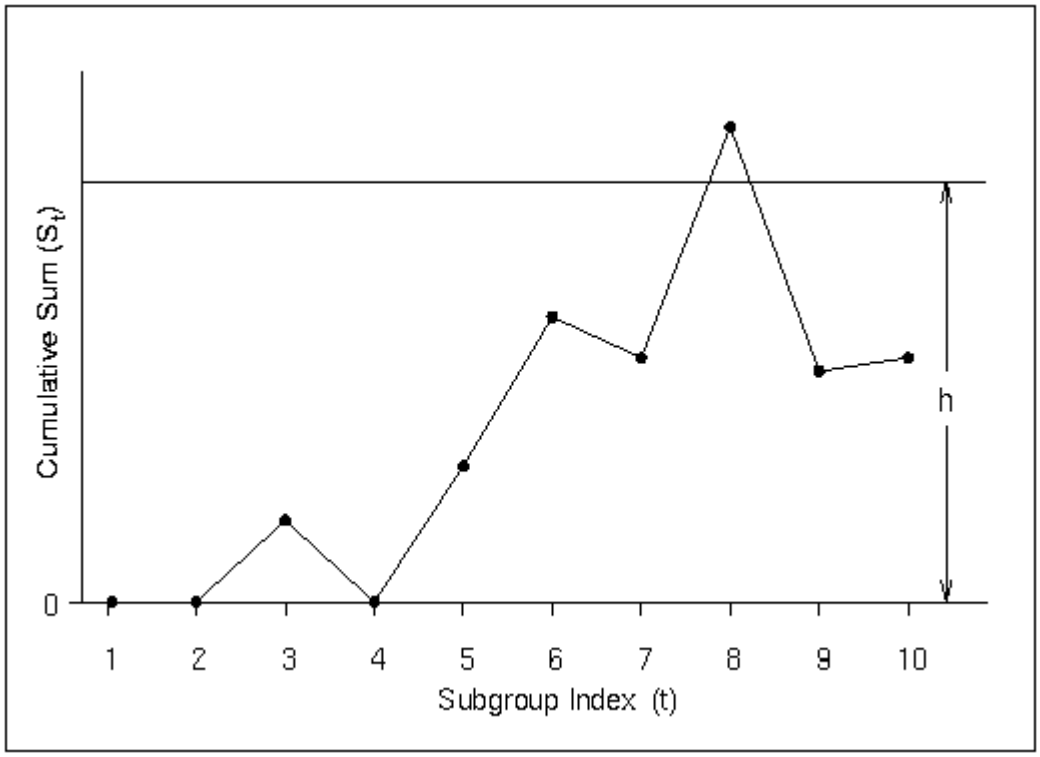
with sample size n are analyzed unless you also specify the option ALLN. You can further specify the option NMARKERS to request special symbol markers for points corresponding to sample sizes not equal to n.

If the process is in control and the mean μ is at or near the target μ_0, the points will not exhibit a trend since positive and negative displacements from μ_0 tend to cancel each other. If μ shifts in the positive direction, the points exhibit an upward trend, and if μ shifts in the negative direction, the points exhibit a downward trend.

Defining the Decision Interval for a One-Sided Cusum Scheme

The height of the decision interval is h, expressed as a multiple of the standard error of the subgroup mean. You can specify h with the H= option in the XCHART statement or with the variable _H_ in a LIMITS= data set. The decision interval is displayed as a horizontal line on the cusum chart, as illustrated in Figure 7.13.

Figure 7.13 Decision Interval



Interpreting One-Sided Cusum Charts

A shift or out-of-control condition is signaled at time t if the cusum S_t plotted at time t exceeds the decision interval line.

Defining the V-Mask for a Two-Sided Cusum Scheme

The dimensions of the V-mask can be specified using two distinct sets of two parameters.

- θ, defined as half of the angle formed by the V-mask arms, and d, the distance between the origin and the vertex, as shown in Figure 7.14. This parameterization is used by many authors, including Johnson and Leone (1962, 1974) and Montgomery (1996).

- h, the vertical distance between the origin and the upper (or lower) V-mask arm, and k, the rise (drop) in the lower (upper) arm corresponding to an interval of one subgroup unit on the horizontal axis. You can specify the definition of an interval with the INTERVAL= option. This parameterization is used by Lucas (1976) and Wadsworth, Stephens, and Godfrey (1986). Lucas (1976) uses the symbols h^* for h and k^* for k, and Wadsworth, Stephens, and Godfrey (1986) use the symbol f in place of k.

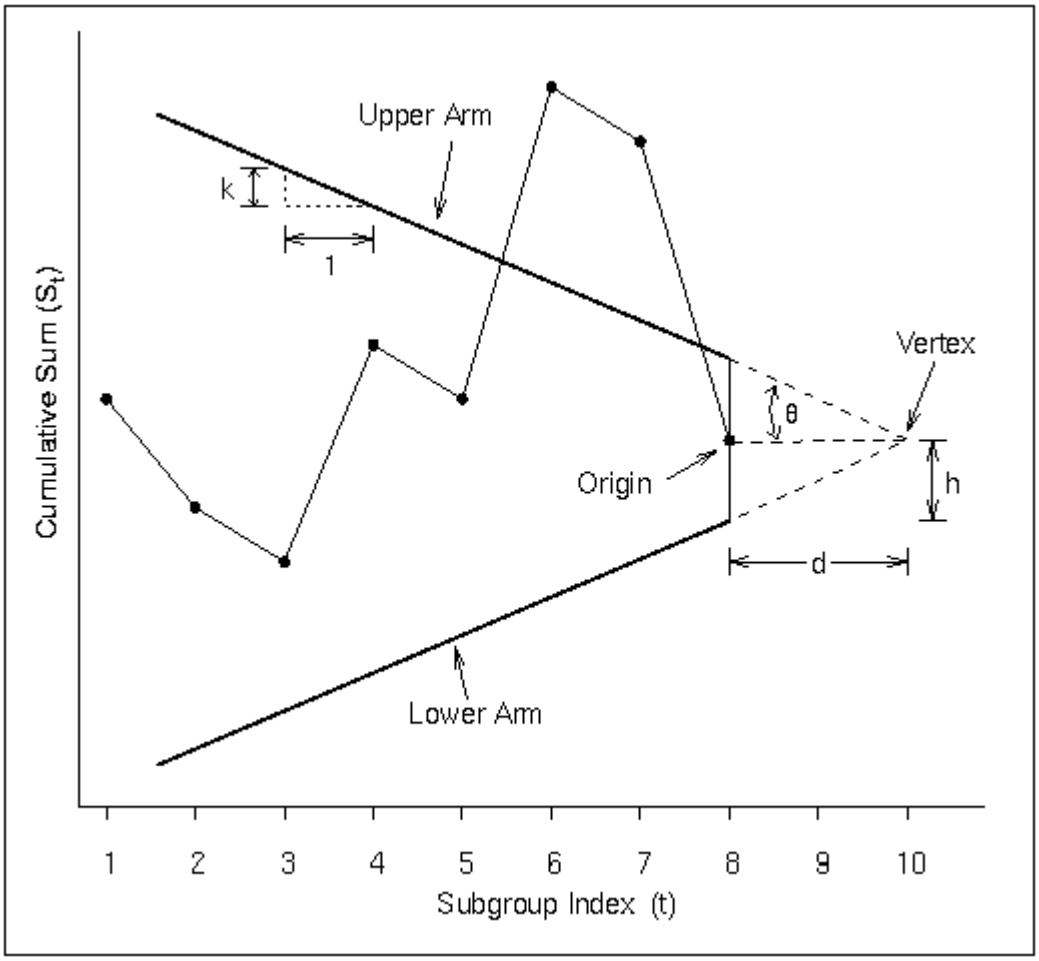
The two parameterizations are related by the equations

$$\theta = \arctan(k/a)$$

$$d = h/k$$

where the aspect ratio a is the number of units on the vertical axis corresponding to one unit on the horizontal axis. The CUSUM procedure uses the h and k parameterization because it eliminates the need for working with aspect ratios. Furthermore, h and k are also useful for average run length computations and for parameterizing one-sided cusum schemes.

Figure 7.14 V-Mask Parameters



You can specify the V-mask in two ways:

- geometrically, by providing h and k (or simply h) with the H= and K= options or with the variables _H_ and _K_ in a LIMITS= data set
- in terms of error probabilities, by providing α and β (or simply α) with the ALPHA= and BETA= options or with the variables _ALPHA_ and _BETA_ in a LIMITS= data set. The SIGMAS= option is an alternative to the ALPHA= option, and the variable _SIGMAS_ is an alternative to the variable _ALPHA_ (if the READSIGMAS option is specified).

If you provide α and β, h and k are computed using the formulas

$$h = |\delta|^{-1} \log((1-\beta)/(\alpha/2))$$

$$k = |\delta|/2$$

If you provide α but not β, h and k are computed using the formulas

$$h = -|\delta|^{-1} \log(\alpha/2)$$

$$k = |\delta|/2$$

In the preceding equations, the error probability α is divided by two because two-sided deviations from the target mean are detected. Refer to Johnson and Leone (1962, 1974).

Interpreting Two-Sided Cusum Charts

The origin of the V-mask is located at the most recently plotted point, as illustrated in Figure 7.14. As additional data are collected and the cumulative sum sequence is updated, the origin is relocated at the newest point. A shift or out-of-control condition is signaled at time t if one or more of the points plotted up to time t cross an arm of the V-mask. An upward shift is signaled by points crossing the lower arm, and a downward shift is signaled by points crossing the upper arm. The time at which the shift occurred corresponds to the time at which a distinct change is observed in the slope of the plotted points.

Designing a Cusum Scheme

There are three main methods for designing a cusum scheme: the *average run length (ARL) approach*, the *error probability approach*, and the *economic design approach*.

Average Run Length (ARL) Approach

With the ARL approach, the parameters h and k are chosen to yield desired average run lengths when the process is operating at the target mean and when a shift of magnitude δ has occurred. The average run length is the expected number of samples taken before an out-of-control condition is signaled. Ideally, the ARL should be long when $\mu = \mu_0$ and short when μ shifts away from μ_0.

The ARL method typically involves the use of a table or nomogram. Refer to Kemp (1961), Van Dobben de Bruyn (1968), Goel and Wu (1971), Duncan (1974), Lucas (1976), Montgomery (1996), and Wadsworth, Stephens, and Godfrey (1986).

For one-sided charts, average run lengths are tabulated as a function of h, k, and δ in Table 7.5. No headstart is assumed in this table. For two-sided charts, average run lengths are tabulated as a function of h, k, and δ in Table 7.6, which is formatted similarly to Table 2 given by Lucas (1976).

The ARLs in Table 7.5 and Table 7.6 were calculated with the DATA step function CUSUMARL (see the section "CUSUMARL Function" on page 2224). This function uses the method of Goel and Wu (1971). You can use this function to generate more detailed, interpolated versions of the tables or to compute ARLs with headstart values.

It can be shown that the two-sided (V-mask) cusum scheme parameterized by h and k is equivalent to two simultaneously operating one-sided cusum schemes, one that computes an upper cusum and one that computes a lower cusum. Both one-sided schemes use the same parameters h and k.

You can specify h, k, and δ with the options H=, K=, and DELTA= or with the variables _H_, _K_, and _DELTA_ in a LIMITS= data set. The reference value k is optional, and its default value is $k = |\delta|/2$, referred to as the *central reference value*.

Error Probability Approach

This approach is available only for two-sided cusum schemes. Values of α (the probability of incorrectly signaling the occurrence of a shift) and β (the probability of failing to detect a shift) are specified, and h and k are computed from α and β as described in "Defining the V-Mask for a Two-Sided Cusum Scheme" on page 588. The error probability approach interprets the cusum as a sequence of reversed sequential probability ratio tests. Refer to Johnson (1961), Johnson and Leone (1962, 1974), Van Dobben de Bruyn (1968), Montgomery (1996), and Wadsworth, Stephens, and Godfrey (1986).

Although the error probability method is intuitively appealing, the actual error probabilities achieved may not be close to those specified since the V-mask does not provide for an acceptance region. This has been pointed out by various authors, including Johnson (1961) and Van Dobben de Bruyn (1968). If you follow this approach, it is recommended that you examine the average run lengths for the cusum scheme (these are tabulated by the TABLESUMMARY option and are saved in OUTLIMITS= data sets).

You can specify α and β with the ALPHA= and BETA= options or with the variables _ALPHA_ and _BETA_ in a LIMITS= data set. It is not necessary to specify β, and the interpretation of β is somewhat questionable. The SIGMAS= option is an alternative to the ALPHA= option, and the variable _SIGMAS_ is an alternative to the variable _ALPHA_ (if you specify the READSIGMAS option).

Economic Design

The parameters n, h, and k are chosen so that the long-run average cost of the cusum scheme is minimized. Refer to Chiu (1974), Montgomery (1980), Svoboda (1991), and Ho and Case (1994) for reviews of the literature on economic design. This approach typically requires numerical optimization techniques, which are available in SAS/IML software and in the NLP procedure in SAS/OR software.

You can pass the optimal parameters to the CUSUM procedure as values of the variables _LIMITN_, _H_, and _K_ in a LIMITS= data set.

Table 7.5 Average Run Lengths for One-Sided V-Mask Cusum Charts as a Function of h, k, and δ.

Parameters		δ **(shift in mean)**										
h	**k**	0.00	0.25	0.50	0.75	1.00	1.50	2.00	2.50	3.00	4.00	5.00
2.50	0.25	27.27	13.43	7.96	5.42	4.06	2.71	2.06	1.68	1.42	1.11	1.01
4.00	0.25	77.08	26.68	13.29	8.38	6.06	3.91	2.93	2.38	2.05	1.61	1.23
6.00	0.25	350.80	51.34	20.90	12.37	8.73	5.51	4.07	3.26	2.74	2.13	1.90
8.00	0.25	736.78	84.00	28.76	16.37	11.39	7.11	5.21	4.15	3.48	2.67	2.14
10.00	0.25	2071.51	124.66	36.71	20.37	14.06	8.71	6.36	5.04	4.20	3.20	2.65
2.00	0.50	38.55	18.19	10.00	6.32	4.45	2.74	1.99	1.58	1.32	1.07	1.01
3.00	0.50	117.60	39.47	17.35	9.68	6.40	3.75	2.68	2.12	1.77	1.31	1.07
4.00	0.50	335.37	77.08	26.68	13.29	8.38	4.75	3.34	2.62	2.19	1.71	1.31
5.00	0.50	930.89	141.69	38.01	17.05	10.38	5.75	4.01	3.11	2.57	2.01	1.69
6.00	0.50	2553.11	250.80	51.34	20.90	12.37	6.75	4.68	3.62	2.98	2.24	1.95
1.50	0.75	42.57	21.09	11.59	7.09	4.78	2.73	1.90	1.48	1.24	1.04	1.00
2.25	0.75	139.71	51.46	22.38	11.66	7.13	3.73	2.51	1.91	1.56	1.16	1.02
3.00	0.75	442.80	117.60	39.47	17.35	9.68	4.73	3.12	2.36	1.93	1.41	1.11
3.75	0.75	1375.71	258.96	65.65	24.16	12.37	5.73	3.71	2.79	2.27	1.72	1.31
4.50	0.75	4251.69	559.95	105.12	32.09	15.15	6.73	4.31	3.21	2.59	1.97	1.60
1.00	1.00	35.29	19.22	11.21	7.03	4.75	2.63	1.78	1.38	1.17	1.02	1.00
1.50	1.00	93.85	42.57	21.09	11.59	7.09	3.50	2.24	1.66	1.34	1.07	1.01
2.00	1.00	258.67	94.34	38.55	18.19	10.00	4.45	2.74	1.99	1.58	1.16	1.02
2.50	1.00	716.00	205.97	68.19	27.27	13.43	5.42	3.25	2.34	1.85	1.31	1.07
3.00	1.00	1962.79	442.80	117.60	39.47	17.35	6.40	3.75	2.68	2.12	1.52	1.16
3.50	1.00	5341.40	943.73	199.57	55.69	21.76	7.39	4.25	3.01	2.37	1.73	1.31
0.70	1.50	67.72	36.03	20.26	12.07	7.63	3.66	2.18	1.55	1.25	1.04	1.00
1.10	1.50	184.28	86.36	42.72	22.50	12.74	5.17	2.80	1.86	1.43	1.08	1.01
1.50	1.50	549.69	221.49	93.85	42.57	21.09	7.09	3.50	2.24	1.66	1.16	1.02
1.90	1.50	1762.09	595.61	210.95	80.54	34.26	9.38	4.26	2.64	1.92	1.29	1.05
2.30	1.50	5897.30	1638.15	476.90	151.04	54.47	12.00	5.03	3.04	2.20	1.45	1.12

Table 7.6 Average Run Lengths for Two-Sided V-Mask Cusum Charts as a Function of h, k, and δ.

Parameters		δ **(shift in mean)**										
h	**k**	0.00	0.25	0.50	0.75	1.00	1.50	2.00	2.50	3.00	4.00	5.00
2.50	0.25	13.64	11.22	7.67	5.38	4.06	2.71	2.06	1.68	1.42	1.11	1.01
4.00	0.25	38.54	24.71	13.20	8.38	6.06	3.91	2.93	2.38	2.05	1.61	1.23
6.00	0.25	125.40	50.33	20.89	12.37	8.73	5.51	4.07	3.26	2.74	2.13	1.90
8.00	0.25	368.39	83.63	28.76	16.37	11.39	7.11	5.21	4.15	3.48	2.67	2.14
10.00	0.25	1035.75	124.55	36.71	20.37	14.06	8.71	6.36	5.04	4.20	3.20	2.65
2.00	0.50	19.27	15.25	9.63	6.27	4.44	2.74	1.99	1.58	1.32	1.07	1.01
3.00	0.50	58.80	36.24	17.20	9.67	6.40	3.75	2.68	2.12	1.77	1.31	1.07
4.00	0.50	167.68	74.22	26.63	13.29	8.38	4.75	3.34	2.62	2.19	1.71	1.31
5.00	0.50	465.44	139.49	38.00	17.05	10.38	5.75	4.01	3.11	2.57	2.01	1.69
6.00	0.50	1276.55	249.26	51.34	20.90	12.37	6.75	4.68	3.62	2.98	2.24	1.95
1.50	0.75	21.28	17.22	11.01	7.00	4.77	2.73	1.90	1.48	1.24	1.04	1.00
2.25	0.75	69.85	45.97	22.04	11.63	7.13	3.73	2.51	1.91	1.56	1.16	1.02
3.00	0.75	221.40	110.95	39.31	17.34	9.68	4.73	3.12	2.36	1.93	1.41	1.11
3.75	0.75	687.85	251.56	65.58	24.16	12.37	5.73	3.71	2.79	2.27	1.72	1.31
4.50	0.75	2125.85	552.11	105.09	32.09	15.15	6.73	4.31	3.21	2.59	1.97	1.60
1.00	1.00	17.65	15.03	10.39	6.88	4.72	2.63	1.78	1.38	1.17	1.02	1.00
1.50	1.00	46.92	35.70	20.31	11.49	7.07	3.50	2.24	1.66	1.34	1.07	1.01
2.00	1.00	129.34	84.00	37.93	18.14	10.00	4.45	2.74	1.99	1.58	1.16	1.02
2.50	1.00	358.00	191.48	67.76	27.25	13.43	5.42	3.25	2.34	1.85	1.31	1.07
3.00	1.00	981.39	423.29	117.32	39.47	17.35	6.40	3.75	2.68	2.12	1.52	1.16
3.50	1.00	2670.70	917.89	199.40	55.69	21.76	7.39	4.25	3.01	2.37	1.73	1.31
0.70	1.50	33.86	28.41	18.90	11.84	7.59	3.66	2.18	1.55	1.25	1.04	1.00
1.10	1.50	92.14	71.41	40.91	22.29	12.71	5.17	2.80	1.86	1.43	1.08	1.01
1.50	1.50	274.84	191.58	91.58	42.39	21.07	7.09	3.50	2.24	1.66	1.16	1.02
1.90	1.50	881.05	536.07	208.31	80.41	34.25	9.38	4.26	2.64	1.92	1.29	1.05
2.30	1.50	2948.65	1523.15	474.09	150.96	54.47	12.00	5.03	3.04	2.20	1.45	1.12

Cusum Charts Compared with Shewhart Charts

Although cusum charts and Shewhart charts are both used to detect shifts in the process mean, there are important differences in the two methods.

- Each point on a Shewhart chart is based on information for a single subgroup sample or measurement. Each point on a cusum chart is based on information from all samples (measurements) up to and including the current sample (measurement).

- On a Shewhart chart, upper and lower control limits are used to decide whether a point signals an out-of-control condition. On a cusum chart, the limits take the form of a decision interval or a V-mask.
- On a Shewhart chart, the control limits are commonly computed as 3σ limits. On a cusum chart, the limits are determined from average run length specifications, specified error probabilities, or an economic design.

A cusum chart offers several advantages over a Shewhart chart.

- A cusum chart is more efficient for detecting small shifts in the process mean, in particular, shifts of 0.5 to 2 standard deviations from the target mean (refer to Montgomery 1996). Lucas (1976) noted that "a V-mask designed to detect a 1σ shift will detect it about four times as fast as a competing Shewhart chart."
- Shifts in the process mean are visually easy to detect on a cusum chart since they produce a change in the slope of the plotted points. The point at which the slope changes is the point at which the shift has occurred.

These advantages are not as pronounced if the Shewhart chart is augmented by the tests for special causes described by Nelson (1984, 1985). Also see "Tests for Special Causes: SHEWHART Procedure" on page 2119. Moreover,

- cusum schemes are more complicated to design.
- a cusum chart can be slower to detect large shifts in the process mean.
- it can be difficult to interpret point patterns on a cusum chart since the cusums are correlated.

Methods for Estimating the Standard Deviation

It is recommended practice to provide a stable estimate or standard value for σ with either the SIGMA0= option or the variable _STDDEV_ in a LIMITS= data set. However, if such a value is not available, you can compute an estimate $\hat{\sigma}$ from the data, as described in this section.

This section provides formulas for various methods used to estimate the standard deviation σ. One method is applicable with individual measurements, and three are applicable with subgrouped data. The methods can be requested with the SMETHOD= option.

Method for Individual Measurements

When the cumulative sums are calculated from individual observations

$$x_1, x_2, \ldots, x_N$$

rather than subgroup samples of two or more observations, the CUSUM procedure estimates σ as $\sqrt{\hat{\sigma}^2}$, where

$$\hat{\sigma}^2 = \frac{1}{2(N-1)} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2$$

where N is the number of observations. Wetherill (1977) states that the estimate of the variance is biased if the measurements are autocorrelated.

Note that you can compute alternative estimates (for instance, robust estimates or estimates based on variance components models) by analyzing the data with SAS modeling procedures or your own DATA step program. Such estimates can be passed to the CUSUM procedure as values of the variable _STDDEV_ in a LIMITS= data set.

NOWEIGHT Method for Subgroup Samples

This method is the default for cusum charts for subgrouped data. The estimate is

$$\hat{\sigma} = \frac{(s_1/c_4(n_1)) + \cdots + (s_N/c_4(n_N))}{N}$$

where n_i is the sample size of the ith subgroup, N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$s_i = \sqrt{(1/(n_i - 1)) \sum_{j=1}^{n_i} (x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

where $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is

$$\mathrm{E}(s_i) = c_4(n_i)\sigma$$

Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the *ASTM Manual on Presentation of Data and Control Chart Analysis*.

MVLUE Method for Subgroup Samples

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed, as introduced by Burr (1969, 1976). This estimate is a weighted average of unbiased estimates of σ of the form

$$s_i/c_4(n_i)$$

where

s_i	is the standard deviation of the ith subgroup.
$c_4(n_i)$	is the unbiasing factor defined previously.
n_i	is the ith subgroup sample size, $i = 1, 2, \ldots, N$.
N	is the number of subgroups for which $n_i \geq 2$.

The estimate is

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where $h_i = c_4^2(n_i)/(1 - c_4^2(n_i))$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$.

The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes and is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate (NOWEIGHT).

RMSDF Method for Subgroup Samples

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where

n_i	is the sample size of the ith subgroup.
N	is the number of subgroups for which $n_i \geq 2$.
s_i	is the sample standard deviation of the ith subgroup.
$c_4(n_i)$	is the unbiasing factor defined previously.
n	is equal to $(n_1 + \cdots + n_N) - (N - 1)$.

The weights in the root-mean-square expression are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, as noted by Burr (1969), "the constancy of σ is the very thing under test," and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Output Data Sets

OUTLIMITS= Data Set

When you save the parameters for the cusum scheme in an OUTLIMITS= data set, the following variables are included:

Table 7.7 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of Type 1 error
ARLIN	Average run length for zero shift
ARLOUT	Average run length for shift of δ
BETA	Probability (β) of Type 2 error
DELTA	Shift (δ) to be detected
_H_	Decision interval h for one-sided scheme; distance h between origin and upper arm V-mask for two-sided scheme
HSTART	Headstart value
INDEX	Optional identifier for cusum parameters (if the OUTINDEX= option is specified)
_K_	Reference value k for one-sided scheme; slope of lower V-mask arm for two-sided scheme
LIMITN	Nominal sample size for cusum scheme
MEAN	Estimated process mean ($\overline{\overline{X}}$)
_MU0_	Target mean μ_0
ORIGIN	Origin of V-mask
SCHEME	Type of scheme ('ONESIDED' or 'TWOSIDED')

Table 7.7 *continued*

Variable	Description
SIGMAS	$z_{1-\alpha/2}$
STDDEV	Estimated or known standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in XCHART statement
TYPE	Type ('ESTIMATE' or 'STANDARD') of _STDDEV_
VAR	*Process* specified in XCHART statement

Notes:

1. If the subgroup sample sizes vary, the special missing value V is assigned to the variable _LIMITN_.
2. If a V-mask is specified with SIGMAS=k, _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If a V-mask is specified with ALPHA=α, _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XCHART statement. For an example, see "Saving Cusum Scheme Parameters" on page 565.

OUTHISTORY= Data Set

When you save subgroup summary statistics in an OUTHISTORY= data set, the following variables are included:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a cusum variable named by *process* suffixed with *C*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup summary statistics are created for each *process* specified in the XCHART statement. For example, consider the following statements:

```
proc cusum data=Steel limits=Stparm;
   xchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary would contain nine variables named Lot, WidthX, WidthS, WidthN, WidthC, DiameterX, DiameterS, DiameterN, and DiameterC.

Additionally, if specified, the following variables are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example creating an OUTHISTORY= data set, see "Saving Summary Statistics" on page 560.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup means, subgroup sample sizes, cusums, and cusum limits. Table 7.8 lists the variables that are included.

Table 7.8 OUTTABLE= Data Set Variables

Variable	Description
CUSUM	Cumulative sum
EXLIM	Decision interval or V-mask arm exceeded
_H_	Decision interval
MASKL	Lower arm of V-mask
MASKU	Upper arm of V-mask
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample size
SUBX	Subgroup mean
SUBS	Subgroup standard deviation
VAR	*Process* specified in XCHART statement

In addition, the following variables are saved if specified:

- BY variables
- *block-variables*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)
- *symbol-variable*

Note that the variables _VAR_ and _EXLIM_ are character variables of length eight. The variable _PHASE_ is a character variable of length 16.

ODS Tables

The following table summarizes the ODS tables that you can request with the XCHART statement.

Table 7.9 ODS Tables Produced with the XCHART Statement

Table Name	Description	Options
CompCusum	Computational form of the cusum scheme	TABLEALL, TABLECOMP
Parameters	Cusum parameters and computed average run lengths	TABLEALL, TABLESUMMARY
XChartSummary	Cusum chart summary statistics	TABLEALL, TABLECHART, TABLEOUT

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. XCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the XCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 7.10.

Table 7.10 ODS Graphics Produced by the XCHART Statement

ODS Graph Name	Plot Description
XChart	Cusum chart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (measurements) from a DATA= data set specified in the PROC CUSUM statement. Each *process* specified in the XCHART statement must be a SAS variable in the DATA= data set. The values of this variable are typically measurements of a quality characteristic taken on items in subgroup samples indexed by the values of the subgroup variable. The *subgroup-variable* specified in the XCHART statement must also be a SAS variable in the DATA= data set. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

Each observation in a DATA= data set should contain a raw measurement for each *process* and a value for the subgroup variable. If the ith subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the ith subgroup. For example, if each of 30 subgroup samples contains five items, the DATA= data set should contain 150 observations.

By default, the CUSUM procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option in the XCHART statement.

For an example of a DATA= data set, see "Creating a V-Mask Cusum Chart from Raw Data" on page 555.

LIMITS= Data Set

You can read cusum scheme parameters from a LIMITS= data set specified in the PROC CUSUM statement. As an alternative to specifying the parameters with options, a LIMITS= data set provides the following advantages: it facilitates reusing a permanently saved set of parameters, reading a distinct set of parameters for each *process* specified in the XCHART statement, and keeping track of multiple sets of parameters for the same *process* over time.

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the CUSUM procedure. Such data sets always contain the variables required for a LIMITS= data set; consequently, this is the easiest way to construct a LIMITS= data set.

A LIMITS= data set can also be created directly using a DATA step. The variables required for the data set depend on the type of cusum scheme and how the scheme is specified. The following restrictions apply:

- The variables _VAR_, _SUBGRP_, _DELTA_, and _MU0_ are required.
- For a one-sided cusum scheme, _H_ is required.
- For a two-sided cusum scheme, one of the following three variables is required: _ALPHA_, _H_, or _SIGMAS_.

- If you plan to use the READINDEX= option, the variable _INDEX_ is required; otherwise, it is optional.
- For a one-sided scheme, the variable _SCHEME_ is required; otherwise, it is optional.
- If you want to provide a value for the process standard deviation σ, the variable _STDDEV_ is required; otherwise, it is optional.

Variable names in a LIMITS= data set are predefined; the procedure reads only variables with these predefined names. With the exception of BY variables, all names start and end with an underscore. In addition, note the following:

- The variables _VAR_, _SUBGRP_, _TYPE_, and _SCHEME_ must be character variables of length eight. The variable _INDEX_ must be a character variable of length 16.
- The variable _TYPE_ is a bookkeeping variable that uses the values 'ESTIMATE' and 'STANDARD' to record whether the value of _STDDEV_ represents an estimate or standard (known) value.
- BY variables are required if specified with a BY statement.

For an example of reading control limit information from a LIMITS= data set, see "Reading Cusum Scheme Parameters" on page 567.

HISTORY= Data Set

Instead of reading raw data from a DATA= data set, you can read subgroup summary statistics from a HISTORY= data set specified in the PROC CUSUM statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the CUSUM, MACONTROL, or SHEWHART procedures or to read output data sets created with SAS summarization procedures such as PROC MEANS. A HISTORY= data set must contain the following variables:

- *subgroup-variable*
- subgroup mean variable for each *process*
- subgroup standard deviation variable for each *process*
- subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* concatenated with the special suffix characters *X*, *S*, and *N* respectively.

For example, consider the following statements:

```
proc cusum history=Steel limits=Steelparm;
   xchart (Weight Yieldstrength)*Batch;
run;
```

The data set Steel must contain the variables Batch, WeightX, WeightS, WeightN, YieldstrengthX, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the CUSUM procedure reads all of the observations in a HISTORY= data set. However, if the HISTORY= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as phases) by specifying the READPHASES= option.

For an example of reading summary information from a HISTORY= data set, see "Creating a V-Mask Cusum Chart from Subgroup Summary Data" on page 558.

Missing Values

An observation read from a DATA= or HISTORY= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: XCHART Statement

This section provides advanced examples of the XCHART statement.

Example 7.1: Cusum and Standard Deviation Charts

NOTE: See *Cusum and Standard Deviation Charts* in the SAS/QC Sample Library.

When you are working with subgrouped data, it can be helpful to accompany a cusum chart for means with a Shewhart *s* chart for monitoring the variability of the process. This example creates this combination for the variable Weight in the data set Oil (see "Creating a V-Mask Cusum Chart from Raw Data" on page 555).

The first step is to create a one-sided cusum chart for means that detects a shift of one standard error ($\delta = -1$) below the target mean.

```
proc cusum data=Oil;
   xchart Weight*Hour /
      nochart
      mu0=8.100        /* target mean for process  */
      sigma0=0.050     /* known standard deviation */
      delta=-1         /* shift to be detected     */
      h=3              /* cusum parameter h        */
      k=0.5            /* cusum parameter k        */
      scheme=onesided
      outtable = Tabcusum
         ( drop   = _var_ _subn_ _subx_ _exlim_
           rename = ( _cusum_ = _subx_ _h_ = _uclx_ ) )
      ;
run;
```

The results are saved in an OUTTABLE= data set named Tabcusum. The cusum variable (_CUSUM_) and the decision interval variable (_H_) are renamed to _SUBX_ and _LCLX_ so that they can later be read by the SHEWHART procedure.

The next step is to construct a Shewhart $\bar{X}$ and s chart for Weight and save the results in a data set named Tabxscht.

```
proc shewhart data=Oil;
   xschart Weight*Hour /
      nochart
      outtable = Tabxscht
         ( drop = _subx_  _uclx_ );
run;
```

Note that the variables _SUBX_ and _UCLX_ are dropped from Tabxscht.

The third step is to merge the data sets Tabcusum and Tabxscht.

```
data taball;
   merge Tabxscht Tabcusum; by Hour;
   _mean_ = _uclx_ * 0.5;
   _lclx_ = 0.0;
run;
```

The variable _LCLX_ is assigned the role of the lower limit for the cusums, and the variable _MEAN_ is assigned a dummy value. Now, TABALL, which is listed in Output 7.1.1, has the structure required for a TABLE= data set used with the XSCHART statement in the SHEWHART procedure (see "TABLE= Data Set" on page 1956).

Output 7.1.1 Listing of the Data Set TABALL

Obs	_VAR_	Hour	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_MEAN_	_STDDEV_	_EXLIM_	_LCLS_
1	Weight	1	3	4	4	0	1.5	0.05		0
2	Weight	2	3	4	4	0	1.5	0.05		0
3	Weight	3	3	4	4	0	1.5	0.05		0
4	Weight	4	3	4	4	0	1.5	0.05		0
5	Weight	5	3	4	4	0	1.5	0.05		0
6	Weight	6	3	4	4	0	1.5	0.05		0
7	Weight	7	3	4	4	0	1.5	0.05		0
8	Weight	8	3	4	4	0	1.5	0.05		0
9	Weight	9	3	4	4	0	1.5	0.05		0
10	Weight	10	3	4	4	0	1.5	0.05		0
11	Weight	11	3	4	4	0	1.5	0.05		0
12	Weight	12	3	4	4	0	1.5	0.05		0

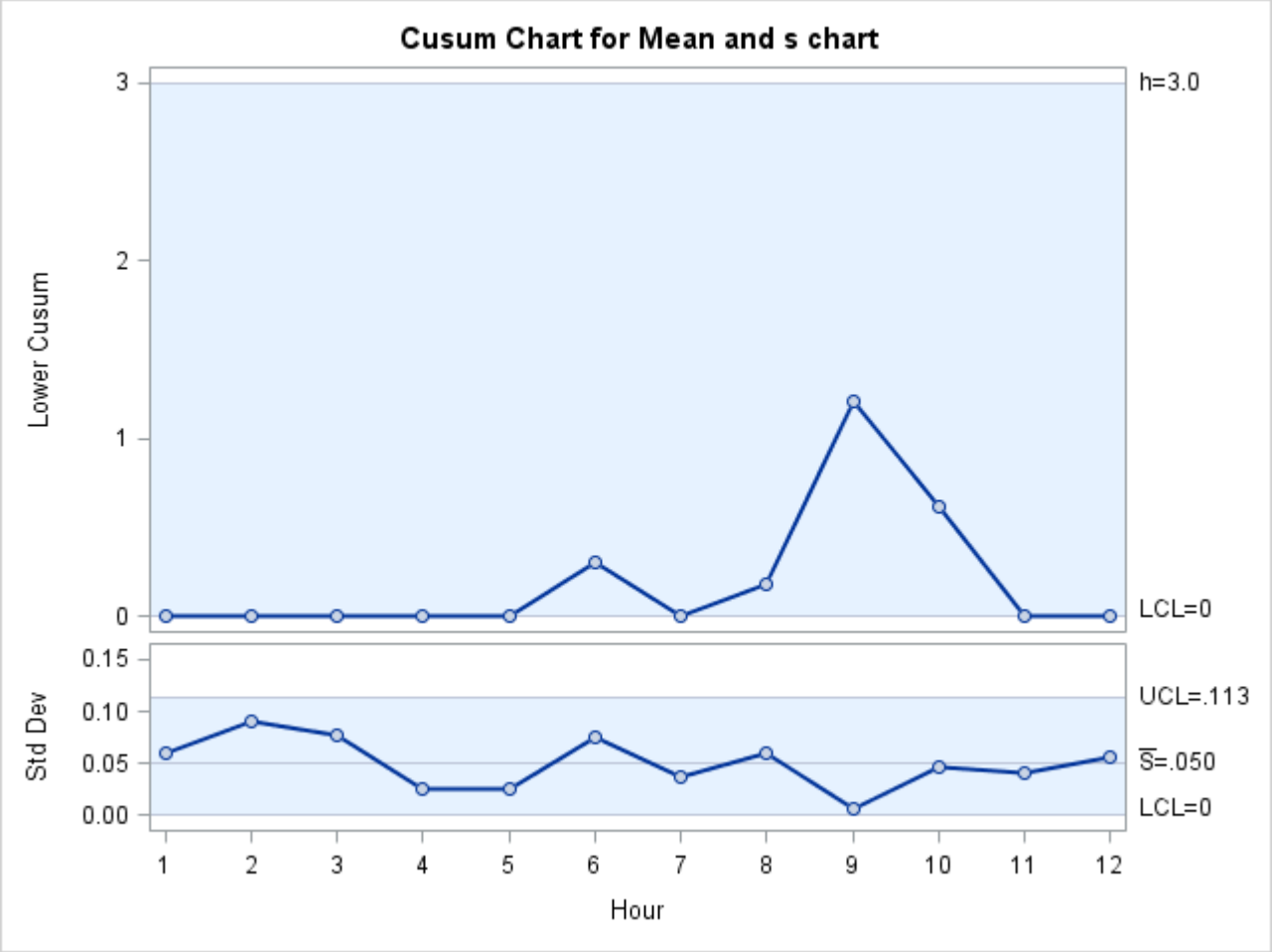
Obs	_SUBS_	_S_	_UCLS_	_EXLIMS_	_subx_	_uclx_
1	0.059640	0.049943	0.11317		0.00	3
2	0.090220	0.049943	0.11317		0.00	3
3	0.076346	0.049943	0.11317		0.00	3
4	0.025552	0.049943	0.11317		0.00	3
5	0.026500	0.049943	0.11317		0.00	3
6	0.075617	0.049943	0.11317		0.30	3
7	0.037242	0.049943	0.11317		0.00	3
8	0.059290	0.049943	0.11317		0.18	3
9	0.005737	0.049943	0.11317		1.21	3
10	0.046522	0.049943	0.11317		0.62	3
11	0.040542	0.049943	0.11317		0.00	3
12	0.056103	0.049943	0.11317		0.00	3

The final step is to use the SHEWHART procedure to read TABALL as a TABLE= data set and to display the cusum and *s* charts.

```
ods graphics on;
title 'Cusum Chart for Mean and s chart';
proc shewhart table=taball;
   xschart Weight * Hour /
      nolimitslegend
      ucllabel = 'h=3.0'
      odstitle = title
      markers
      noctl
      split = '/'
      nolegend ;
   label _subx_ = 'Lower Cusum/Std Dev';
run;
```

The central line for the primary (cusum) chart is suppressed with the NOCTL option, and the default *3σ Limits* legend is suppressed with the NOLIMITLEGEND option. The charts are shown in Output 7.1.2.

Output 7.1.2 Combined Cusum Chart and s Chart



The process variability is stable, and there is no signal of a downward shift in the process mean.

Example 7.2: Upper and Lower One-Sided Cusum Charts

NOTE: See *Upper and Lower One-Sided Cusum Charts* in the SAS/QC Sample Library.

This example illustrates how to combine upper and lower one-sided cusum charts for means in the same display. As in the preceding example, OUTTABLE= data sets are created with the CUSUM procedure, and the display is created with the SHEWHART procedure.

The following statements analyze the variable Weight in the data set Oil (see "Creating a V-Mask Cusum Chart from Raw Data" on page 555). The first step is to compute and save upper and lower one-sided cusums for shifts of one standard error in the positive and negative directions.

```
proc cusum data=Oil;
   xchart Weight*Hour /
      nochart
      mu0=8.100         /* target mean for process  */
      sigma0=0.050      /* known standard deviation */
```

```
      delta=1          /* shift to be detected     */
      h=3              /* cusum parameter h        */
      k=0.5            /* cusum parameter k        */
      scheme=onesided
      outtable = tabupper
         ( drop   = _subx_ _subs_ _exlim_
           rename = ( _cusum_ = _subx_ _h_ = _uclx_ ) )
      ;
   xchart Weight*Hour /
      nochart
      mu0=8.100        /* target mean for process  */
      sigma0=0.050     /* known standard deviation */
      delta=-1         /* shift to be detected     */
      h=3              /* cusum parameter h        */
      k=0.5            /* cusum parameter k        */
      scheme=onesided
      outtable = tablower
         ( drop   = _var_ _subn_ _subx_ _subs_ _exlim_
           rename = ( _cusum_ = _subs_ _h_ = _ucls_ ) )
      ;
run;
```

Next, the OUTTABLE= data sets are merged.

```
data Tabboth;
   merge tabupper tablower; by Hour;
   _mean_   = _uclx_ * 0.5;
   _s_      = _ucls_ * 0.5;
   _lclx_   = 0.0;
   _lcls_   = 0.0;
run;
```

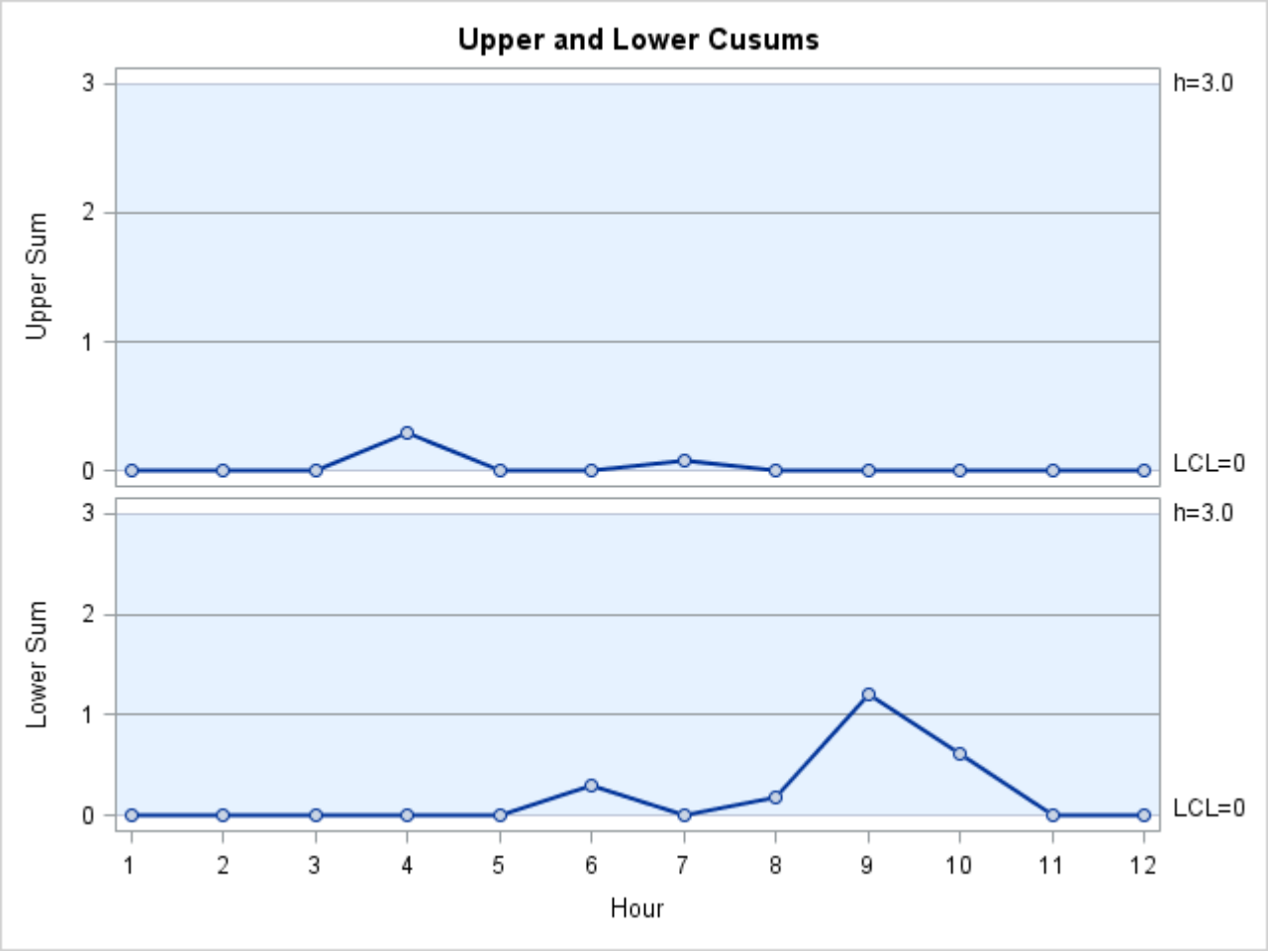
The variables _LCLX_ and _UCLX_ are assigned lower limits of zero for the cusums, and the variables _MEAN_ and _S_ are assigned dummy values. Now, Tabboth has the structure required for a TABLE= data set used with the XSCHART statement in the SHEWHART procedure (see "TABLE= Data Set" on page 1956).

The final step is to read Tabboth as a TABLE= data set with the SHEWHART procedure.

```
ods graphics on;
title 'Upper and Lower Cusums';
proc shewhart table=Tabboth;
   xschart Weight * Hour /
      nolimitslegend
      markers
      odstitle  = title
      ucllabel  = 'h=3.0'
      ucllabel2 = 'h=3.0'
      ypct1     = 50
      vref      = 1 2
      vref2     = 1 2
      noctl
      noctl2
      split = '/'
      nolegend ;
   label _subx_ = 'Upper Sum/Lower Sum';
run;
```

The combined display is shown in Output 7.2.1. There is no evidence of a shift in either direction.

Output 7.2.1 Upper and Lower One-Sided Cusum Charts



Example 7.3: Combined Shewhart–Cusum Scheme

NOTE: See *Combined Shewhart-Cusum Scheme* in the SAS/QC Sample Library.

Lucas and Crosier (1982) introduced a combined Shewhart-cusum scheme that is illustrated in this example. Also refer to Ryan (1989). The data set used here is Cans, which is created in "Creating a One-Sided Cusum Chart with a Decision Interval" on page 561.

The first step is to compute and save one-sided cusums to detect a positive shift from the mean.

```
proc cusum data=Cans;
   xchart Weight*Hour /
      nochart
      mu0    = 8.100     /* target mean for process  */
      sigma0 = 0.050     /* known standard deviation */
      delta  = 1         /* shift to be detected     */
      h      = 3         /* cusum parameter h        */
```

```
      k      = 0.5       /* cusum parameter k        */
      scheme = onesided
      outtable = Tabcus
         ( drop   = _var_ _subn_ _exlim_
           rename = ( _cusum_ = _subr_ _h_ = _uclr_ ) )
      ;
run;
```

Note that a headstart value is not used here but can be specified with the HSTART= option. Several variables in the OUTTABLE= data set are dropped or renamed so that they can later be read by the SHEWHART procedure.

The next step is to construct a Shewhart chart (not shown) for individual measurements.

```
proc shewhart data=Cans;
   irchart Weight*Hour /
      nochart
      mu0      = 8.100
      sigma0   = 0.050
      outtable = Tabx
         ( drop   = _subr_  _lclr_ _r_ _uclr_ );
   id comment;
run;
```

By default, 3σ limits are computed, but the multiple of σ can be modified with the SIGMAS= option. As before, the results are saved in an OUTTABLE= data set.

Next, the two OUTTABLE= data sets are merged.

```
data Combine;
   merge Tabx Tabcus; by Hour;
   _lclr_ = 0.0;
   _r_    = 0.5 * _uclr_;
run;
```

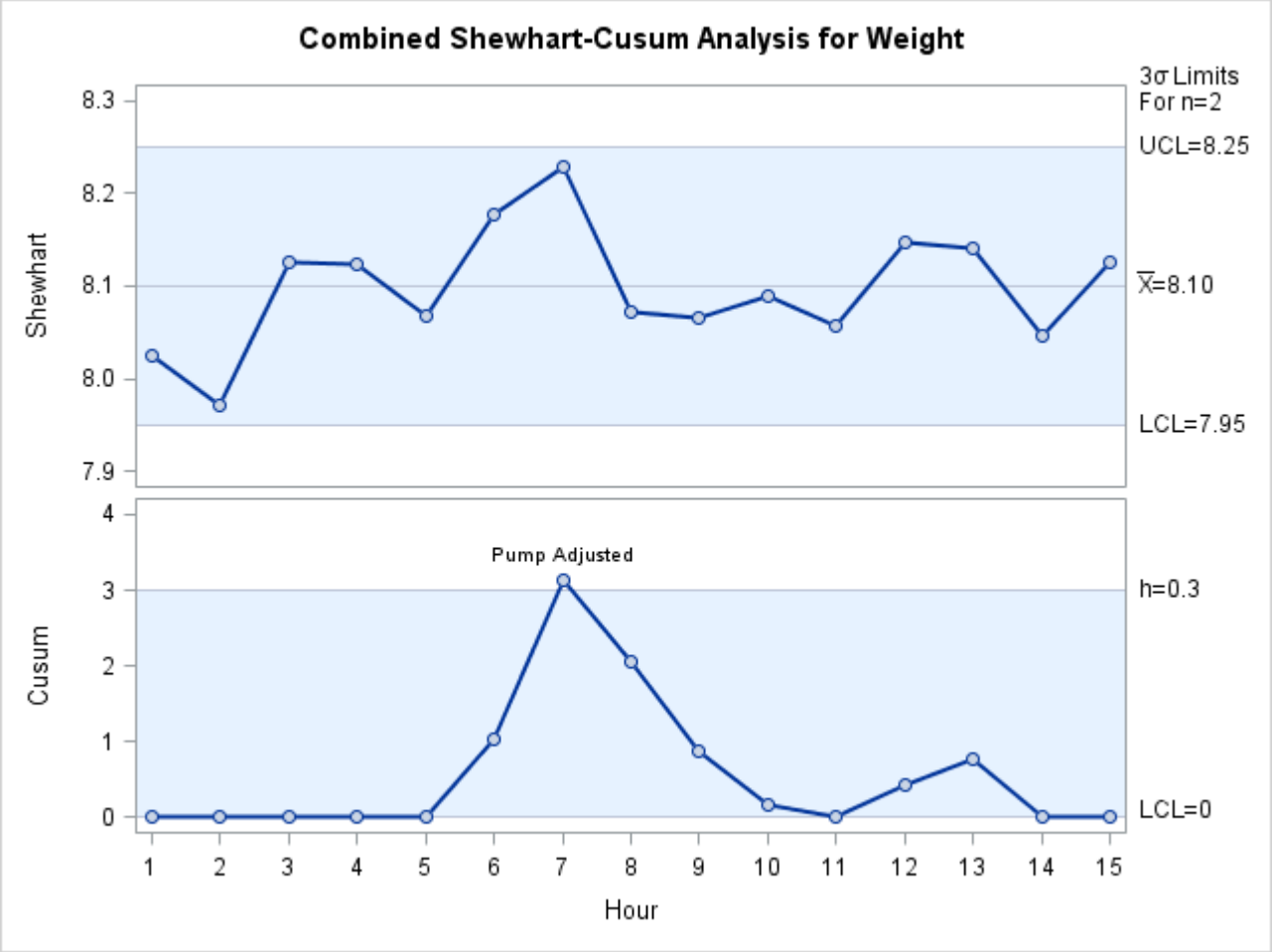
The data set Combine has the structure required for a TABLE= data set used with the IRCHART statement in the SHEWHART procedure (see the section "TABLE= Data Set" on page 1549).

Finally, the combined scheme is displayed with the SHEWHART procedure.

```
ods graphics on;
title "Combined Shewhart-Cusum Analysis for Weight";
proc shewhart table=Combine;
   irchart Weight*Hour /
      odstitle  = title
      ypct1     = 50
      noctl2
      markers
      ucllabel2 = 'h=0.3'
      outlabel  = ( comment )
      outlabel2 = ( comment )
      split     = '/';
   label _subi_ = 'Shewhart/Cusum';
run;
```

The chart is shown in Output 7.3.1.

Output 7.3.1 Combined Shewhart–Cusum Scheme



Note that a shift is detected by the cusum scheme but not by the Shewhart chart. The point exceeding the decision interval is labeled with the variable comment created in the data set Cans.

Lucas and Crosier (1982) tabulates average run lengths for combined Shewhart-cusum schemes. The scheme used here has an ARL of 111.1 for $\delta = 0$ and an ARL of 6.322 for $\delta = 1$.

INSET Statement: CUSUM Procedure

Overview: INSET Statement

The INSET statement enables you to enhance a cusum chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. A possible application of an inset is to present cusum parameters on the chart rather than displaying them in a legend. An inset can also display arbitrary values provided in a SAS data set.

Note that the INSET statement by itself does not produce a display but must be used in conjunction with an XCHART statement. Insets are not available with line printer charts, so the INSET statement is not applicable when the LINEPRINTER option is specified in the PROC CUSUM statement.

You can use options in the INSET statement to do the following:

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started: INSET Statement

This section introduces the INSET statement with a basic example showing how it is used. See the section "INSET and INSET2 Statements: SHEWHART Procedure" on page 1975 in Chapter 19, "The SHEWHART Procedure," for a complete description of the INSET statement.

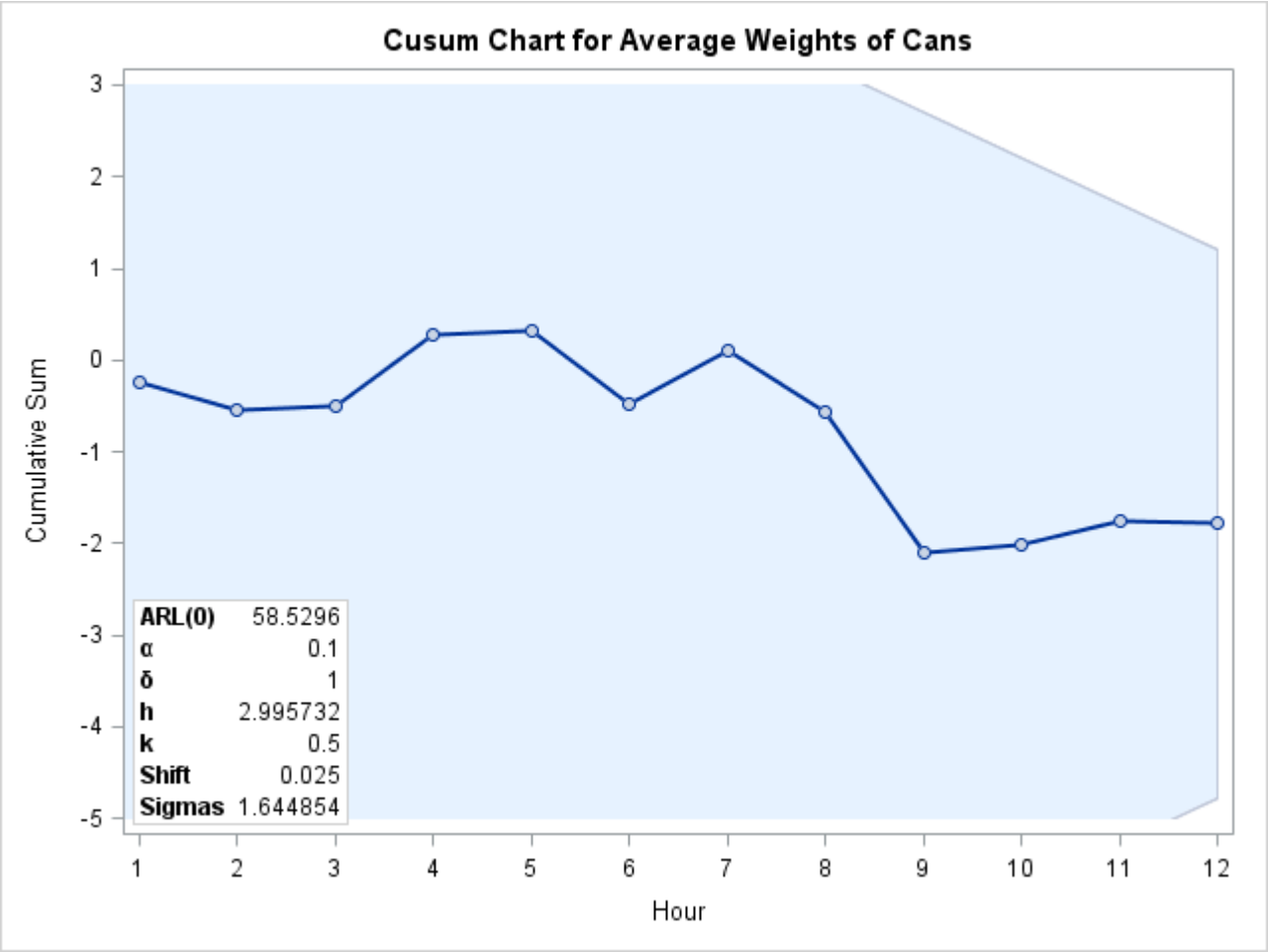
This example is based on the same scenario as the first example in the "Getting Started" subsection of "XCHART Statement: CUSUM Procedure" on page 554. A machine fills cans with oil additive and a two-sided cusum chart is used to detect shifts from the target mean of 8.100 ounces. The following statements create the data set Oil and request a two-sided cusum chart with an inset:

```
data Oil;
   label Hour = 'Hour';
   input Hour @;
   do i=1 to 4;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
 1  8.024  8.135  8.151  8.065
 2  7.971  8.165  8.077  8.157
 3  8.125  8.031  8.198  8.050
 4  8.123  8.107  8.154  8.095
 5  8.068  8.093  8.116  8.128
 6  8.177  8.011  8.102  8.030
 7  8.129  8.060  8.125  8.144
 8  8.072  8.010  8.097  8.153
 9  8.066  8.067  8.055  8.059
10  8.089  8.064  8.170  8.086
11  8.058  8.098  8.114  8.156
12  8.147  8.116  8.116  8.018
;



title 'Cusum Chart for Average Weights of Cans';
ods graphics on;
proc cusum data=Oil;
   xchart Weight*Hour /
      mu0      = 8.100          /* Target mean for process  */
      sigma0   = 0.050          /* Known standard deviation */
      delta    = 1              /* Shift to be detected     */
      alpha    = 0.10           /* Type I error probability */
      vaxis    = -5 to 3
      odstitle = title
      markers
      nolegend;
   label Weight = 'Cumulative Sum';
   inset arl0 ualpha udelta h k shift sigmas / pos = sw;
run;
```

The ODS GRAPHICS ON statement specified before the PROC CUSUM statement enables ODS Graphics, so the cusum chart is created using ODS Graphics instead of traditional graphics.

The resulting cusum chart is shown in Figure 7.3.2.

Output 7.3.2 Two-Sided Cusum Chart with an Inset



Syntax: INSET Statement

The syntax for the INSET statement is as follows:

INSET *keyword-list* < / *options* > **;**

You can use any number of INSET statements in the CUSUM procedure. However, when ODS Graphics is enabled, at most two insets are displayed inside the plot area and at most two are displayed in the chart margins. Each INSET statement produces a separate inset and must follow an XCHART statement. The inset appears on every panel (page) produced by the last XCHART statement preceding it.

Keywords specify the statistics to be displayed in an inset; options control the inset's location and appearance. A complete description of the INSET statement syntax is given in the section "Syntax: INSET and INSET2 Statements" on page 1981 of Chapter 19, "The SHEWHART Procedure." The INSET statement options are identical in the CUSUM and SHEWHART procedures, but the available keywords are different. The options are listed in Table 19.89. The keywords available with the CUSUM procedure are listed in Table 7.11 to Table 7.14.

Table 7.11 Summary Statistics

Keyword	Description
ARL0	average run length for zero shift
ARLDELTA	average run length for shift of δ
DATA=	arbitrary values from *SAS-data-set*
N	nominal subgroup size
NMIN	minimum subgroup size
NMAX	maximum subgroup size

Table 7.12 Parameters for One-Sided (Decision Interval) Cusum Scheme

Keyword	Description
DELTA	shift to be detected as multiple of standard error
H	decision interval h as a multiple of standard error
HEADSTART	headstart value S_0 as a multiple of standard error
K	reference value k
MU0	target mean μ_0
SHIFT	shift to be detected in data units
STDDEV	estimated or specified process standard deviation

Table 7.13 Parameters for Two-Sided (V-Mask) Cusum Scheme

Keyword	Description
ALPHA	probability of Type 1 error
BETA	probability of Type 2 error
H	vertical distance between V-mask origin and upper (or lower) arm
K	slope of lower arm of V-mask
SIGMAS	probability of Type 1 error as probability that standard normally distributed variable exceeds a specified value in absolute value

You can use the keywords in Table 7.14 only when producing ODS Graphics output. The labels for the statistics use Greek letters.

Table 7.14 Keywords Specific to ODS Graphics Output

Keyword	Description
UALPHA	probability of Type 1 error
UARLDELTA	average run length for shift of δ
UBETA	probability of Type 2 error
UDELTA	shift to be detected as multiple of standard error
UMU0	target mean μ_0
USIGMA	estimated or specified process standard deviation

References

American Society for Quality Control (1983). *ASQC Glossary and Tables for Statistical Quality Control*. Milwaukee: ASQC.

Burr, I. W. (1969). "Control Charts for Measurements with Varying Sample Sizes." *Journal of Quality Technology* 1:163–167.

Burr, I. W. (1976). *Statistical Quality Control Methods*. New York: Marcel Dekker.

Chiu, W. K. (1974). "The Economic Design of Cusum Charts for Controlling Normal Means." *Journal of the Royal Statistical Society, Series C* 23:420–433.

Duncan, A. J. (1974). *Quality Control and Industrial Statistics*. 4th ed. Homewood, IL: Irwin.

Goel, A. L. (1982). "Cumulative Sum Control Charts." In *Encyclopedia of Statistical Sciences*, vol. 2, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons.

Goel, A. L., and Wu, S. M. (1971). "Determination of A.R.L. and a Contour Nomogram for Cusum Charts to Control Normal Mean." *Technometrics* 13:221–230.

Ho, C., and Case, K. E. (1994). "Economic Design of Control Charts: A Literature Review for 1981–1991." *Journal of Quality Technology* 26:39–53.

Johnson, N. L. (1961). "A Simple Theoretical Approach to Cumulative Sum Control Chart." *Journal of the American Statistical Association* 56:835–840.

Johnson, N. L., and Leone, F. C. (1962). "Cumulative Sum Control Charts: Mathematical Principles Applied to Their Construction and Use." *Industrial Quality Control* 18: June, 15–21; July, 29–36; August, 22–28.

Johnson, N. L., and Leone, F. C. (1974). *Statistics and Experimental Design*. 2nd ed. Vol. 1. New York: John Wiley & Sons.

Kemp, K. W. (1961). "The Average Run Length of the Cumulative Sum Control Chart When a 'V' Mask Is Used." *Journal of the Royal Statistical Society, Series B* 23:149–153.

Lucas, J. M. (1976). "The Design and Use of V-Mask Control Schemes." *Journal of Quality Technology* 8:1–12.

Lucas, J. M., and Crosier, R. B. (1982). "Fast Initial Response for CUSUM Quality Control Schemes: Give Your CUSUM a Head Start." *Technometrics* 24:199–205.

Montgomery, D. C. (1980). "The Economic Design of Control Charts: A Review and Literature." *Journal of Quality Technology* 12:75–87.

Montgomery, D. C. (1996). *Introduction to Statistical Quality Control*. 3rd ed. New York: John Wiley & Sons.

Nelson, L. S. (1984). "The Shewhart Control Chart—Tests for Special Causes." *Journal of Quality Technology* 15:237–239.

Nelson, L. S. (1985). "Interpreting Shewhart $\bar{X}$ Control Charts." *Journal of Quality Technology* 17:114–116.

Ryan, T. P. (1989). *Statistical Methods for Quality Improvement*. New York: John Wiley & Sons.

Svoboda, L. (1991). "Economic Design of Control Charts: A Review and Literature Survey (1979–1989)." In *Statistical Process Control in Manufacturing*, edited by J. B. Keats and D. C. Montgomery, 311–330. New York: Marcel Dekker.

Van Dobben de Bruyn, C. S. (1968). *Cumulative Sum Tests: Theory and Practice, Griffin's Statistical Monographs and Courses, No. 24*. New York: Hafner Publishing.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986). *Modern Methods for Quality Control and Improvement*. New York: John Wiley & Sons.

Wetherill, G. B. (1977). *Sampling Inspection and Quality Control*. 2nd ed. New York: Chapman & Hall.

Chapter 8
The FACTEX Procedure

Contents

Overview: FACTEX Procedure **618**
Features 619
Getting Started: FACTEX procedure **620**
Example of a Two-Level Full Factorial Design 620
Example of a Full Factorial Design in Two Blocks 622
Example of a Half-Fraction Factorial Design 624
Using the FACTEX Procedure Interactively 626
Syntax: FACTEX Procedure **627**
Summary of Functions 627
PROC FACTEX Statement 629
BLOCKS Statement 630
EXAMINE Statement 632
FACTORS Statement 634
MODEL Statement 634
OUTPUT Statement 636
SIZE Statement 639
UNITEFFECT Statement 640
Details: FACTEX Procedure **641**
Theory of Orthogonal Designs 641
Overview 641
Structure of General Factorial Designs 641
Suitable Confounding Rules 642
Searching for Confounding Rules 644
Speeding Up the Search 645
General Recommendations 646
Design Details 646
Types of Factors 646
Specifying Effects in the MODEL Statement 647
Factor Variable Characteristics in the Output Data Set 648
Statistical Details 648
Resolution 648
Randomization 649
Replication 650
Confounding Rules 653
Alias Structure 653
Minimum Aberration 654

MaxClear Designs . . . 655
Split-Plot Designs . . . 655
Summary of Designs . . . 656
Output . . . 658
ODS Tables . . . 659
Examples: FACTEX Procedure . . . **659**
Example 8.1: Completely Randomized Design . . . 659
Example 8.2: Resolution 4 Augmented Design . . . 660
Example 8.3: Factorial Design with Center Points . . . 663
Example 8.4: Fold-Over Design . . . 664
Example 8.5: Randomized Complete Block Design . . . 666
Example 8.6: Two-Level Design with Design Replication and Point Replication . . . 667
Example 8.7: Mixed-Level Design Using Design Replication and Point Replication . . . 670
Example 8.8: Mixed-Level Design Using Pseudofactors . . . 672
Example 8.9: Mixed-Level Design by Collapsing Factors . . . 673
Example 8.10: Design That Uses a Hyper-Graeco-Latin Square . . . 674
Example 8.11: Resolution 4 Design with Minimum Aberration . . . 676
Example 8.12: Replicated Blocked Design with Partial Confounding . . . 678
Example 8.13: Incomplete Block Design . . . 682
Example 8.14: Design with Inner Array and Outer Array . . . 685
Example 8.15: Fractional Factorial Split-Plot Designs . . . 689
Example 8.16: Design for a Three-Step Process . . . 693
Example 8.17: Strip-Split-Split-Plot Design . . . 697
Example 8.18: Design and Analysis of a Complete Factorial Experiment . . . 699
References . . . **701**

Overview: FACTEX Procedure

The FACTEX procedure constructs orthogonal factorial experimental designs. These designs can be either full or fractional factorial designs, and they can be with or without blocks. You can also construct designs for experiments that have multiple stages, such as split-plot designs(Huang, Chen, and Voelkel 1998) and split-lot designs (Butler 2004). After you have constructed a design by using the FACTEX procedure and run the experiment, you can analyze the results by using a variety of SAS procedures including the GLM and REG procedures.

Factorial experiments are useful for studying the effects of various factors on a response. Texts that discuss experimental design include Box, Hunter, and Hunter (1978), Cochran and Cox (1957), Montgomery (1991), and Wu and Hamada (2000). For more information about the general mathematical theory of orthogonal factorial designs, see Bose (1947).

NOTE: For two-level designs, instead of using PROC FACTEX directly, a more appropriate tool might be the ADX Interface for Design of Experiments. The ADX Interface is designed primarily for engineers and researchers who require a point-and-click solution for the entire experimental process, from building the designs through determining significant effects to optimization and reporting. ADX gives you most

of the two-level designs provided by the FACTEX procedure in a system that integrates construction and analysis of designs, without the need for programming. In addition to two-level designs for standard models (with and without blocking), ADX makes it easy to use PROC FACTEX to construct designs for estimating particular effects of interest. Moreover, ADX also uses the OPTEX procedure to construct two-level designs of nonstandard sizes. For more information, see *Getting Started with the SAS ADX Interface for Design of Experiments*.

Features

There is no inherent limit to the number of factors and the size of the design that you can construct with the FACTEX procedure. Instead of looking up designs in an internal table, the FACTEX procedure uses a general algorithm to search for the construction rules for a specified design.

You can use the FACTEX procedure to generate designs such as the following:

- factorial designs, such as 2^3 designs, with and without blocking
- fractional factorial designs, such as 2^{4-1}_{IV}, with and without blocking
- split-plot and fractional split-plot designs
- three-level designs, with and without blocking
- mixed-level factorial designs, such as 4×3 designs, with and without blocking
- randomized complete block design
- factorial designs with outer arrays
- hyper-Graeco-Latin square designs

You can also create more complex designs, such as incomplete block designs, by using the FACTEX procedure in conjunction with the DATA step.

You can save the design constructed by the FACTEX procedure in a SAS data set. After you have run your experiment, you can add the values of the response variable and use the GLM procedure to perform analysis of variance and to study significance of effects.

The FACTEX procedure is an interactive procedure. After specifying an initial design, you can submit additional statements without reinvoking the procedure. After you have constructed a design, you can do the following:

- print the design points
- examine the alias structure for the design
- modify the design by changing its size, changing the use of blocking, or specifying the effects of interest in the model again
- output the design to a data set

- examine the confounding rules that generate the design
- randomize the design
- replicate the design
- recode the design from standard values (such as ± 1) to values appropriate for your situation
- find another design

Getting Started: FACTEX procedure

Example of a Two-Level Full Factorial Design

NOTE: See *Two-Level Full Factorial Design* in the SAS/QC Sample Library.

This example introduces the basic syntax of the FACTEX procedure.

An experimenter is interested in studying the effects of three factors—cutting speed (Speed), feed rate (FeedRate), and tool angle (Angle)—on the surface finish of a metallic part and decides to run a complete factorial experiment that has two levels for each factor as follows:

Factor	Low Level	High Level
Cutting speed	300	500
Feed rate	20	30
Tool angle	6	8

This is a 2^3 factorial design—in other words, a complete factorial experiment that has three factors, each at two levels. Hence the experiment has eight runs. Because complete factorial designs have full resolution, all the main effects and interaction terms can be estimated. For a definition of the design resolution, see the section "Resolution" on page 648.

The following statements create the required design:

```
proc factex;
   factors Speed FeedRate Angle;
   examine design;
run;
```

These statements invoke the FACTEX procedure, list factor names, and display the generated design points. By default, the FACTEX procedure assumes that the size of the design is a full factorial and that each factor has only two levels.

After you submit the preceding statements, you see the following messages in the SAS log:

```
NOTE: No design size specified.
      Default is a full replicate in 8 runs.
NOTE: Design has 8 runs, full resolution.
```

The output is shown in Figure 8.1. The two factor levels are represented by the coded values –1 and +1.

Figure 8.1 2^3 Factorial Design

The FACTEX Procedure

Design Points			
Experiment Number	Speed	FeedRate	Angle
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	1	-1	-1
6	1	-1	1
7	1	1	-1
8	1	1	1

If you prefer to work with the actual (decoded) values of the factors, you can specify these values in an OUTPUT OUT= statement, as follows:

```
proc factex;
   factors Speed FeedRate Angle;
   output out=SavedDesign
          Speed    nvals=(300 500)
          FeedRate nvals=(20 30)
          Angle    nvals=(6 8);
run;
proc print;
run;
```

The OUTPUT statement in PROC FACTEX recodes the factor levels and saves the constructed design in the SavedDesign data set. Because the levels in this example are of numeric type, you use the NVALS= option to list the factor levels. Optionally, you can use the CVALS= option for levels of character type (see the section "Example of a Full Factorial Design in Two Blocks" on page 622). The design is saved in a user-specified output data set (SavedDesign), as verified by the following message in the SAS log:

```
NOTE: The data set WORK.SAVEDDESIGN has 8 observations
      and 3 variables.
```

Figure 8.2 shows a listing of the data set SavedDesign.

Figure 8.2 2^3 Factorial Design after Decoding

Obs	Speed	FeedRate	Angle
1	300	20	6
2	300	20	8
3	300	30	6
4	300	30	8
5	500	20	6
6	500	20	8
7	500	30	6
8	500	30	8

Although small complete factorial designs are not difficult to create manually, you can easily extend this example to construct a design that has many factors.

Example of a Full Factorial Design in Two Blocks

NOTE: See *Full Factorial Design in Two Blocks* in the SAS/QC Sample Library.

The previous example illustrates a complete factorial experiment that involves eight runs and three factors: cutting speed (Speed), feed rate (FeedRate), and tool angle (Angle).

Now, suppose two machines (A and B) are used to complete the experiment, with four runs being performed on each machine. Because the particular machine might affect the part finish, you should consider machine as a block factor and account for the block effect in assigning the runs to machines.

The following statements construct a blocked design:

```
proc factex;
   factors Speed FeedRate Angle;
   blocks nblocks=2;
   model resolution=max;
   examine design;
run;
```

The FACTORS statement in PROC FACTEX specifies three factors of a 2^3 factorial. The BLOCKS statement specifies that the number of blocks is 2. The RESOLUTION=MAX option in the MODEL statement specifies a design with the highest resolution—that is, the best design in a general sense. Optionally, if you know the resolution of the design, you can replace RESOLUTION=MAX with RESOLUTION=*r*, where *r* is the resolution number. For information about resolution, see the section "Resolution" on page 648.

By default, the FACTEX procedure assumes that the size of the design is a full factorial and that each factor has two levels.

After you submit the preceding statements, you see the following messages in the SAS log:

```
NOTE: No design size specified.
      Default is a full replicate in 8 runs.
NOTE: Design has 8 runs in 2 blocks of size 4,
      resolution = 6.
```

The output is shown in Figure 8.3. By default, the name for the block variable is BLOCK, its levels are 1 and 2, and the default factor levels for a two-level design are –1 and 1.

Figure 8.3 2^3 Factorial Design in Two Blocks before Decoding

The FACTEX Procedure

Design Points				
Experiment Number	Speed	FeedRate	Angle	Block
1	-1	-1	-1	1
2	-1	-1	1	2
3	-1	1	-1	2
4	-1	1	1	1
5	1	-1	-1	2
6	1	-1	1	1
7	1	1	-1	1
8	1	1	1	2

You can rename the block variable and use actual levels for the block variable that is appropriate for your situation as follows:

```
proc factex;
   factors Speed FeedRate Angle;
   blocks nblocks=2;
   model resolution=max;
   output out=BlockDesign
     Speed    nvals=(300 500)
     FeedRate nvals=(20 30)
     Angle    nvals=(6 8)
     blockname=Machine cvals=('A' 'B');
run;

proc print;
run;
```

Figure 8.4 shows the listing of the design that is saved in the data set BlockDesign.

Figure 8.4 2^3 Factorial Design in Two Blocks after Decoding

Obs	Machine	Speed	FeedRate	Angle
1	A	300	20	6
2	A	300	30	8
3	A	500	20	8
4	A	500	30	6
5	B	300	20	8
6	B	300	30	6
7	B	500	20	6
8	B	500	30	8

Example of a Half-Fraction Factorial Design

NOTE: See *Half-Fraction Factorial Design* in the SAS/QC Sample Library.

Often you do not have the resources for a full factorial design. In this case, a fractional factorial design is a reasonable alternative, provided that the effects of interest can be estimated.

Box, Hunter, and Hunter (1978) describe a fractional factorial design for studying a chemical reaction to determine what percentage of the chemicals responded in a reactor. The researchers identified the following five treatment factors that were thought to influence the percentage of reactant:

- the feed rate of the chemicals (FeedRate), ranging from 10 to 15 liters per minute
- the percentage of the catalyst (Catalyst), ranging from 1% to 2%
- the agitation rate of the reactor (AgitRate), ranging from 100 to 120 revolutions per minute
- the temperature (Temperature), ranging from 140 to 180 degrees Celsius
- the concentration (Concentration), ranging from 3% to 6%

The complete 2^5 factorial design requires 32 runs, but the experimenter can afford only 16 runs.

Suppose that all main effects and two-factor interactions are to be estimated. An appropriate design for this situation is a design of resolution 5 (denoted as 2^{5-1}_{V}), in which no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction but in which two-factor interactions are aliased with three-factor interactions. This design loses the ability to estimate interactions between three or more factors, but this is usually not a serious loss. For more information about resolution, see the section "Resolution" on page 648.

You can use the following statements to construct a 16-run factorial design that has five factors and resolution 5:

```
proc factex;
   factors FeedRate Catalyst AgitRate Temperature Concentration;
   size design=16;
   model resolution=5;
   output out=Reaction FeedRate      nvals=(10 15)
                       Catalyst      nvals=(1 2)
                       AgitRate      nvals=(100 120)
                       Temperature   nvals=(140 180)
                       Concentration nvals=(3 6);
run;
proc print;
run;
```

The design is saved in the Reaction data set and shown in Figure 8.5.

Figure 8.5 Half-Fraction of a 2^5 Design for Reactors

Obs	FeedRate	Catalyst	AgitRate	Temperature	Concentration
1	10	1	100	140	6
2	10	1	100	180	3
3	10	1	120	140	3
4	10	1	120	180	6
5	10	2	100	140	3
6	10	2	100	180	6
7	10	2	120	140	6
8	10	2	120	180	3
9	15	1	100	140	3
10	15	1	100	180	6
11	15	1	120	140	6
12	15	1	120	180	3
13	15	2	100	140	6
14	15	2	100	180	3
15	15	2	120	140	3
16	15	2	120	180	6

The use of a half-fraction causes some interaction terms to be confounded with each other. You can use the ALIASING option in the EXAMINE statement to determine which interaction terms are aliased, as follows:

```
proc factex;
   factors FeedRate Catalyst AgitRate Temperature Concentration;
   size design=16;
   model resolution=5;
   examine aliasing;
run;
```

The alias structure summarizes the estimability of all main effects and two- and three-factor interactions. Figure 8.6 indicates that each of the three-factor interactions is confounded with a two-factor interaction. Thus, if a particular three-factor interaction is believed to be significant, the aliased two-factor interaction cannot be estimated with this half-fraction design.

Figure 8.6 Alias Structure of Reactor Design

The FACTEX Procedure

Aliasing Structure
FeedRate
Catalyst
AgitRate
Temperature
Concentration
FeedRate*Catalyst = AgitRate*Temperature*Concentration
FeedRate*AgitRate = Catalyst*Temperature*Concentration
FeedRate*Temperature = Catalyst*AgitRate*Concentration
FeedRate*Concentration = Catalyst*AgitRate*Temperature
Catalyst*AgitRate = FeedRate*Temperature*Concentration
Catalyst*Temperature = FeedRate*AgitRate*Concentration
Catalyst*Concentration = FeedRate*AgitRate*Temperature
AgitRate*Temperature = FeedRate*Catalyst*Concentration
AgitRate*Concentration = FeedRate*Catalyst*Temperature
Temperature*Concentration = FeedRate*Catalyst*AgitRate

When you submit the preceding statements, the following message is displayed in the SAS log:

```
NOTE: Design has 16 runs, resolution = 5.
```

This message confirms that the design exists. If you specify a factorial design that does *not* exist, an error message is displayed in the SAS log. For example, suppose that you replaced the MODEL statement in the preceding example with the following statement:

```
model resolution=6;
```

Since the maximum resolution of a 2^{5-1} design is 5, the following message appears in the SAS log:

```
ERROR: No such design exists.
```

In general, it is good practice to check the SAS log to see if a design exists.

Using the FACTEX Procedure Interactively

By using the FACTEX procedure interactively, you can quickly explore many design possibilities. The following steps provide one strategy for interactive use:

1. Invoke the procedure by using the PROC FACTEX statement, and use a FACTORS statement to identify factors in the design.
2. For a design that involves blocking, use the BLOCKS and MODEL statements. You might want to use the optimization features for the BLOCKS statement.
3. For a fractional replicate of a design, use the SIZE and MODEL statements to specify the characteristics of the design. If the design involves blocking, use a BLOCKS statement too. If you are unsure of the size of the design or of the number of blocks, use the optimization features for either the BLOCKS statement or the SIZE statement.

4 Enter a RUN statement and check the SAS log to see if the design exists. If a design exists, go on to the next step; otherwise, modify the characteristics that are specified in the SIZE, BLOCKS, and MODEL statements.

5 Examine the alias structure of the design. If it is not appropriate for your situation, go back to step 2 and search for another design.

6 After you have repeated steps 2, 3, and 4 and found an acceptable design, use the OUTPUT statement to save the design. You can optionally recode factor values, recode and rename the block factor, and create new factors by using output-value settings.

Syntax: FACTEX Procedure

The following statements are available in the FACTEX procedure. Items within angle brackets (<>) are optional.

PROC FACTEX < *options* > **;**
FACTORS *factor-names* < / *option* > **;**
SIZE *size-specification* **;**
MODEL *model-specification* < / < **MINABS** < (*d*) > > < **MAXCLEAR** < (*d*) > > > **;**
BLOCKS *block-specification* **;**
UNITEFFECT *unit-effect* / < **WHOLE=(***whole-unit-effects***)** > < **SUB=(***subunit-effects***)** > **;**
EXAMINE < *options* > **;**
OUTPUT OUT=*SAS-data-set* < *options* > **;**

To generate a design and save it in a data set, you use at least the PROC FACTEX, FACTORS, and OUTPUT statements. The FACTORS statement should immediately follow the PROC FACTEX statement. You use the MODEL and SIZE statements for designs that are less than a full replicate (for example, fractional factorial designs). You can use the BLOCKS statement for designs that involve blocking. The EXAMINE statement can be used as needed.

The following sections summarize which statements and options you use for various functions, describe the PROC FACTEX statement, and then describe the other statements in alphabetical order.

Summary of Functions

Table 8.1 to Table 8.4 classify the statements and options in PROC FACTEX by function.

Table 8.1 Summary of Options for Specifying the Design

Function	Statement	Option
Factor Specification		
Factor names	FACTORS	$factor_1 \ldots factor_f$
Number of levels	FACTORS	$factor_1 \ldots factor_f$ / NLEV=q

Table 8.1 *continued*

Function	Statement	Option
Design Size Specification (one of the following)		
Number of runs	SIZE	DESIGN=*n*
Fraction of one full replicate	SIZE	FRACTION=*h*
Number of run-indexing factors	SIZE	NRUNFACS=*m*
Minimum number of runs	SIZE	DESIGN=MINIMUM or FRACTION=MAXIMUM or NRUNFACS=MINIMUM
Block Specification (one of the following)		
Number of blocks	BLOCKS	NBLOCKS=*b*
Block size	BLOCKS	SIZE=*k*
Number of block pseudofactors	BLOCKS	NBLKFACS=*s*
Minimum block size	BLOCKS	NBLOCKS=MAXIMUM or SIZE=MINIMUM or NBLKFACS=MAXIMUM
Model Specification (one of the following)		
Estimated effects	MODEL	ESTIMATE=(*effects*)
Estimated effects and nonnegligible effects	MODEL	ESTIMATE=(*effects*) NONNEG=(*nonnegligible-effects*)
Design resolution number	MODEL	RESOLUTION=*r*
Design with highest resolution	MODEL	RESOLUTION=MAXIMUM
Minimum aberration design (up to *d*th-order interactions)	MODEL	EST=(. . .) <NONNEG=(. . .)> or RES=. . . / MINABS<(*d*)>

Table 8.2 Summary of Options for Searching the Design

Function	Statement	Option
Search for the Design		
Allow maximum time of *t* seconds	PROC FACTEX	SECONDS=*t* or TIME=*t*
Limit the design searches	PROC FACTEX	NOCHECK

Table 8.3 Summary of Options for Replicating and Randomizing the Design

Function	Statement	Option
Replication		
Replicate entire design *c* times	OUTPUT OUT=*SAS-data-set*	DESIGNREP=*c*
Replicate design for each point in the data set	OUTPUT OUT=*SAS-data-set*	DESIGNREP=*SAS-data-set*

Table 8.3 *continued*

Function	Statement	Option
Replicate each point in design *p* times	OUTPUT OUT=*SAS-data-set*	POINTREP=*p*
Replicate data set for each point in the design	OUTPUT OUT=*SAS-data-set*	POINTREP=*SAS-data-set*
Randomization		
Randomize the design	OUTPUT OUT=*SAS-data-set*	RANDOMIZE
Randomize the design but not the assignment of factor levels	OUTPUT OUT=*SAS-data-set*	RANDOMIZE NOVALRAN
Specify the seed number	OUTPUT OUT=*SAS-data-set*	RANDOMIZE (u)

Table 8.4 Summary of Options for Examining and Saving the Design

Function	Statement	Option
List the Design		
Coded factor and block levels	EXAMINE	DESIGN
List the Design Characteristics		
Alias structure (up to *d*th-order interactions)	EXAMINE	ALIASING<(*d*)>
Confounding rules	EXAMINE	CONFOUNDING
Save the Design		
Coded factor levels	OUTPUT OUT=*SAS-data-set*	
Decoded factor levels (numeric type)	OUTPUT OUT=*SAS-data-set*	*factor-name* NVALS=(*level1* ... *levelq*)
Decoded factor levels (character type)	OUTPUT OUT=*SAS-data-set*	*factor-name* CVALS=('*level1*' ... '*levelq*')
Block variable name	OUTPUT OUT=*SAS-data-set*	BLOCKNAME=*block-name*
Decoded block levels (numeric type)	OUTPUT OUT=*SAS-data-set*	BLOCKNAME=*block-name* NVALS=(*level1* ... *levelb*)
Decoded block levels (character type)	OUTPUT OUT=*SAS-data-set*	BLOCKNAME=*block-name* CVALS=('*level1*' ... '*levelq*')

PROC FACTEX Statement

PROC FACTEX < *options* > **;**

The PROC FACTEX statement invokes the FACTEX procedure. You can specify the following *options*:

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. By default, NAMELEN=20.

NOCHECK
suppresses a technique for limiting the amount of search required to find a design. The technique dramatically reduces the search time by pruning branches of the search tree that are unlikely to contain the specified design, but in rare cases it can keep the FACTEX procedure from finding a design that does in fact exist. The NOCHECK option turns off this technique at the potential cost of an increase in run time. (However, the run time is always bounded by the TIME= option or its default value.) For more information about the NOCHECK option, see the section "Speeding Up the Search" on page 645.

TIME=*t*
SECONDS=*t*
specifies the maximum number of seconds to spend on the search. By default, TIME=60.

BLOCKS Statement

BLOCKS *block-specification* **;**

The BLOCKS statement specifies the blocks or split-plot units in the design. (By default, the FACTEX procedure constructs designs that do not contain blocks.) If you use the BLOCKS statement, you also need to use the MODEL statement or SIZE statement. In particular, if you use the BLOCKS statement and your design is a fractional factorial design, you must use the MODEL statement.

You can specify one, and only one, of the following *block-specifications* (the simplest explicit *block-specifications* are NBLOCKS=*b* to specify the number of blocks in the design and SIZE=*k* to specify the number of runs in each block):

NBLKFACS=*s*
specifies the number of block pseudofactors for the design. The design contains a different block for each possible combination of the levels of the block pseudofactors. Values of *s* are the integers 1, 2, and so on. For more information, see the section "Block Size Restrictions" on page 632.

If each factor in the design has q levels, then NBLKFACS=*s* specifies a design with q^s blocks. The size of each block depends on the number of runs in the design, as specified in the SIZE statement. If the design has n runs, then each block has n/q^s runs.

The following statement requests a two-level factorial design arranged in eight (2^3) blocks:

```
blocks nblkfacs=3;
```

For more information about pseudofactors, see the section "Types of Factors" on page 646.

NBLOCKS=*b*
specifies the number of blocks in the design. The values of *b* must be a power of q, the number of levels of each factor in the design. For more information, see the section "Block Size Restrictions" on page 632. The size of each block depends on the number of runs in the design, as specified in the SIZE statement. If the design has n runs, then each block has n/b runs. For an illustration of this option, see the section "Example of a Full Factorial Design in Two Blocks" on page 622.

The following statement specifies a design arranged in four blocks:

```
blocks nblocks=4;
```

SIZE=*k*

specifies the number of runs (*k*) per block in the design. The value *k* must be a power of *q*, the number of levels for each factor in the design. The number of blocks depends on the number of runs in the design, as specified in the SIZE statement. If the design has *n* runs, then it has n/k blocks.

NOTE: Do not confuse the SIZE= option in the BLOCKS statement with the SIZE statement, which you use to specify the overall size of the design. For more information about the SIZE statement, see the section "SIZE Statement" on page 639.

The following statement specifies blocks of size two:

```
blocks size=2;
```

NBLKFACS=MAXIMUM

NBLOCKS=MAXIMUM

SIZE=MINIMUM

constructs a blocked design that has the minimum number of runs per block, given all the other characteristics of the design. In other words, the block size is optimized. You cannot specify this option if you specify the DESIGN=MINIMUM option (or either of its aliases, FRACTION=MAXIMUM and NRUNFRACS=MINIMUM) in the SIZE statement.

UNITS=(*unit-factor* **=** *number-of-levels* **<** *unit-factor* **=** *number-of-levels* **... >)**

specifies one or more unit factors that index the runs of the experiment, where the *number-of-levels* for each *unit-factor* must be a power of the number of levels specified in the FACTORS statement (2 by default). The product of all the *number-of-levels* must be less than the size of the experiment, as specified in the SIZE statement.

Unit factors are not involved in the model structure of the design. Instead, you use a UNITS= blocks specification in conjunction with one or more UNITEFFECT statements to constrain how the factor levels can change across the runs of the experiment.

The following statement specifies two unit factors:

```
blocks units=(Unit1=4 Unit2=8);
```

For more information about how to use the UNITS= option and the UNITEFFECT statement to construct split-plot designs, see the section "Split-Plot Designs" on page 655.

Equivalent BLOCK Specifications

The three explicit *block-specifications* (NBLKFACS=*s*, NBLOCKS=*b*, and SIZE=*k*)are related to each other, as demonstrated by the following example.

Suppose you want to construct a design for 11 two-level factors in 128 runs in blocks of size 8. Because $128/2^4 = 128/16 = 8$, the three equivalent block specifications are as follows:

```
blocks nblkfacs=4;
blocks nblocks=16;
blocks size=8;
```

Block Size Restrictions

The number of blocks and the number of runs in each block must be less than the total number of runs in the design. Hence, the block size is restricted as follows:

- If you use SIZE=*k* or NBLOCKS=*b*, the numbers you specify for *k* and *b* must be less than or equal to the size of the design, as specified in the SIZE statement. Or, if you do not use a SIZE statement, *k* and *b* must be less than or equal to the number of runs for a full replication of all possible combinations of the factors.

  For example, you cannot specify a design arranged in 8 blocks (NBLOCKS=8) for a 2^3 design. Likewise, you cannot construct a design with block size greater than 8 (SIZE=8).

- If you use NBLKFACS=*s*, the value of *s* can be no greater than the number of run-indexing factors, which give the number of runs needed to index the design. For more information, see the sections "Types of Factors" on page 646 and "Theory of Orthogonal Designs" on page 641.

EXAMINE Statement

EXAMINE < *options* > **;**

The EXAMINE statement specifies the characteristics of the design that are to be listed in the output. The *options* are remembered by the procedure; once specified, they remain in effect until you submit a new EXAMINE statement with different *options* or until you turn off all options by submitting the statement with no *options* as follows:

```
examine;
```

You can specify the following *options*:

ABERRATION

AB

displays the design's aberration vector, which summarizes the confounded interactions. For more information, sSee the section "Minimum Aberration" on page 654.

ALIASING < (< *d* > < UNITS < = ONE | ALL > >) >

A < (< *d* > < UNITS < = ONE | ALL > >) >

displays the design's alias structure, which identifies effects that are confounded with one another and are thus indistinguishable.

You can specify the following suboptions in parentheses:

d

displays the alias structure with effects up to and including order *d*. For example, the following statement requests aliases for up to fourth-order effects (for example, A*B*C*D):

```
examine aliasing(4);
```

Each line of the alias structure is displayed in the following form for as many effects as are aliased with one another:

effect = effect = . . . = effect

The default value for *d* is determined automatically from the model as follows:

- If you use RESOLUTION=*r* in the MODEL statement to specify the model, then *d* is the integer part of $(r + 1)/2$.
- If you use ESTIMATE=*effects* in the MODEL statement specify the model, then *d* is the larger of the following, where main effects have order 1, two-factor interactions have order 2, and so on:
  - one plus the largest order of an effect to be estimated
  - the largest order of an effect considered to be nonnegligible

UNITS

UNITS=ONE

displays the first unit effect with which each treatment effect is aliased. Specifying this suboption can give you information about which error stratum can be used to estimate the background error variance for each estimable treatment effect. This option applies only when *unit-effects* are specified in the UNITEFFECTS statement.

UNITS=ALL

displays all unit effects with which each treatment effect is aliased. This suboption is useful when unit effects are nested, as they typically are in complex split-plot designs, because treatment effects can be aliased with more than one unit effect. This option applies only when *unit-effects* are specified in the UNITEFFECTS statement.

For more information about aliasing, see the section "Alias Structure" on page 653.

CONFOUNDING

C

displays the confounding rules that are used to construct the design. For the definition of confounding rules, see the sections "Confounding Rules" on page 653 and "Suitable Confounding Rules" on page 642.

DESIGN

D

displays the points in the design in standard order with the factor levels coded. For a description of the randomization and coding rules, see the section "OUTPUT Statement" on page 636.

SUMMARY < (< *d* >) >

S < (< *d* >) >

displays the design's modeling summary, which summarizes how many interactions of each order are estimable and how many are clearly estimable (that is, unaliased with any other interactions of interest).

You can specify *d* in parentheses to display a modeling summary that accounts for effects up to and including order *d*. The default value for *d* is determined automatically from the model as it is for the ALIASING option.

FACTORS Statement

FACTORS *factor . . . factor* < / *option* > **;**

The FACTORS statement starts the construction of a new design by naming the factors in the design. The FACTORS statement clears all previous specifications for the design (number of runs, block size, and so on); use it when you want to start a new design.

NOTE: If you want to specify the FACTORS statement, it must be the first statement following the PROC FACTEX statement.

You must specify the following argument:

factor **...** *factor*
names the factors in the design. You must specify at least one *factor*. These names must be valid SAS variable names. For more information, see the section "Types of Factors" on page 646.

You can also specify the following *option*:

NLEV=*q*
specifies the number of levels for each factor in the design. The value of *q* must be an integer greater than or equal to 2. In order to construct a design that involves either fractionation or blocking, *q* must be either a prime number or an integer power of a prime number. For the reason behind this restriction, see the section "Structure of General Factorial Designs" on page 641. By default, NLEV=2.

MODEL Statement

MODEL *model-specification* < / < **MINABS** < (*d*) > > < **MAXCLEAR** < (*d*) > > > **;**

The MODEL statement provides the model for the construction of the factorial design. You can specify the model either directly by specifying the effects to be estimated in the ESTIMATE= option or indirectly by specifying the resolution of the design in the RESOLUTION= option.

NOTE: If you create a fractional factorial design or if you create a design that involves blocking, the MODEL statement is required.

You must specify one, and only one, of the following *model-specifications*:

ESTIMATE=(*effects***)** < *option* >

EST=(*effects***)** < *option* >

E =(*effects***)** < *option* >
identifies the *effects* that you want to estimate with the design. To specify *effects*, simply list the names of main effects, and use asterisks to join terms in interactions. The *effects* must be enclosed within parentheses. For more information, see the section "Specifying Effects in the MODEL Statement" on page 647.

You can specify the following *option*:

NONNEGLIGIBLE=(*nonnegligible-effects***)**

NONNEG =(*nonnegligible-effects***)**

N =(*nonnegligible-effects***)**

identifies nonnegligible effects. These are the effects whose magnitudes are unknown but that you do not necessarily want to estimate with the design and that you do not want to be aliased with the *effects*. The *nonnegligible-effects* must be enclosed within parentheses.

For example, suppose that you want to construct a fraction of a 2^4 design in order to estimate the main effects of the four factors. To specify the model, simply list the main effects in the ESTIMATE= option, since these are the effects of interest. Furthermore, if you consider the two-factor interactions to be significant but you are not interested in estimating them, then list these interactions in the NONNEGLIGIBLE= option.

Example 8.8 uses the ESTIMATE= option. For more information about how the FACTEX procedure interprets the model and derives an appropriate confounding scheme, see the section "Theory of Orthogonal Designs" on page 641.

RESOLUTION=*r* | **MAXIMUM**

RES= *r* | **MAXIMUM**

R= *r* | **MAXIMUM**

specifies the resolution of the design. You can specify one of the following values:

r is a positive integer greater than or equal to 3, which is interpreted as follows:

- If *r* is odd, then the effects of interest are taken to be those of order $(r-1)/2$ or less.
- If *r* is even, then the effects of interest are taken to be those of order $(r-2)/2$ or less, and the nonnegligible effects are taken to be those of order $r/2$ or less.

MAXIMUM searches for a design that has the highest resolution and satisfies the SIZE statement requirements.

For more information about design resolution, see the section "Resolution" on page 648. For an example that uses the RESOLUTION=*r* option, see the section "Example of a Half-Fraction Factorial Design" on page 624. For an example that uses the RESOLUTION=MAX option, see the section "Example of a Full Factorial Design in Two Blocks" on page 622.

You can also specify the following options in the MODEL statement:

MAXCLEAR <(*d***)>**

searches for a design that maximizes the number of clear interactions. Clear interactions are interactions that are not aliased with any other effects that are either required to be estimable or assumed to be nonnegligible. Specifying (*d*) after the MAXCLEAR option requests a search for a maximum-clarity design that involves interactions up to order *d*. The default value for *d* is determined automatically from the model (as it is for the ALIASING option in the EXAMINE statement) as follows:

- If you use RESOLUTION=*r* in the MODEL statement to specify the model, then *d* is the integer part of $(r+1)/2$.
- If you use ESTIMATE=*effects* in the MODEL statement to specify the model, then *d* is the larger of the following, where main effects have order 1, two-factor interactions have order 2, and so on:

- one plus the largest order of an effect to be estimated
- the largest order of an effect considered to be nonnegligible

For more information about MaxClear designs, see the section "MaxClear Designs" on page 655.

MINABS < (*d*) >
searches for a design that has minimum aberration. Specifying (*d*) after the MINABS option requests a search for a minimum aberration design that involves interactions up to order *d*. The default value for *d* is determined automatically from the model as follows:

- If you use RESOLUTION=*r* in the MODEL statement to specify the model, then $d = r + 2$.
- If you use ESTIMATE=*effects* in the MODEL statement to specify the model, then *d* is the larger of the following, where main effects have order 1, two-factor interactions have order 2, and so on:
  - three plus twice the largest order of an effect to be estimated
  - one plus twice the largest order of an effect considered to be nonnegligible

For more information, see the section "Minimum Aberration" on page 654. For an example of the MINABS option, see Example 8.11.

Examples of the MODEL Statement

Suppose you use the following FACTORS statement to specify a design, where the number of factors *f* can be replaced with a number:

```
factors x1-xf;
```

Then Table 8.5 lists equivalent ways to specify common models.

Table 8.5 Equivalent of Model Specifications

RESOLUTION= Option	ESTIMATE= and NONNEGLIGIBILE= Options
`model res=3`	`model est=(x1-x+f);`
`model res=4`	`model est=(x1-x+ f) nonneg=(x1|x2|x3|+...+|x+f+@2);`
`model res=5`	`model est=(x1|x2|x3|+...+|x+ f+@2);`

The RESOLUTION= specification is more concise than the ESTIMATE= specification and is also more efficient in an algorithmic sense. To decrease the time required to find a design, particularly for designs that have a large number of factors, you should specify your model by using the RESOLUTION= option rather than listing the effects in the ESTIMATE= option. For more information about interpreting the resolution number, see the section "Resolution" on page 648.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set* < *options* > **;**

The OUTPUT statement saves a design in an output data set. Optionally, you can use the OUTPUT statement to modify the design by specifying values to be output for factors, creating new factors, randomizing the design, and replicating the design.

You must specify the following argument:

OUT=*SAS-data-set*
names the output data set in which the design is saved.

You can also specify the following *options*:

variable-specification **< NVALS=(***level1 level2 . . . levelq***) >**

variable-specification **< CVALS=(**'*level1*' '*level2*' . . . '*levelq*'*)* **>**
names and optionally recodes the values for design factors, block factors, or derived factors. If you rename and recode a factor, the type and length of the new variable are determined by whether you use the CVALS= option (the new variable is a character variable with length equal to the longest string) or the NVALS= option (the new variable is a numeric variable).

Specify one of the following as the *variable-specification*:

factor-name
names the design factors to be recoded by the CVALS= or NVALS= option.

BLOCKNAME=*block-name*
gives a new name (*block-name*) for the block factor and optionally recodes its values. If the design uses blocking, the output data set automatically contains a block variable named Block, for which the default values are $1, 2, \ldots, b$ for a design that has b blocks. You can rename the block variable and optionally recode the block levels from the default levels to levels that are appropriate for your situation.

For example, for a design arranged in four blocks, suppose that the block variable is the day of the week (Day) and that the four block levels of character type are *Mon*, *Tue*, *Wed*, and *Thu*. You can use the following statement to rename the block variable, recode the block levels, and save the design in a SAS data set named Recode:

```
output out=recode blockname=Day cvals=('Mon''Tue''Wed''Thu');
```

[*design-factors***]=** *derived-factor*
creates derived factors that are based on the joint values of a set of the design factors, where *design-factors* names factors that are currently in the design and *derived-factor* names the new derived factor. The *design-factors* are combined to create the new derived factor. The *derived-factor* must not be used in the design.

Each distinct combination of levels of the design factors corresponds to a single level for the derived factor. Thus, when you create a derived factor from k design factors, each with q levels, the derived factor has q^k levels. Derived factors are useful when you create mixed-level designs; see Example 8.8. For more information about how the levels of design factors are mapped into levels of the derived factor, see the section "Structure of General Factorial Designs" on page 641.

If you create a derived factor but do not use the NVALS= or CVALS= option to assign levels to the derived factor, the FACTEX procedure assigns the values $0, 1, \ldots, q^k - 1$, where the derived factor is created from k design factors, each with q levels. In general, the CVALS= or NVALS= list for a derived factor must contain q^k values.

The following statement is an example of creating a derived factor and then renaming the levels of the factor:

```
output out=new [A1 A2]=A cvals=('A' 'B' 'C' 'D');
```

This statement converts two 2-level factors (A1 and A2) into one 4-level factor (A), which has the levels A, B, C, and D.

You can also specify one of the following options after the *variable-specification*:

NVALS=(*level1 level2 . . . levelq***)**
lists new numeric levels for the design factors and maps *level1* to the lowest level for the factor, *level2* to the next lowest level, and so on.

CVALS=('*level1***' '***level2***' . . . '***levelq***')**
lists new character levels for the design factors and maps '*level1*' to the lowest level for the factor, '*level2*' to the next lowest level, and so on. Each string can be up to 40 characters long. The length of the new variable is equal to the longest string.

By default, the output data set contains a variable for each factor in the design. These variables are coded with standard values, as follows:

- For factors that have two levels ($q = 2$), the values are –1 and +1.
- For factors that have three levels ($q = 3$), the values are –1, 0, and +1.
- For factors with q levels ($q > 3$), the values are $0, 1, 2, \ldots, q - 1$.

You can recode the levels of the factor from the standard levels to levels that are appropriate for your situation. For example, suppose you want to recode a three-level factorial design from the standard levels –1, 0, and +1 to the actual levels. Suppose the factors are pressure (Pressure) with character levels, agitation rate (Rate) with numeric levels, and temperature (Temperature) with numeric levels. You can use the following statement to recode the factor levels and save the design in a SAS data set named Recode:

```
output out=recode Pressure    cvals=('low' 'medium' 'high')
                  Rate        nvals=(20   40  60)
                  Temperature cvals=(100 150 200);
```

For more information about recoding a factor, see the section "Factor Variable Characteristics in the Output Data Set" on page 648.

DESIGNREP=*c* | *SAS-data-set*
replicates the entire design. Specify one of the following values:

c	replicates the design *c* times, where *c* is an integer.
SAS-data-set	replicates the design once for each point in the *SAS-data-set*. The OUT= data set contains the variables in the *SAS-data-set* in addition to the design variables. In mathematical notation, the OUT= data set is the direct product of the *SAS-data-set* and the design. If the design is a and the *SAS-data-set* is b, then the OUT= data set is $b \otimes a$, where $\otimes$ denotes the direct product.

For more information, see the section "Replication" on page 650. For illustrations of the difference between the DESIGNREP= and POINTREP= options, see Example 8.6 and Example 8.7.

POINTREP=*p* | *SAS-data-set*
replicates each point of the design. Specify one of the following values:

p	replicates each design point *p* times, where *p* is an integer.
SAS-data-set	replicates the *SAS-data-set* once for each point in the design. The OUT= data set contains the variables in the *SAS-data-set* in addition to the design variables. In mathematical notation, the OUT= data set is the direct product of the design and the *SAS-data-set*. If the design is a and the *SAS-data-set* is b, then the OUT= data set is $a \otimes b$, where $\otimes$ denotes the direct product.

For more information, see the section "Replication" on page 650. For illustrations of the difference between the POINTREP= and DESIGNREP= options, see Example 8.6 and Example 8.7.

RANDOMIZE < (*u***) > < NOVALRAN >**
randomizes the design. You can specify the following options:

(*u***)**
specifies an integer to use as a seed to start the pseudorandom number generator for randomizing the design. The value of *u* must be enclosed in parentheses and be specified as the first option after the keyword RANDOMIZE. If you do not specify *u* or if you specify a value less than or equal to 0, the seed is generated from reading the time of day from the computer's clock.

NOVALRAN
prevents the randomization of theoretical factor levels to actual levels. The randomization of run order is still performed.

For more information, see the section "Randomization" on page 649.

SIZE Statement

SIZE *size-specification* **;**

The SIZE statement specifies the size of the design, which is the number of runs in the design. The SIZE statement is required for designs of less than a full replicate (for example, fractional factorial designs). By default, the design consists of one full replication of all possible combinations of the factors.

You can specify one, and only one, of the following *size-specifications* (the simplest explicit *size-specifications* are DESIGN=*n* to specify the number of runs (*n*) in the design and FRACTION=*h* to specify $1/h$):

DESIGN=*n*
specifies the actual number of runs in the design. The number of runs must be a power of the number of levels *q* for the factors in the design. (See the NLEV= option.) If the last FACTORS statement does not contain the NLEV= option, then $q = 2$ by default, and as a result, *n* must be a power of 2. For an example, see Example 8.1.

FRACTION=*h*

specifies the fraction of one full replication of all possible combinations of the factors. For example, FRACTION=2 specifies a half-fraction, FRACTION=4 specifies a quarter-fraction, and so on. In general, FRACTION=*h* specifies a design with $1/h$ of the runs in a full replicate. If the design has f factors, each with q levels, then the size of the design is q^f/h. If you use FRACTION=*h*, *h* must be a power of q. See Example 8.4.

NRUNFACS=*m*

specifies the number of run-indexing factors in the design. The design contains one run for each possible combination of the levels of the run-indexing factors. Run-indexing factors are the first *m* factors for a design in q^m runs. All possible combinations of the levels of the run-indexing factors occur in the design. As a result, if each factor has q levels, the number of runs in the design is q^m. For more information about run-indexing factors, see the sections "Types of Factors" on page 646 and "Structure of General Factorial Designs" on page 641.

DESIGN=MINIMUM

FRACTION=MAXIMUM

NRUNFACS=MINIMUM

constructs a design that has the minimum number of runs (no larger than one full replicate) given all of the other characteristics of the design. In other words, the design size is optimized. You cannot specify this option if you specify NBLKFACS=MAXIMUM (or any of its aliases, NBLOCKS=MAXIMUM or SIZE=MINIMUM) in the BLOCKS statement.

The three explicit *size-specifications* (DESIGN=*n*, FRACTION=*h*, and NRUNFRACS=*m*) are related to each other, as demonstrated by the following example. Suppose you want to construct a design for 11 two-level factors in 128 runs. Since $128 = 2^{11}/16 = 2^7$, the three equivalent size specifications for this design are as follows:

```
size design=128;
size fraction=16;
size nrunfacs=7;
```

UNITEFFECT Statement

UNITEFFECT *unit-effect* / < **WHOLE=(***whole-unit-effects***)** > < **SUB=(***subunit-effects***)** > **;**

You use the UNITEFFECT statement to specify constraints on how the factor levels can change across the runs of the experiment. Such constraints are known as randomization restrictions. UNITEFFECT statements are used in conjunction with a UNITS= option in the BLOCKS statement, which defines unit factors that index the runs of the experiment.

You must specify a *unit-effect*, which is an interaction between *unit-factors* that are specified in the UNITS= option in the BLOCKS statement. Specify the *unit-effect* as follows:

unit-factor * . . . * *unit-factor*

The *unit-effect* defines a partition of the runs on which to apply whole-unit and subunit effects of the factors that are named in the FACTORS statement.

In addition, you can specify the following options after a slash (/):

WHOLE=*whole-unit-effects*
typically defines a necessary feature of how the experiment must be designed, and are thus known as "design constraints." You must enclose the *whole-unit-effects* in parentheses.

SUB=*subunit-effects*
indicates which unit mean contrasts will be used to compute the *subunit-effects* and which random error terms will be used to test them. Thus, the *subunit-effects* are known as "model constraints." You must enclose the *subunit-effects* in parentheses.

For more information, see the section "Specifying Effects in the MODEL Statement" on page 647.

Suppose you have specified units in the BLOCKS statement as follows:

```
blocks units=(WholePlot=4);
```

Then the following statement illustrates how to specify unit effects that correspond to these units:

```
uniteffect WholePlot / whole=(x1-x3) sub=(x4-x6);
```

For more information about how to use the UNITS= option and the UNITEFFECT statement to construct split-plot designs, see the section "Split-Plot Designs" on page 655.

Details: FACTEX Procedure

Theory of Orthogonal Designs

Overview

This section provides the mathematical and statistical background for designs that are constructed by the FACTEX procedure; it also outlines the search algorithm that is used to find suitable construction rules. The material in this section is general and theoretical; you do not need to read this section in order to use the procedure for constructing most common experimental designs. On the other hand, you might want to read this section for the following reasons:

- to understand the general structure of designs that can be constructed with the FACTEX procedure
- to construct designs for factors that have more than two levels, especially if interactions are involved
- to improve the search that the procedure uses when it constructs complicated designs that involve many factors

Structure of General Factorial Designs

The FACTEX procedure constructs a fractional design for q-level factors by using the *Galois field* (also called the *finite field*) of size q. This system has q elements and two operations + and $\times$, which satisfy the usual mathematical axioms for addition and multiplication. When q is a prime number, finite field arithmetic is equivalent to regular integer arithmetic modulo q. When $q = 2$, addition of the two elements of the finite field is equivalent to multiplication of the integers +1 and –1. Because designs for factors that have levels +1

and –1 are the factorial designs most commonly covered in textbooks, the arithmetic for fractional factorial designs is usually shown in multiplicative form. However, throughout this section a more general notation is used.

A design for q-level factors in q^m runs constructed by the FACTEX procedure has the following general form: The first m factors are taken to index the runs in the design, with one run for each different combination of the levels of these factors, where the levels run from 0 to $q - 1$. These factors are called *run-indexing factors*. For a particular run, the value F of any other factor in the design is derived from the levels $P_1, P_2, \ldots, P_m$ of the run-indexing factors by means of *confounding rules*. These rules are of the general form

$$F = r_1 P_1 + r_2 P_2 + \ldots + r_m P_m$$

where all the arithmetic is performed in the finite field of size q. The linear combination on the right-hand side of the preceding equation is called a *generalized interaction* between the run-indexing factors. A generalized interaction is part of the statistical interaction between the factors that have nonzero coefficients in the linear combination. The factor F is said to be *confounded* (*aliased*) with this generalized interaction; two terms are confounded when the levels they take in the design yield identical partitions of the runs, so that their effects cannot be distinguished. The confounding rules characterize the design, and the problem of constructing the design reduces to finding suitable confounding rules.

Suitable Confounding Rules

Design Factors

This section explains how the criteria for a design can be reduced to prescribing that certain generalized interactions are *not* to be "confounded with zero."

Suitable confounding rules depend on the effects you want to estimate with the design. For example, if you want to estimate the main effects of both A and B, the following rule is inappropriate:

$$A = B$$

With this rule, the levels of A and B are the same in every run of the design, and the main effects of the two factors cannot be estimated independently of one another. Thus, the first criterion for a suitable confounding rule is that no two effects you want to estimate should be confounded with each other.

Furthermore, an effect you want to estimate should not be confounded with an effect that is nonnegligible. For example, if the interaction between C and D is nonnegligible and you want to estimate the main effect of A, the following confounding rule is inappropriate:

$$A = C + D$$

(Recall that this section uses a general linear form for confounding rules instead of the usual multiplicative form. For factors that have levels +1 and –1, the preceding rule is equivalent to $A = C * D$.)

Another kind of confounding involves *confounding with zero*. If a factor or a generalized interaction F has the same value in every run of the design, then F is *confounded with zero*. Such confounding is denoted as

$$0 = F$$

Interactions can be estimated by the design if and only if they are not confounded with zero. Consequently, another criterion for a suitable confounding rule is that no effect that you want to estimate can be confounded with zero. The confounding rule for two main effects is

$$A = B$$

This rule can be written as a generalized interaction confounded with zero:

$$0 = -A + B$$

The right-hand side of the preceding equation is part of the interaction between A and B. Thus, for any two effects to be unconfounded, it is equivalent to prescribe that no part of their generalized interaction be confounded with zero.

It is not enough to make sure that only the confounding rules themselves satisfy these restrictions. The consequences of the confounding rules must also satisfy the restrictions. For example, suppose you want to make sure that main effects are not confounded with two-factor interactions and suppose that the confounding rule for factor E is

$$E = A + B + C + D$$

Then the following rule cannot be used for factor F:

$$F = A + B + C$$

Even though the rule for F does not confound F with a two-factor interaction, this rule forces a generalized interaction between E and F to be aliased with the main effect of D, because

$$E - F = (A + B + C + D) - (A + B + C) = D$$

Block Factors

If your design involves blocks, additional confounding criteria need to be considered. Blocks are introduced into designs by means of *block pseudofactors*. (For more information, see the section "Types of Factors" on page 646.) A design for q-level factors in q^s blocks contains s block pseudofactors. Denoting the levels of these factors for any particular run by $B_1, B_2, \ldots, B_s$, the index of the block in which the run occurs is determined by

$$B_1 + qB_2 + q^2B_3 + \ldots + q^{s-1}B_s$$

For each block to occur in the design, every possible combination of block pseudofactors must occur. This can happen only if all main effects and interactions between the block factors are estimable, which leads to yet another criterion for the confounding rules. Moreover, the effects you want to estimate cannot be confounded with blocks. In general, the following restrictions exist:

- No generalized block pseudofactors can be confounded with zero.
- No generalized interactions between block pseudofactors and effects you want to estimate can be confounded with zero.

General Criteria

The criteria for an orthogonally confounded q^k design reduce to requiring that no generalized interactions in a certain set $\mathcal{M}$ can be confounded with zero. (For a definition of *generalized interaction*, see the section "Structure of General Factorial Designs" on page 641.) This section presents the general definition of $\mathcal{M}$. First, define the following three sets:

$\mathcal{E}$	the set of effects that you want to estimate
$\mathcal{N}$	the set of effects that you do not want to estimate but that have unknown nonzero magnitudes (referred to as *nonnegligible* effects)
$\mathcal{B}$	the set of all generalized interactions between block pseudofactors

Furthermore, for any two sets of effects $\mathcal{A}$ and $\mathcal{B}$, denote by $\mathcal{A} \times \mathcal{B}$ the set of all generalized interactions between the effects in $\mathcal{A}$ and the effects in $\mathcal{B}$.

Then the general rules for creating the set of effects $\mathcal{M}$ that are not to be confounded with zero are as follows:

- Put $\mathcal{E}$ in $\mathcal{M}$. This ensures that all effects in $\mathcal{E}$ are estimable.
- Put $\mathcal{E} \times \mathcal{E}$ in $\mathcal{M}$. This ensures that all pairs of effects in $\mathcal{E}$ are not confounded with each other.
- Put $\mathcal{E} \times \mathcal{N}$ in $\mathcal{M}$. This ensures that effects in $\mathcal{E}$ are not confounded with effects in $\mathcal{N}$.
- Put $\mathcal{B}$ in $\mathcal{M}$. This ensures that all q^s blocks occur in the design.
- Put $\mathcal{E} \times \mathcal{B}$ in $\mathcal{M}$. This ensures that effects in $\mathcal{E}$ are not confounded with blocks.

Searching for Confounding Rules

The goal in constructing a design, then, is to find confounding rules that do not confound with zero any of the effects in the set $\mathcal{M}$ defined previously. This section describes the sequential search that the FACTEX procedure performs to accomplish this goal.

First, construct the set C_1 of candidates for the first confounding rule, taking into account the set $\mathcal{M}$ of effects not to be confounded with zero. If C_1 is empty, then no design is possible; otherwise, choose one of the candidates $r_1 \in C_1$ for the first confounding rule and construct the set C_2 of candidates for the second confounding rule, taking both $\mathcal{M}$ and r_1 into account. If C_2 is empty, choose another candidate from C_1; otherwise, choose one of the candidates rules $r_2 \in C_2$ and go on to the third rule. The search continues either until it succeeds in finding a rule for every factor that is not a run-indexing factor or until the search fails because the set C_1 is exhausted.

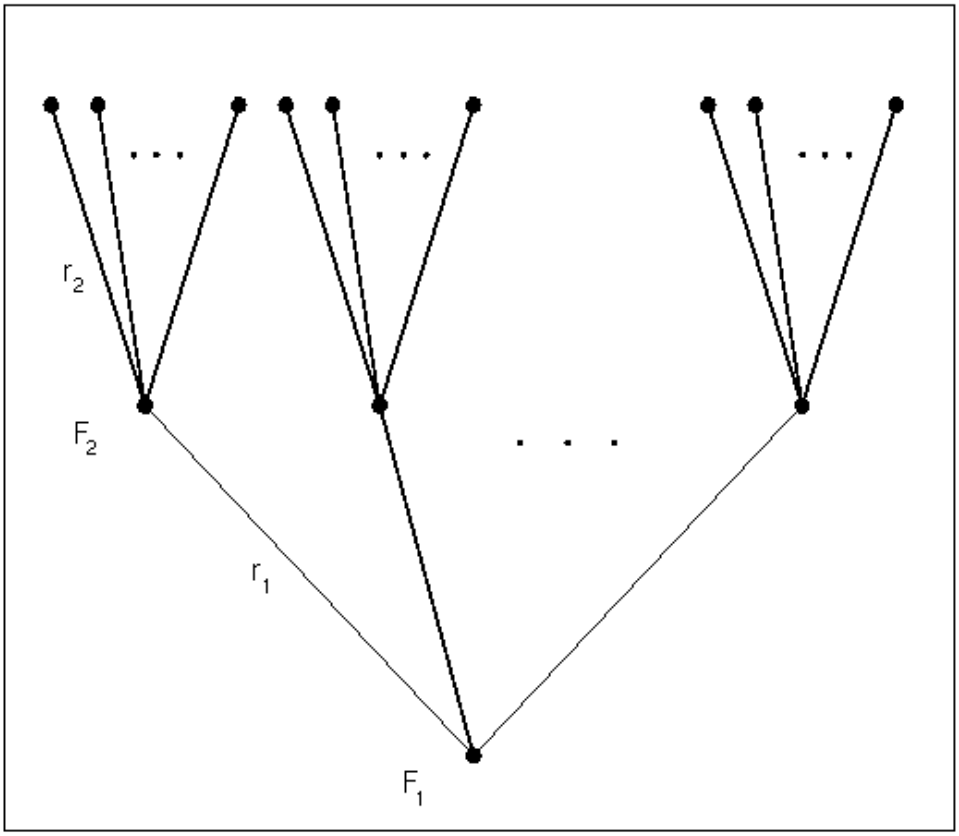
The algorithm used by the FACTEX procedure to select confounding rules is essentially a depth-first tree search. Imagine a tree structure in which the branches connected to the root node correspond to the candidates C_1. Traversing one of these branches corresponds to choosing the corresponding rule r_1 from C_1. The branches attached to the node at the next level correspond to the candidates for the second rule if r_1 is specified. In general, each node at level i of the tree corresponds to a set of feasible choices for rules $r_1, \ldots, r_i$, and the rest of the tree above this node corresponds to the set of all possible feasible choices for the rest of the rules.

Speeding Up the Search

For designs that contain many factors or blocks, the tree of candidate confounding rules can be very large and the search can take a very long time. In these cases, the FACTEX procedure spends a lot of time exploring sets of rules that are essentially the same and that all result in failure. A technique for pruning the search tree (Figure 8.7) is as follows. Suppose that for some selection r_i for rule i, all the branches for the next rule eventually result in failure. Then any other selection r_i' is immediately declared a failure if the resulting number of candidates is the same as for the failed rule r_i. The search goes on to the next selection for rule i.

This method of pruning is not perfect; it might prune a branch of the search tree that would have resulted in a success. In mathematical terms, candidate sets C_i are not necessarily isomorphic because they have the same size. You can use the NOCHECK option in the PROC FACTEX statement to turn off the pruning. When the NOCHECK option is specified, the FACTEX procedure searches the entire tree of feasible confounding rules and will find a design if one exists and given enough time. The default value for the TIME= option in the PROC FACTEX statement limits the search time to one minute.

Figure 8.7 Search Tree



On the other hand, the NOCHECK option is rarely needed to produce a design that has a particular resolution. For example, consider all possible blocked and unblocked two-level designs that have minimum resolution for 20 or fewer factors and 128 or fewer runs. Of the nearly 400 different designs, the NOCHECK option is required to find a design in only nine cases. In one case (seven factors in 128 runs and blocks of size 2), the NOCHECK option is actually unable to find a design in the default time of 60 seconds, whereas the default search has no trouble finding a design.

General Recommendations

Choosing appropriate confounding rules can be difficult, especially if the set $\mathcal{M}$ is complicated. Even if a design is found that satisfies the model specification, it is a good idea to examine the alias structure to make sure that you understand the alias structure that the confounding rules generate. To do so, use the ALIAS option in the EXAMINE statement.

For more information about the general mathematical theory of orthogonal factorial designs, see Bose (1947).

Design Details

Types of Factors

The *factors* of a design are variables that an experimenter can set at several values. In general, experiments are performed to study the effects of different levels of the factors on the response of interest. For example, consider an experiment to maximize the percentage of raw material that responds to a chemical reaction. The factors might include the reaction temperature and the feed rate of the chemicals, whereas the response is the yield rate. Factors of different types are used in different ways in constructing a design. This section defines the different types of factors.

Block factors are unavoidable factors that are known to affect the response, but in a relatively uninteresting way. For example, in the chemical experiment, the technician operating the equipment might have a noticeable effect on the yield of the process. Although the operator effect might be unavoidable, it is usually not very interesting. On the other hand, factors whose effects are directly of interest are called *design factors*. One goal in designing an experiment is to avoid mixing up (*confounding*) the effects of the design factors with the effects of any block factors.

When you construct a design by orthogonal confounding, all factors formally have the same number of levels q, where q is a prime number or a power of a prime number. Usually, $q = 2$ and the factor levels are chosen to represent high and low values.

However, this does not mean, for example, that a design for 2-level factors is restricted to no more than two blocks. Instead, the values of several 2-level factors can be used to index the values of a single factor that has more than two levels. For example, the values of three 2-level factors (P_1, P_2, and P_3) can be used to index the values of an 8-level factor (F), as follows:

P_1	P_2	P_3	F
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The factors P_i are used only to derive the levels of the factor F; thus, they are called *pseudofactors*. F is called a *derived factor*. In general, k q-level pseudofactors give rise to a single q^k-level derived factor.

Block factors can be derived factors, and their associated formal factors (the P_i factors) are called *block pseudofactors*.

The method for constructing an orthogonally confounded design for q-level factors in q^m runs distinguishes between the first m factors and the remaining factors. Each of the q^m different combinations of the first m factors occurs once in the design in an order similar to the preceding table. For this reason, the first m factors are called the *run-indexing factors*.

Table 8.6 summarizes the different types of factors discussed in this section.

Table 8.6 Types of Factors

Block factor	Unavoidable factor whose effect is not of direct interest
Block pseudofactor	Pseudofactor that is used to derive levels of a block factor
Derived factor	Factor whose levels are derived from pseudofactors
Design factor	Factor whose effect is of direct interest
Pseudofactor	Formal factor combined to derive the levels of a real factor
Run-indexing factors	The first m design factors, whose q^m combinations index the runs in the design

Specifying Effects in the MODEL Statement

The FACTEX procedure accepts models that contain terms for main effects and interactions. *Main effects* are specified by writing variable names by themselves:

```
A B C
```

Interactions are specified by joining variable names with asterisks:

```
A*B B*C A*B*C
```

In addition, the *bar operator* (`|`) simplifies specification for interactions. The `@` *operator*, used in combination with the bar operator, further simplifies specification of interactions. For example, two ways of writing the complete set of effects for a model with up to three-factor interactions are as follows:

```
model estimate=(A B C A*B A*C B*C A*B*C);

model estimate=(A|B|C);
```

When the bar (`|`) is used, the right- and left-hand sides become effects and their cross becomes an interaction effect. Multiple bars are permitted. The expressions are expanded from left to right, using rules given by Searle (1971). For example, `A|B|C` is evaluated as follows:

```
A | B | C  →  { A | B } | C
           →  { A B A*B } | C
           →  A  B  A*B  C  A*C  B*C  A*B*C
```

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying the number, preceded by an @ sign, at the end of the bar effect. For example, the specification `A|B|C@2` results in only those effects that contain two or fewer factors. In this case, the effects A, B, A*B, C, A*C, and B*C are generated.

Factor Variable Characteristics in the Output Data Set

When you use the OUTPUT statement to save a design in a data set and you rename and recode a factor, the type and length of the new variable are determined by whether you use the NVALS= options or the CVALS= option. A factor variable whose values are coded by using the NVALS= specification is of numeric type. A factor variable whose values are coded by using the CVALS= option is of character type, and the length of the variable is set to the length of the longest character string; shorter strings are padded with trailing blanks.

For example, consider the following specifications:

```
cvals=('String 1' 'A longer string')
cvals=('String 1' 'String 2')
```

The first value in the first CVALS= specification is padded with seven trailing blanks. One consequence is that it no longer matches the 'String 1' of the second specification. To match two such values (for example, when you merge two designs), use the TRIM function in the DATA step (see *SAS Functions and CALL Routines: Reference*).

Statistical Details

Resolution

The resolution (r) of a design indicates which effects can be estimated free of other effects. The resolution of a design is generally defined as the smallest *order*[1] of the interactions that are confounded with zero. Since having an effect of order $n + m$ confounded with zero is equivalent to having an effect of order n confounded with an effect of order m, the resolution can be interpreted as follows:

- If r is odd, then effects of order $e = (r - 1)/2$ or less can be estimated free of each other. However, at least some of the effects of order e are confounded with interactions of order $e + 1$. A design of odd resolution is appropriate when effects of interest are those of order e or less, and those of order $e + 1$ or higher are all negligible.
- If r is even, then effects of order $e = (r - 2)/2$ or less can be estimated free of each other and are also free of interactions of order $e + 1$. A design of even resolution is appropriate when effects of order e or less are of interest, effects of order $e + 1$ are not negligible, and effects of order $e + 2$ or higher are negligible. If the design uses blocking, interactions of order $e + 1$ or higher might be confounded with blocks.

In particular, for resolution 5 designs, all main effects and two-factor interactions can be estimated free of each other. For resolution 4 designs, all main effects can be estimated free of each other and free of two-factor interactions, but some two-factor interactions are confounded with each other or with blocks (or with both). For resolution 3 designs, all main effects can be estimated free of each other, but some of them are confounded with two-factor interactions.

In general, higher resolutions require larger designs. Resolution 3 designs are popular because they handle relatively many factors in a minimal number of runs. However, they offer no protection against interactions. If resources are available, you should use a resolution 5 design so that all main effects and two-factor

[1] The order of an effect is the number of factors involved in it. For example, main effects have order one, two-factor interactions have order two, and so on.

interactions are independently estimable. If a resolution 5 design is too large, you should use a design of resolution 4, which ensures estimability of main effects free of any two-factor interactions. In this case, if data from the initial design reveal significant effects associated with confounded two-factor interactions, further experiments can be run to distinguish between effects that are confounded with each other in the design. See Example 8.2.

Many references on fractional factorial designs use roman numerals to denote resolution of a design: III, IV, V, and so on. A common notation for an orthogonally confounded design of resolution r for k q-level factors in q^{k-p} runs is

$$q_r^{k-p}$$

For example, 2_{V}^{5-1} denotes a design for five 2-level factors in 16 runs that permits estimation of all main effects and two-factor interactions. This chapter uses arabic numerals for resolution because they correspond directly to the value you can specify in the RESOLUTION= option in the MODEL statement.

Randomization

In many experiments, proper randomization is crucial to the validity of the conclusions. Randomization neutralizes the effects of systematic biases that might be involved in implementing the design and provides a basis for the assumptions underlying the analysis. For a discussion, see Kempthorne (1975).

The way in which randomization is handled depends on whether the design involves blocking:

- For designs that do not have block factors, proper randomization consists of randomly permuting the overall order of the runs and randomly assigning the actual levels of each factor to the theoretical levels it has for the purpose of constructing the design.
- For designs that have block factors, proper randomization calls for first performing separate random permutations for the runs within each block, and then randomly permuting the order in which the blocks are run.

For example, suppose you generate a full factorial design for three 2-level factors A, B, and C, in eight runs. Randomizing this design involves the following steps:

1. Randomly permute the order of the runs:

$$\text{Runs: } \{1,2,3,4,5,6,7,8\} \quad \rightarrow \quad \{3,8,1,2,4,7,6,5\}$$

2. Randomly assign the actual levels to the theoretical levels for each factor:

$$\begin{aligned}\text{Factor A levels: } \{0,1\} &\rightarrow \{1,-1\}\\ \text{Factor B levels: } \{0,1\} &\rightarrow \{1,-1\}\\ \text{Factor C levels: } \{0,1\} &\rightarrow \{-1,1\}\end{aligned}$$

Thus, the effect of the randomization is to transform the original design, as follows:

Run	A	B	C
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

$\longrightarrow$

Run	A	B	C
3	1	–1	–1
8	–1	–1	1
1	1	1	–1
2	1	1	1
4	1	–1	1
7	–1	–1	–1
6	–1	1	1
5	–1	1	–1

If the original design is in two blocks, then the first step is replaced with the following two steps:

1. Randomly permute the order of the runs within each block:

$$\text{Block 1 runs: } \{1,2,3,4\} \rightarrow \{4,1,2,3\}$$
$$\text{Block 2 runs: } \{5,6,7,8\} \rightarrow \{8,7,6,5\}$$

2. Randomly permute the order of the blocks:

$$\text{Block levels: } \{1,2\} \rightarrow \{2,1\}$$

The resulting transformation is shown in the following:

Run	Block	A	B	C
1	1	0	0	0
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0
5	2	0	0	1
6	2	0	1	0
7	2	1	0	0
8	2	1	1	1

$\longrightarrow$

Run	Block	A	B	C
8	2	–1	–1	1
7	2	–1	1	–1
6	2	1	–1	–1
5	2	1	1	1
4	1	–1	–1	–1
1	1	1	1	–1
2	1	1	–1	1
3	1	–1	1	1

If you use the RANDOMIZE option in the OUTPUT statement, the output data set contains a randomized design. In some cases, it is appropriate to randomize the run order but not the assignment of theoretical factor levels to actual levels. In these cases, specify both the NOVALRAN and RANDOMIZE options in the OUTPUT statement.

Replication

In quality improvement applications, it is often important to analyze both the mean response of a process and the variability around the mean. To study variability with an experimental design, you must take several measurements of the response for each different combination of the factors of interest; that is, you must *replicate* the design runs.

Replicating a Fixed Number of Times

A simple method of replication is to take a specified number of measurements for each combination of factor levels in the basic design. You can replicate runs in the design by specifying numbers for the POINTREP= and DESIGNREP= options in the OUTPUT statement. For example, the following code constructs a full 2^2 design and uses both of these options to replicate the design three times:

```
proc factex;
   factors A B;
   output out=one pointrep =3;
run;
   output out=two designrep=3;
run;
```

The output data sets One and Two have the same 12 runs, but they are in different orders. In the data set One, the POINTREP= option causes all three replications of each run to occur together, as shown in Figure 8.8.

Figure 8.8 Four-Run Design Replicated Using the POINTREP= Option

OBS	*A*	*B*	
1	−1	−1	Three replicates of run 1
2	−1	−1	
3	−1	−1	
4	−1	1	Three replicates of run 2
5	−1	1	
6	−1	1	
7	1	−1	Three replicates of run 3
8	1	−1	
9	1	−1	
10	1	1	Three replicates of run 4
11	1	1	
12	1	1	

On the other hand, in the data set Two, the DESIGNREP= option causes all four runs of the design to occur together three times, as shown in Figure 8.9.

Figure 8.9 Four-Run Design Replicated Using the DESIGNREP= Option

	OBS	A	B
Replicate 1	1	−1	−1
	2	−1	1
	3	1	−1
	4	1	1
Replicate 2	5	−1	−1
	6	−1	1
	7	1	−1
	8	1	1
Replicate 3	9	−1	−1
	10	−1	1
	11	1	−1
	12	1	1

Replicating with an Outer Array

Another method of design replication considers the range of environmental conditions over which the process should maintain consistency. This method distinguishes between control factors and noise factors. *Control factors* are factors that are under the control of the designer or the process engineer. *Noise factors* cause the performance of a product to vary when the nominal values of the control variables are fixed (noise factors are controllable for the purposes of experimenting with the process). Typical noise factors are variations in the manufacturing environment or the customer's environment that are due to temperature or humidity. The object of experimentation is to find the best settings for the control factors for a variety of settings for the noise factors. In other words, the goal is to develop a process that runs well in a variety of environments. For further discussion, see Dehnad (1989) and Phadke (1989).

To achieve this goal, a collection of environmental conditions (settings for the noise factors) is determined. This collection is called the *outer array*. Each run in the control factor design (*inner array*) is replicated within each of these environments. The mean and variance of the process over the outer array are computed for each run in the inner array. Either the outer array or the inner array might consist of all possible different settings for the associated factors, or they might be fractions of all possible settings.

You can replicate designs in this way by specifying *SAS-data-set* names for the POINTREP= and DESIGNREP= options in the OUTPUT statement. If you construct a design for your control factors and you want to run a noise factor design for each run in the control factor design, specify the *SAS-data-set* that holds the noise factor design (that is, the *outer array*) in the POINTREP= option in the OUTPUT statement. See Example 8.14.

Confounding Rules

Confounding rules determine the values of factors in terms of the values of the run-indexing factors for a design. (For a discussion of run-indexing factors, see "Types of Factors" on page 646.) The FACTEX procedure uses these rules to construct designs. The confounding rules also determine the alias structure of the design. To display the confounding rules for a design, use the CONFOUNDING option in the EXAMINE statement.

For 2-level factors, the rules are displayed in a multiplicative notation that uses the default values of –1 and +1 for the factors. For example, the following confounding rule means that the level of factor X8 is derived as the product of the levels of factors X1 through X7 for each run in the design:

```
X8 = X1*X2*X3*X4*X5*X6*X7
```

X8 always has a value of –1 or +1 because these are the values of X1 through X7. For factors with $q > 2$ levels, confounding rules are printed in an additive notation and the arithmetic is performed in the Galois field of size q. For example, in a design for 3-level factors, the following confounding rule means that the level of factor F is computed by adding the levels of B and D and two times the levels of C and E, all modulo 3:

```
F = B + (2*C) + D + (2*E)
```

Note that if q is not a prime number, Galois field arithmetic is not equivalent to arithmetic modulo q.

Blocks are introduced into designs by using block pseudofactors. The confounding rule for the ith block pseudofactor has `[B i]` on the left-hand side.

For more information about how confounding rules are constructed, see the section "Suitable Confounding Rules" on page 642.

Alias Structure

The alias structure of a design identifies which effects are confounded (aliased) with each other in the design. The alias structure and confounding rules are different: the confounding rules are used to construct the design, whereas the alias structure is a result of using a particular set of confounding rules. To display the alias structure for a design, use the ALIAS option in the EXAMINE statement.

Examining the alias structure is important because aliased effects cannot be estimated separately from each other. When several effects are listed as equal, the effects are all jointly aliased with one another and form an *alias chain* or *alias string*. For example, the following string is an alias chain that shows the relationship between four 2-factor interactions:

```
Temperature*Moisture=HoldPress*Gage=Thickness*Screw=BoostPress*Time
```

If you want separate estimates of Temperature*Moisture and Thickness*Screw (for example), a design that uses this alias chain would not be acceptable. Designs of even resolution $2k$ contain one or more such chains of confounded k-factor interactions.

By default, the FACTEX procedure displays alias chains that contain effects up to a certain order d, where main effects are order 1, two-factor interactions are order 2, and so on. You can specify the value of d in the ALIASING option, or you can use the default that is calculated by the procedure. Alias chains that are confounded with blocks are displayed with [B] on the left-hand side.

Minimum Aberration

As discussed in the section "Speeding Up the Search" on page 645, the FACTEX procedure uses a tree search algorithm to find the confounding rules of a design that matches the size and resolution you specify. There might be more than one solution set of confounding rules, and usually the FACTEX procedure chooses the first one it finds. However, designs that have the same resolution can still have important differences; to deal with these differences, Fries and Hunter (1980) introduced the concept of *aberration* in confounded fractional factorial designs. This section defines aberration and discusses how to request minimum aberration designs with the FACTEX procedure.

Recall that a design has resolution *r* if *r* is the smallest order of the interactions that are confounded with zero. The idea behind minimum aberration is that the preferred design is a resolution *r* design that confounds as few *r*th-order interactions as possible. Technically, the aberration of a design is the vector $\mathbf{k} = \{k_1, k_2, \ldots\}$, where k_i is the number of *i*th-order interactions that are confounded with zero. A design that has aberration $\mathbf{k}$ has *minimum aberration* if $\mathbf{k} \leq \mathbf{k}'$ for any other design that has aberration $\mathbf{k}'$, in the sense that $k_i < k_i'$ for the first i for which $k_i \neq k_i'$.

For example, consider the resolution 4 design for seven 2-level factors in 32 runs (2^{7-2}_{IV}) discussed in Example 8.11.

By specifying 5 for the order *d* for the ALIASING option, you can see how many fourth- and fifth-order interactions are confounded with zero. By default, the FACTEX procedure constructs a design that confounds two fourth-order interactions and no fifth-order interactions with zero.

```
0 = A*B*F*G = C*D*E*G
```

Thus, part of the aberration for this design is

$$\{k_3, k_4, k_5, \ldots\} \quad = \quad \{0, 2, 0, \ldots\}$$

On the other hand, the MINABS option constructs a design that confounds only one fourth-order interaction and two fifth-order interactions with zero, as follows:

```
0 = C*D*E*F = A*B*C*F*G = A*B*D*E*G
```

Thus, part of the aberration for this design is

$$\{k_3', k_4', k_5', \ldots\} \quad = \quad \{0, 1, 2, \ldots\}$$

Because the two aberrations first differ for k_4 and k_4' and because $k_4' < k_4$, the aberration for the second design is less than the aberration for the first design.

The definition of aberration requires evaluating the number of *i*th-order interactions that are confounded with zero for all $i \leq k$, where k is the number of factors. Because there are q^k generalized interactions between k q-level factors, this evaluation can be prohibitive when there are many factors. Moreover, it is unnecessary if you are interested only in small-order interactions, as is usually the case. Therefore, when you specify the MINABS option, by default, the FACTEX procedure evaluates the aberration only up to order *d*, where *d* is the same as the default maximum order for listing the aliasing (see the specifications for the EXAMINE

statement in the section "EXAMINE Statement" on page 632). You can set *d* to any level by specifying (*d*) as the first argument after the MINABS option.

The discussion so far has dealt only with fractional unblocked designs, but one more point to consider is the definition of aberration for block designs. Define a vector, $\mathbf{b} = b_1, b_2, \ldots$, similar to the aberration vector $\mathbf{k}$, except that b_i is the number of ith-order interactions that are confounded with blocks. A block design with $\mathbf{k}$ and $\mathbf{b}$ has minimum aberration if the following are true:

- $\mathbf{k}$ is minimum
- among all designs with minimum $\mathbf{k}$, $\mathbf{b}$ is minimum

MaxClear Designs

As discussed in the section "Alias Structure" on page 653, the alias structure for a factorial design can tell you important information about which effects are confounded and hence cannot be estimated separately from one another. In some cases, you cannot avoid the fact that some potentially active effects are aliased; for example, in resolution 4 designs, some two-factor interactions are aliased with each other and hence cannot be jointly estimated. In this case, you might want a design that has as many two-factor interactions as possible unaliased with any other interaction—that is, as many *clear* two-factor interactions as possible. This is known as the *MaxClear* design, and you can use the MAXCLEAR option in the MODEL statement to request it.

To explore how well a particular design performs on the MaxClear criterion, you can use the ALIASING option in the EXAMINE statement to examine the alias structure. Clear interactions are interactions that are displayed by themselves, with no other interactions in their alias chain. Alternatively, the SUMMARY option in the EXAMINE statement displays the total number of interactions up to a certain order *d*, how many of those are unaliased with interactions of lower order and are thus in a sense estimable, and how many are unaliased with any interactions of order *d* or lower and are thus clear.

Obviously, whether an interaction is clear depends on what other effects are considered to be potentially of interest. For a particular design, the default order d for considering interaction clarity is the same as the default order d of interactions that are included in the alias structure. As with the alias structure, you can specify an alternative value of *d* in the MAXCLEAR option in the MODEL statement or in the SUMMARY option in the EXAMINE statement.

Split-Plot Designs

As discussed in the section "Structure of General Factorial Designs" on page 641, for a design that has q-level factors in q^m runs, the FACTEX procedure usually treats the first m factors of the design as the run-indexing factors, and computes the levels of all other factors as linear combinations of these over the Galois field of order q. However, when you restrict the design's randomization by using the BLOCKS UNITS= option and UNITEFFECT statement to specify *unit-factors* and *unit-effects*, PROC FACTEX instead computes the levels of all factors (including the first m) in terms of underlying plot-indexing pseudofactors that are distinct from the factors named in the FACTORS statement. These plot-indexing pseudofactors are denoted $[i]$, for $i = 1, \ldots, m$, and they are associated with *unit-factors* as follows. Suppose the BLOCK UNIT= specification has the form

```
blocks units=(Stage1=n1 Stage2=n2 ...);
```

where $n_1 = q^{k_1}, n_2 = q^{k_2}, \ldots$ Then the first unit factor, Stage1, is identified with all possible interactions between the first k_1 plot-indexing pseudofactors, the second with the next k_2 pseudofactors, and so on. If you save a split-plot design to a data set by using the OUTPUT statement, then the plot-indexing pseudofactors are also included in the data set with names _1_, _2_, ..., up to the base-q logarithm of the number of runs.

The whole-plot and subplot constraints that are specified in the UNITEFFECT statement define the relation between the plot-indexing pseudofactors that correspond to the specified *unit-effect* and the factor effects that are specified in the WHOLE= and SUB= options. In particular, with a BLOCK UNIT= specification of the previous form, a UNITEFFECT statement of the following form means that the *Stage-1-effects* should be aliased only with interactions between the first k_1 plot-indexing pseudofactors:

```
uniteffect Stage1 / whole=(Stage-1-effects);
```

In contrast, a UNITEFFECT statement of the following form means that the *Stage-2-effects* should not be aliased with interactions between the first $k_1 + k_2$ plot-indexing pseudofactors:

```
uniteffect Stage1*Stage2 / sub=(Stage-2-effects);
```

Summary of Designs

Table 8.7 summarizes basic design types that you can construct with the FACTEX procedure by providing example code for each type.

Table 8.7 Basic Designs Constructed by the FACTEX Procedure

Design Type	Example Statements

A full factorial design in three factors, each at two levels coded as –1 and +1.	`proc factex;`
	`  factors Pressure Temperature Time;`
	`  examine design;`
	`run;`

A full factorial design in three factors, each at three levels coded as –1, 0, and +1.	`proc factex;`
	`  factors Pressure Temperature Time / nlev= 3;`
	`  examine design;`
	`run;`

A full factorial design in three factors, each at two levels. The entire design is replicated twice, and the design with recoded factor levels is saved in a SAS data set.	`proc factex;`
	`  factors Pressure Temperature Time;`
	`  output out= SavedDesign designrep= 2`
	`  Pressure cvals=('low' 'high')`
	`  Temperature nvals=(200 300)`
	`  Time nvals=(10 20);`
	`run;`

Table 8.7 *continued*

Design Type	Example Statements

A full factorial design in three factors, each at two levels coded as –1 and +1. Each run in the design is replicated three times, and the replicated design is randomized and saved in a SAS data set.	`proc factex;`
	`  factors Pressure Temperature Time;`
	`  output out= SavedDesign`
	`  pointrep= 3 randomize;`
	`run;`

A full factorial design in three control factors, each at two levels coded as –1 and +1. A noise factor design (*outer array*) is read from a SAS data set and replicated for each run in the control factor design (*inner array*), and the product design is saved in a SAS data set.	`proc factex;`
	`factors+ Pressure Temperature Time;`
	`  output out =+ SavedDesign`
	`  pointrep=+ OutArray;`
	`run;`

A full factorial blocked design in three factors, each at two levels coded as –1 and +1. The design is arranged in two blocks and saved in a SAS data set. By default, the block variable is named BLOCK and the two block levels are numbered 1 and 2.	`proc factex;`
	`  factors Pressure Temperature Time;`
	`  blocks nblocks= 2;`
	`  output out= SavedDesign;`
	`run;`

A full factorial blocked design in three factors, each at two levels coded as –1 and +1. Each block contains four runs; the block variable is renamed and the block levels of character type are recoded. The design is saved in a SAS data set.	`proc factex;`
	`  factors Pressure Temperature Time;`
	`  blocks size= 4;`
	`  output out= SavedDesign`
	`  blockname= Machine cvals=('A' 'B' );`
	`run;`

A fractional factorial design of resolution 4 in four factors, each at two levels coded as –1 and +1. The size of the design is eight runs.	`proc factex;`
	`  factors Pressure Temperature Time Catalyst;`
	`  size design= 8;`
	`  model resolution= 4;`
	`  examine design;`
	`run;`

Table 8.7 *continued*

Design Type	Example Statements

A one-half fraction of a factorial design in four factors, each at two levels coded as –1 and +1. The design is of maximum resolution. The design points, the alias structure, and the confounding rules are listed.	`proc factex;`
	`  factors Pressure Temperature Time Catalyst;`
	`  size fraction= 2;`
	`  model resolution=maximum;`
	`  examine design aliasing confounding;`
	`run;`

A one-quarter fraction of a factorial design in six factors, each at two levels coded as –1 and +1. Main effects are estimated, and some two-factor interactions are considered nonnegligible. The design is saved in a SAS data set.	`proc factex;`
	`  factors x1-x6;`
	`  size fraction= 4;`
	`  model estimate=( x1 x2 x3 x4 x5 x6 )`
	`  nonneg =( x1*x5 x1*x6 x5*x6 );`
	`  output out = SavedDesign;`
	`run;`

Output

By default, the FACTEX procedure does not display any output. For each design that it constructs, the procedure displays a message in the SAS log that provides the following information:

- the number of runs in the design
- the number of blocks and the block size, if appropriate
- the maximum resolution of the design

The DESIGN option in the EXAMINE statement displays the coded runs in the design that uses standard values, as described in the section "OUTPUT Statement" on page 636. The CONFOUNDING option in the EXAMINE statement displays the confounding rules that are used to construct the design. The ALIAS option in the EXAMINE statement displays the aliasing structure for the design.

When you specify the OUTPUT statement, the FACTEX procedure also creates output data sets. Because PROC FACTEX is interactive, you can use many OUTPUT statements in a single run of the FACTEX procedure to produce many output data sets if you separate them with RUN statements.

ODS Tables

The following table summarizes the ODS tables that you can request with the PROC FACTEX statement.

Table 8.8 ODS Tables Produced in PROC FACTEX

ODS Table Name	Description	Statement	Option
DesignPoints	Design points	EXAMINE	DESIGN
FactorRules	Treatment factor confounding rules	EXAMINE	CONFOUNDING
BlockRules	Block factor confounding rules	EXAMINE	CONFOUNDING
Aliasing	Alias structure	EXAMINE	ALIASING

Examples: FACTEX Procedure

Example 8.1: Completely Randomized Design

NOTE: See *A Completely Randomized Design* in the SAS/QC Sample Library.

An experimenter wants to study the effect of cutting speed (Speed) on the surface finish of a component. He considers testing the components at five levels of cutting speed (100, 125, 150, 175, and 200) and decides to test five components at each level.

A single-factor completely randomized design that has five levels and 25 runs is used. The following statements generate the required design:

```
proc factex;
   factors Speed / nlev=5;
   size design=25;
   output out=SurfaceExperiment randomize(713)
      Speed nvals=(100 125 150 175 200);
run;
proc print data=SurfaceExperiment;
run;
```

The RANDOMIZE option in the OUTPUT statement randomizes the run order; the random seed (713 here) is optional. The design, which is saved in the data set SurfaceExperiment, is displayed in Output 8.1.1.

Output 8.1.1 A Completely Randomized Design

Obs	Speed
1	200
2	175
3	200
4	125
5	100
6	150
7	175
8	125
9	100
10	100
11	100
12	200
13	125
14	125
15	150
16	175
17	175
18	150
19	175
20	150
21	200
22	125
23	200
24	150
25	100

If you are working through this example on your computer, you might find a different run order in your output because your computer uses a different seed value for the random number generator. You can specify a seed value in the RANDOMIZE option.

Example 8.2: Resolution 4 Augmented Design

NOTE: See *Resolution IV Augmented Design* in the SAS/QC Sample Library.

Box, Hunter, and Hunter (1978) describe an injection molding experiment that involves eight 2-level factors: mold temperature (Temp), moisture content (Moisture), holding pressure (HoldPress), cavity thickness (Thick), booster pressure (BoostPress), cycle time (Time), screw speed (Speed), and gate size (Gate).

The design used has 16 runs and is of resolution 4; it is often denoted as 2^{8-4}_{IV}. You can generate this design, shown in Output 8.2.1, with the following statements:

```
proc factex;
   factors Temp        Moisture HoldPress Thick
           BoostPress  Time     Speed     Gate;
   size design=16;
   model resolution=4;
   examine design aliasing;
run;
```

The FACTORS statement lists the factor names. The DESIGN=16 option in the SIZE statement specifies the design size. The RESOLUTION=4 specifies the resolution of the design. The EXAMINE statement lists points and aliasing.

Output 8.2.1 A 2^{8-4}_{IV} Design

The FACTEX Procedure

Design Points								
Experiment Number	Temp	Moisture	HoldPress	Thick	BoostPress	Time	Speed	Gate
1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	1	1	1	1	-1
3	-1	-1	1	-1	1	1	-1	1
4	-1	-1	1	1	-1	-1	1	1
5	-1	1	-1	-1	1	-1	1	1
6	-1	1	-1	1	-1	1	-1	1
7	-1	1	1	-1	-1	1	1	-1
8	-1	1	1	1	1	-1	-1	-1
9	1	-1	-1	-1	-1	1	1	1
10	1	-1	-1	1	1	-1	-1	1
11	1	-1	1	-1	1	-1	1	-1
12	1	-1	1	1	-1	1	-1	-1
13	1	1	-1	-1	1	1	-1	-1
14	1	1	-1	1	-1	-1	1	-1
15	1	1	1	-1	-1	-1	-1	1
16	1	1	1	1	1	1	1	1

The alias structure is shown in Output 8.2.2.

Output 8.2.2 Alias Structure for a 2^{8-4}_{IV} Design

Aliasing Structure
Temp
Moisture
HoldPress
Thick
BoostPress
Time
Speed
Gate
Temp*Moisture = HoldPress*Gate = Thick*Speed = BoostPress*Time
Temp*HoldPress = Moisture*Gate = Thick*Time = BoostPress*Speed
Temp*Thick = Moisture*Speed = HoldPress*Time = BoostPress*Gate
Temp*BoostPress = Moisture*Time = HoldPress*Speed = Thick*Gate
Temp*Time = Moisture*BoostPress = HoldPress*Thick = Speed*Gate
Temp*Speed = Moisture*Thick = HoldPress*BoostPress = Time*Gate
Temp*Gate = Moisture*HoldPress = Thick*BoostPress = Time*Speed

Subsequent analysis of the data collected for the design suggests that HoldPress and BoostPress have statistically significant effects. There also seems to be a significant effect associated with the sum of the aliased two-factor interactions Temp*BoostPress, Moisture*Time, HoldPress*Speed, and Thick*Gate. This chain of confounded interactions is identified in Output 8.2.2.

A few runs can be added to the design to distinguish between the effects that are caused by these four interactions. You simply need a design in which these four effects are estimable, regardless of all other main effects and interactions. For example, the following statements generate a suitable set of runs:

```
proc factex nocheck;
   factors Temp        Moisture HoldPress Thick
           BoostPress  Time     Speed     Gate;
   model estimate=(Temp*BoostPress
                   Moisture*Time
                   HoldPress*Speed
                   Thick*Gate);
   size design=min;
   examine design aliasing(2);
run;
```

The DESIGN=MIN option directs PROC FACTEX to search for the smallest design that allows all four interactions to be estimated. Eight runs are required: see Output 8.2.3.

Output 8.2.3 Additional Runs to Resolve Ambiguities

The FACTEX Procedure

	Design Points							
Experiment Number	Temp	Moisture	HoldPress	Thick	BoostPress	Time	Speed	Gate
1	-1	-1	-1	-1	-1	-1	-1	1
2	-1	-1	1	1	1	1	1	-1
3	-1	1	-1	1	1	1	1	-1
4	-1	1	1	-1	-1	-1	-1	1
5	1	-1	-1	1	1	1	1	1
6	1	-1	1	-1	-1	-1	-1	-1
7	1	1	-1	-1	-1	-1	-1	-1
8	1	1	1	1	1	1	1	1

Output 8.2.4 shows the alias structure of the additional eight runs. Note that the following alias chain of interest from the original design is broken:

```
Temp*BoostPress=Moisture*Time=HoldPress*Speed=Thick*Gate
```

In this new set of runs, these four interactions are aliased with main effects and with other two-factor interactions, but they are unaliased with each other. Therefore, when these four runs are added to the original 16 runs, the main effects of the eight factors plus the four 2-factor interactions that were originally aliased with each other can all be estimated with the 20 runs.

Output 8.2.4 Alias Structure of the Additional Experiment

Aliasing Structure
0 = Thick*BoostPress = Thick*Time = Thick*Speed = BoostPress*Time = BoostPress*Speed = Time*Speed
Temp = Thick*Gate = BoostPress*Gate = Time*Gate = Speed*Gate
Moisture = HoldPress*Gate
HoldPress = Moisture*Gate
Thick = BoostPress = Time = Speed = Temp*Gate
Gate = Temp*Thick = Temp*BoostPress = Temp*Time = Temp*Speed = Moisture*HoldPress
Temp*Moisture = HoldPress*Thick = HoldPress*BoostPress = HoldPress*Time = HoldPress*Speed
Temp*HoldPress = Moisture*Thick = Moisture*BoostPress = Moisture*Time = Moisture*Speed

Example 8.3: Factorial Design with Center Points

NOTE: See *A Factorial Design with Center Points* in the SAS/QC Sample Library.

Factorial designs that involve two levels are the most popular experimental designs. For two-level designs, it is assumed that the response is close to linear over the range of the factor levels. To check for curvature and to obtain an independent estimate of error, you can replicate points at the center of a two-level design. Adding center points to the design does not affect the estimates of factorial effects.

To construct a design that has center points, you first create a data set that has factorial points by using the FACTEX procedure and then augment it with center points by using a simple DATA step. This example illustrates this technique.

A researcher is studying the effect of three 2-level factors—current (Current), voltage (Voltage), and time (Time)—by conducting an experiment that uses a complete factorial design. The researcher is interested in studying the overall curvature over the range of factor levels by adding four center points.

You can construct this design in two stages. First, create the basic 2^3 design with the following statements:

```
proc factex;
   factors Current Voltage Time;
   output out=Factorial
          Current nvals=(12  28)
          Voltage nvals=(100 200)
          Time    nvals=(50  60);
run;
```

Next, create the center points and append to the basic design as follows:

```
data Center(drop=i);
   do i = 1 to 4;
      Current = 20;
      Voltage = 150;
      Time    = 55;
      output;
   end;
data CPDesign;
   set Factorial Center;
run;
```

```
proc print data=CPDesign;
run;
```

The design, which is saved in the data set CPDesign, is displayed in Output 8.3.1. Observations 1 to 8 are the factorial points, and observations 9 to 12 are the center points.

Output 8.3.1 A 2^3 Design with Four Center Points

Obs	Current	Voltage	Time
1	12	100	50
2	12	100	60
3	12	200	50
4	12	200	60
5	28	100	50
6	28	100	60
7	28	200	50
8	28	200	60
9	20	150	55
10	20	150	55
11	20	150	55
12	20	150	55

Example 8.4: Fold-Over Design

NOTE: See *A Fold-Over Design* in the SAS/QC Sample Library.

Folding over a fractional factorial design is a method for breaking the links between aliased effects in a design. Folding over a design means adding a new fraction that is identical to the original fraction except that the signs of all the factors are reversed. The new fraction is called a *fold-over* design. Combining a fold-over design with the original fraction converts a design of odd resolution r into a design of resolution $r + 1$. (This is not true if the original design has even resolution.) For example, folding over a resolution 3 design yields a resolution 4 design. You can use the FACTEX procedure to construct the original design fraction and a DATA step to generate the fold-over design.

Consider a $\frac{1}{8}$ fraction of a 2^6 factorial design that has factors A, B, C, D, E, and F. The following statements construct a 2^{6-3}_{III} design:

```
proc factex;
   factors A B C D E F;
   size fraction=8;
   model resolution=3;
   examine aliasing;
   output out=Original;
run;

title 'Original Design';
proc print data=Original;
run;
```

The option FRACTION=8 in the SIZE statement specifies a $\frac{1}{8}$ fraction of a complete factorial—that is, 8 ($=\frac{1}{8}2^6$). The design, which is saved in the data set Original, is displayed in Output 8.4.1.

Output 8.4.1 A 2^{6-3}_{III} Design

Original Design

Obs	A	B	C	D	E	F
1	-1	-1	-1	-1	1	1
2	-1	-1	1	1	-1	-1
3	-1	1	-1	1	-1	1
4	-1	1	1	-1	1	-1
5	1	-1	-1	1	1	-1
6	1	-1	1	-1	-1	1
7	1	1	-1	-1	-1	-1
8	1	1	1	1	1	1

Because the design is of resolution 3, the alias structure in Output 8.4.2 indicates that all the main effects are confounded with the two-factor interactions.

Output 8.4.2 Alias Structure for a 2^{6-3}_{III} Design

The FACTEX Procedure

Aliasing Structure
A = C*F = D*E
B = C*E = D*F
C = A*F = B*E
D = A*E = B*F
E = A*D = B*C
F = A*C = B*D
A*B = C*D = E*F

To separate the main effects and the two-factor interactions, augment the original design with a 1/8 fraction in which the signs of all the factors are reversed. The combined design (original design and fold-over design) of resolution 4 breaks the alias links between the main effects and the two-factor interactions. The fold-over design can be created by using the following DATA step:

```
data FoldOver;
   set Original;
   A=-A; B=-B; C=-C;
   D=-D; E=-E; F=-F;
run;
title 'Fold-Over Design';
proc print data=FoldOver;
run;
```

Here, the DATA step creates the fold-over fraction by reversing the signs of the values of the factors in the original fraction. The fold-over design is displayed in Output 8.4.3.

Output 8.4.3 A 2^{6-3}_{III} Design with Signs Reversed

Fold-Over Design

Obs	A	B	C	D	E	F
1	1	1	1	1	-1	-1
2	1	1	-1	-1	1	1
3	1	-1	1	-1	1	-1
4	1	-1	-1	1	-1	1
5	-1	1	1	-1	-1	1
6	-1	1	-1	1	1	-1
7	-1	-1	1	1	1	1
8	-1	-1	-1	-1	-1	-1

Example 8.5: Randomized Complete Block Design

NOTE: See *A Randomized Complete Block Design* in the SAS/QC Sample Library.

In a randomized complete block design (RCBD), each level of a "treatment" appears once in each block, and each block contains all the treatments. The order of treatments is randomized separately for each block. You can use the FACTEX procedure to create RCBDs.

Suppose you want to construct an RCBD that has six treatments in four blocks. To test each treatment once in each block, you need 24 experimental units. The following statements construct the randomized complete block design that is shown in Output 8.5.1:

```
proc factex;
   factors Block / nlev=4;
   output out=Blocks Block nvals=(1 2 3 4) randomize(12345);
run;
   factors Treatment / nlev=6;
   output out=RCBD
          designrep=Blocks
          randomize(54321)
          Treatment cvals=('A' 'B' 'C' 'D' 'E' 'F');
run;
quit;
proc print data=RCBD;
run;
```

Note that the order of the runs within each block is randomized and that the blocks (1, 2, 3, and 4) are in a random order.

Output 8.5.1 A Randomized Complete Block Design

Obs	Block	Treatment
1	3	F
2	3	D
3	3	C
4	3	A
5	3	B
6	3	E
7	2	C
8	2	D
9	2	F
10	2	B
11	2	E
12	2	A
13	1	C
14	1	F
15	1	B
16	1	E
17	1	A
18	1	D
19	4	A
20	4	D
21	4	C
22	4	F
23	4	E
24	4	B

Example 8.6: Two-Level Design with Design Replication and Point Replication

NOTE: See *A Two-Level Design with Replication* in the SAS/QC Sample Library.

You can replicate a design to obtain an independent estimate of experimental error or to estimate effects more precisely. There are two ways you can replicate a design with the FACTEX procedure: you can replicate the entire design by using the DESIGNREP= option, or you can replicate each point in the design by using the POINTREP= option. The following example illustrates the difference.

A process engineer is conducting an experiment to study the shrinkage of an injection-molded plastic component. The engineer chooses to determine the effect of the following four factors, each at two levels: holding pressure (Pressure), molding temperature (Temperature), cooling time (Time), and injection velocity (Velocity).

The design used is a half-fraction of a 2^4 factorial design, denoted as 2^{4-1}_{IV}. The following statements construct the design:

```
proc factex;
   factors Pressure Temperature Time Velocity;
   size fraction=2;
   model res=max;
   output out=Unreplicated;
run;
proc print data=Unreplicated;
run;
```

The design, saved in the data set Unreplicated), is shown in Output 8.6.1.

Output 8.6.1 Unreplicated Design

Obs	Pressure	Temperature	Time	Velocity
1	-1	-1	-1	-1
2	-1	-1	1	1
3	-1	1	-1	1
4	-1	1	1	-1
5	1	-1	-1	1
6	1	-1	1	-1
7	1	1	-1	-1
8	1	1	1	1

To obtain a more precise estimate of the experimental error, the engineer decides to replicate the entire design three times. The following statements generate a 2^{4-1}_{IV} design with three replicates in 24 runs:

```
proc factex;
   factors Pressure Temperature Time Velocity;
   size fraction=2;
   model res=max;
   output out=Replicated designrep=3;
run;
proc print data=Replicated;
run;
```

The design, which is saved in the data set Replicated, is displayed in Output 8.6.2.

Output 8.6.2 Design Replication

Obs	Pressure	Temperature	Time	Velocity
1	-1	-1	-1	-1
2	-1	-1	1	1
3	-1	1	-1	1
4	-1	1	1	-1
5	1	-1	-1	1
6	1	-1	1	-1
7	1	1	-1	-1
8	1	1	1	1
9	-1	-1	-1	-1
10	-1	-1	1	1
11	-1	1	-1	1
12	-1	1	1	-1
13	1	-1	-1	1
14	1	-1	1	-1
15	1	1	-1	-1
16	1	1	1	1
17	-1	-1	-1	-1
18	-1	-1	1	1
19	-1	1	-1	1
20	-1	1	1	-1
21	1	-1	-1	1
22	1	-1	1	-1
23	1	1	-1	-1
24	1	1	1	1

The first replicate contains observations 1 to 8, the second replicate contains observations 9 to 16, and the third replicate contains observations 17 to 24.

Now, instead of replicating the entire design, suppose the engineer decides to replicate each run in the design three times. The following statements construct a 2^{4-1}_{IV} design in 24 runs with point replication:

```
proc factex;
   factors Pressure Temperature Time Velocity;
   size fraction=2;
   model res=max;
   output out=PointReplicated pointrep=3;
run;
proc print data=PointReplicated;
run;
```

The design, which is saved in the data set PointReplicated, is displayed in Output 8.6.3. The first design point is replicated three times (observations 1–3), the second design point is replicated three times (observations 4–6), and so on.

Output 8.6.3 Point Replication

Obs	Pressure	Temperature	Time	Velocity
1	-1	-1	-1	-1
2	-1	-1	-1	-1
3	-1	-1	-1	-1
4	-1	-1	1	1
5	-1	-1	1	1
6	-1	-1	1	1
7	-1	1	-1	1
8	-1	1	-1	1
9	-1	1	-1	1
10	-1	1	1	-1
11	-1	1	1	-1
12	-1	1	1	-1
13	1	-1	-1	1
14	1	-1	-1	1
15	1	-1	-1	1
16	1	-1	1	-1
17	1	-1	1	-1
18	1	-1	1	-1
19	1	1	-1	-1
20	1	1	-1	-1
21	1	1	-1	-1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Note the difference in the arrangement of the designs created by using design replication (Output 8.6.2) and point replication (Output 8.6.3). In design replication, the original design is replicated a specified number of times; but in point replication, each run in the original design is replicated a specified number of times. For more information about design replication, see the section "Replication" on page 650.

Example 8.7: Mixed-Level Design Using Design Replication and Point Replication

NOTE: See *A Mixed-Level Design Using Replication* in the SAS/QC Sample Library.

Orthogonal factorial designs are most commonly used at the initial stages of experimentation. At these stages, it is best to experiment with as few levels of each factor as possible in order to minimize the number of runs required. Thus, these designs usually involve only two levels of each factor. Occasionally some factors naturally have more than two levels of interest—different types of seed, for example.

You can create designs for factors that have different numbers of levels simply by taking the crossproduct of component designs in which the factors all have the same numbers of levels—that is, replicating every run of one design for each run of the other. (See Example 8.14.) All estimable effects in each component design, in addition to all generalized interactions between estimable effects in different component designs, are estimable in the crossproduct (Chakravarti 1956, sec. 3).

This example illustrates how you can construct a mixed-level design by using the POINTREP= option or the DESIGNREP= option in the OUTPUT statement to take the crossproduct between two designs.

Suppose you want to construct a mixed-level factorial design for two 2-level factors (A and B) and one 3-level factor (C) with 12 runs. The following SAS statements use design replication to produce a complete 3×2^2 factorial design:

```
proc factex;
   factors A B;
   output out=ab;
run;
   factors C / nlev=3;
   output out=DesignReplicated designrep=ab;
run;
proc print data=DesignReplicated;
run;
```

Output 8.7.1 lists the mixed-level design that is saved in the data set DesignReplicated.

Output 8.7.1 3×2^2 Mixed-Level Design Using Design Replication

Obs	A	B	C
1	-1	-1	-1
2	-1	-1	0
3	-1	-1	1
4	-1	1	-1
5	-1	1	0
6	-1	1	1
7	1	-1	-1
8	1	-1	0
9	1	-1	1
10	1	1	-1
11	1	1	0
12	1	1	1

You can also create a mixed-level design for the preceding factors by using the point replication feature of the FACTEX procedure. The following SAS statements use point replication to produce a complete $2^2 \times 3$ factorial design:

```
proc factex;
   factors A B;
   output out=ab;
run;
   factors C / nlev=3;
   output out=PointReplicated pointrep=ab;
run;
proc print data=PointReplicated;
run;
```

Output 8.7.2 lists the mixed-level design that is saved in the data set PointReplicated.

Output 8.7.2 $2^2 \times 3$ Mixed-Level Design Using Point Replication

Obs	C	A	B
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	0	-1	-1
6	0	-1	1
7	0	1	-1
8	0	1	1
9	1	-1	-1
10	1	-1	1
11	1	1	-1
12	1	1	1

Note the difference between the designs in Output 8.7.1 and Output 8.7.2. In design replication, the mixed-level design is given by AB ⊗ C, whereas for point replication the mixed-level design is given by C ⊗ AB, where ⊗ denotes the direct product. In design replication, you can view the DESIGNREP= data set as nested *outside* the design; in point replication, you can view the POINTREP= data set as nested *inside* the design.

Example 8.8: Mixed-Level Design Using Pseudofactors

NOTE: See *Mixed-Level Designs Using Pseudofactors* in the SAS/QC Sample Library.

If the numbers of levels for the factors of the mixed-level design are all powers of the same prime power q, you can construct the design by using *pseudofactors*, where the levels of k q-level pseudofactors are associated with the levels of a single *derived factor* that has q^k levels. For more information, see Chakravarti (1956, sec. 5) and the section "Types of Factors" on page 646.

For example, the following statements create a design for one 4-level factor (A) and three 2-level factors (B, C, and D) in 16 runs (a half replicate):

```
proc factex;
   factors A1 A2 B C D;
   model estimate     =(B C D   A1|A2                    )
         nonnegligible=(B|C|D@2 A1|A2|B A1|A2|C A1|A2|D);
   size design=16;
   output out=DesignA [A1 A2]=A cvals = ('A' 'B' 'C' 'D');
run;
proc print;
   var A B C D;
run;
```

The levels of two 2-level pseudofactors (A1 and A2) are used to represent the four levels of A. Hence, the three degrees of freedom associated with A are produced by the main effects of A1 and A2 and their interaction A1*A2, and you can thus refer to (A1|A2) as the main effect of A.

The MODEL statement specifies that the main effects of all factors are to be estimable and that all the two-factor interactions between B, C, and D, in addition to the interactions between each of these and (A1|A2),

are to be nonnegligible. As a result, the mixed-level design has resolution 4. The design is saved in the data set DesignA, combining the levels of the two pseudofactors, A1 and A2, to obtain the levels of the 4-level factor A. The data set DesignA is listed in Output 8.8.1.

Output 8.8.1 4×2^3 Design of Resolution 4 in 16 Runs

Obs	A	B	C	D
1	A	-1	-1	1
2	A	-1	1	-1
3	A	1	-1	-1
4	A	1	1	1
5	C	-1	-1	-1
6	C	-1	1	1
7	C	1	-1	1
8	C	1	1	-1
9	B	-1	-1	-1
10	B	-1	1	1
11	B	1	-1	1
12	B	1	1	-1
13	D	-1	-1	1
14	D	-1	1	-1
15	D	1	-1	-1
16	D	1	1	1

Example 8.9: Mixed-Level Design by Collapsing Factors

NOTE: See *Mixed-Level Design with Collapsing Factors* in the SAS/QC Sample Library.

You can construct a mixed-level design by *collapsing* factors—that is, by replacing a factor that has n levels by a factor that has m levels, where $m < n$. Orthogonality is retained in the sense that estimates of different effects are uncorrelated, although not all estimates have equal variance (Chakravarti 1956, sec. 6). This method has been used by Addelman (1962) to derive main effects plans for factors that have mixed numbers of levels and by Margolin (1967) to construct plans that consider two-factor interactions.

You can use the value specification in the NVALS= option in the OUTPUT statement as a convenient tool for collapsing factors. For example, the following statements create a 27-run design for two 2-level factors (x1 and x2) and two 3-level factors (x3 and x4) such that all main effects and two-factor interactions are uncorrelated:

```
proc factex;
   factors x1-x4 / nlev = 3;
   size design=27;
   model r=4;
   output out=MixedLevel x1 nvals=(-1 1 -1)
                         x2 nvals=(-1 1 -1);
run;
proc print data=MixedLevel;
run;
```

The mixed-level design is a three-quarter fraction with resolution 5 (Margolin 1967, sec. 6). The design is displayed in Output 8.9.1.

Output 8.9.1 $2^2 \times 3^2$ Design of Resolution V in 27 Runs

Obs	x1	x2	x3	x4
1	-1	-1	-1	-1
2	-1	-1	0	1
3	-1	-1	1	0
4	-1	1	-1	1
5	-1	1	0	0
6	-1	1	1	-1
7	-1	-1	-1	0
8	-1	-1	0	-1
9	-1	-1	1	1
10	1	-1	-1	1
11	1	-1	0	0
12	1	-1	1	-1
13	1	1	-1	0
14	1	1	0	-1
15	1	1	1	1
16	1	-1	-1	-1
17	1	-1	0	1
18	1	-1	1	0
19	-1	-1	-1	0
20	-1	-1	0	-1
21	-1	-1	1	1
22	-1	1	-1	-1
23	-1	1	0	1
24	-1	1	1	0
25	-1	-1	-1	1
26	-1	-1	0	0
27	-1	-1	1	-1

Example 8.10: Design That Uses a Hyper-Graeco-Latin Square

NOTE: See *Hyper-Graeco-Latin Square* in the SAS/QC Sample Library.

A $q \times q$ Latin square is an arrangement of q symbols, each repeated q times in a square whose sides have length q such that each symbol appears exactly once in each row and once in each column. Such arrangements are useful as designs for *row-and-column* experiments, where it is necessary to balance the effects of two q-level factors simultaneously.

A Graeco-Latin square is actually a pair of Latin squares; when superimposed, each symbol in one square occurs exactly once with each symbol in the other square. The following is an example of a 5×5 Graeco-Latin square, where Latin letters are used for the symbols of one square and Greek letters are used for the symbols of the other square:

$$\begin{array}{ccccc} A\alpha & B\beta & C\gamma & D\delta & E\epsilon \\ B\gamma & C\delta & D\epsilon & E\alpha & A\beta \\ C\epsilon & D\alpha & E\beta & A\gamma & B\delta \\ D\beta & E\gamma & A\delta & B\epsilon & C\alpha \\ E\delta & A\epsilon & B\alpha & C\beta & D\gamma \end{array}$$

Whenever q is a power of a prime number, you can construct up to $q - 1$ squares, each with q symbols that are balanced over all the other factors. The result is called a *hyper-Graeco-Latin square* or a complete set of *mutually orthogonal* Latin squares. Such arrangements can be useful as designs (Williams 1949), or they can be used to construct other designs.

When q is a prime power, hyper-Graeco-Latin squares are straightforward to construct with the FACTEX procedure. This is because a complete set of $q - 1$ mutually orthogonal $q \times q$ Latin squares is equivalent to a resolution 3 design for $q + 1$ q-level factors in q^2 runs, where two of the factors index rows and columns and each of the remaining factors indexes the treatments of one of the squares.

For example, the following statements generate a complete set of three mutually orthogonal 4×4 Latin squares, with rows indexed by the factor Row, columns indexed by the factor Column, and the treatment factors in the respective squares indexed by t1, t2, and t3. The first step is to construct a resolution 3 design for five 4-level factors in 16 runs.

```
proc factex;
   factors Row Column t1-t3 / nlev=4;
   size design=16;
   model resolution=3;
   output out=OrthArray t1 cvals=('A' 'B' 'C' 'D')
                        t2 cvals=('A' 'B' 'C' 'D')
                        t3 cvals=('A' 'B' 'C' 'D');
run;

data _null_;
   array t{3} $ t1-t3;
   array s{4} $ s1-s4; /* Buffer for holding each row      */
   file print;         /* Direct printing to output screen */
   do square=1 to 3;
      put "Square " square ":";
      n = 1;
      do r=1 to 4;
         do c=1 to 4;
            set OrthArray point=n; n=n+1;
            s{c}=t{square};
         end;
         put "       " s1-s4;
      end;
      put;
   end;
   stop;
run;
```

In most cases, the form that appears in the output data set OrthArray is the most useful. The form that usually appears in textbooks is displayed in Output 8.10.1, which can be produced by using a simple DATA step (not shown here).

Output 8.10.1 Hyper-Graeco-Latin Square

```
Square 1 :
       A D B C
       D A C B
       B C A D
       C B D A

Square 2 :
       A D B C
       C B D A
       D A C B
       B C A D

Square 3 :
       A D B C
       B C A D
       C B D A
       D A C B
```

Example 8.11: Resolution 4 Design with Minimum Aberration

NOTE: See *A Res IV Design with Minimum Aberration* in the SAS/QC Sample Library.

If a design has resolution 4, then you can simultaneously estimate all main effects and *some* two-factor interactions. However, not all resolution 4 designs are equivalent; you might be able to estimate more two-factor interactions with some than with others. Among all resolution 4 designs, a design that has the maximum number of estimable two-factor interactions is said to have *minimum aberration*.

For example, if you use the FACTEX procedure to generate a resolution 4 design for seven 2-level factors in 32 runs, you can estimate all main effects and 15 of the 21 two-factor interactions by using the design that is created by default. The following statements create this design and display its alias structure in Output 8.11.1:

```
proc factex;
   factors A B C D E F G;
   model resolution=4;
   size design=32;
   examine aliasing;
run;
```

Output 8.11.1 Alias Structure for Default 2^{7-2}_{IV} Design

The FACTEX Procedure

Aliasing Structure
A
B
C
D
E
F
G
A*B = F*G
A*C
A*D
A*E
A*F = B*G
A*G = B*F
B*C
B*D
B*E
C*D = E*G
C*E = D*G
C*F
C*G = D*E
D*F
E*F

In contrast, the resolution 4 design shown in Table 12.15 of Box, Hunter, and Hunter (1978) is a minimum aberration design that permits estimation of 18 two-factor interactions, three more than can be estimated with the default design. The FACTEX procedure constructs the minimum aberration design if you specify the MINABS option in the MODEL statement, as in the following statements:

```
proc factex;
   factors A B C D E F G;
   model resolution=4 / minabs;
   size design=32;
   examine aliasing;
run;
```

The alias structure for the resulting design is shown in Output 8.11.2.

Output 8.11.2 Alias Structure for Minimum Aberration 2_{IV}^{7-2} Design

The FACTEX Procedure

Aliasing Structure
A
B
C
D
E
F
G
A*B
A*C
A*D
A*E
A*F
A*G
B*C
B*D
B*E
B*F
B*G
C*D = E*F
C*E = D*F
C*F = D*E
C*G
D*G
E*G
F*G

All the designs listed in Table 12.15 of Box, Hunter, and Hunter (1978) have minimum aberration. For most of these cases, the default design constructed by the FACTEX procedure has minimum aberration—that is, the MINABS option is not required. This is important because the MINABS option forces the FACTEX procedure to check many more designs, and the search can therefore take longer to run. You can limit the search time by specifying the TIME= option in the PROC FACTEX statement. In five of the cases (2_{III}^{10-6}, 2_{IV}^{7-2}, 2_{IV}^{8-3}, 2_{IV}^{9-4}, and 2_{V}^{10-3}), the MINABS option is required to construct a design that has minimum aberration, and in two cases (2_{III}^{9-5}, 2_{IV}^{9-3}), the NOCHECK option is also required. If the FACTEX procedure is given sufficient time to run, specifying both the MINABS option and the NOCHECK option always results in a minimum aberration design. However, with the default search time of 60 seconds, there are three cases (2_{IV}^{10-5}, 2_{IV}^{10-4}, and 2_{IV}^{11-5}) for which the FACTEX procedure is unable to find the minimum aberration design, even with both the MINABS and NOCHECK options specified.

Example 8.12: Replicated Blocked Design with Partial Confounding

NOTE: See *Replicated Blocked Design with Confounding* in the SAS/QC Sample Library.

In an unreplicated blocked design, the interaction effect that is confounded with the block effect cannot be estimated. You can replicate the experiment so that a different interaction effect is confounded in each

replicate. This enables you to obtain information about an interaction effect from the replicates in which it is not confounded.

For example, consider a 2^3 design with factors A, B, and C arranged in two blocks. Suppose you decide to run four replicates of the design. By constructing the design sequentially, you can choose the effects to be estimated in each replicate depending on the interaction that is confounded with the block effect in the other replicates.

In the first replicate, you specify only that the main effects are to be estimable. The following statements generate an eight-run 2-level design arranged in two blocks:

```
proc factex;
   factors A B C;
   blocks nblocks=2;
   model est=(A B C);
   examine confounding aliasing;
   output out=Rep1 blockname=block nvals=(1 2);
run;
```

The alias structure and the confounding scheme are listed in Output 8.12.1. The highest-order interaction A*B*C is confounded with the block effect. The design, with recoded block levels, is saved in a data set named Rep1.

Output 8.12.1 Confounding Rule and Alias Structure for Replicate 1

The FACTEX Procedure

Aliasing Structure
A
B
C
A*B
A*C
B*C

If you were to analyze this replicate by itself, you could not determine whether an effect is due to A*B*C or due to the block effect. You can construct a second replicate that confounds a different interaction effect with the block effect. Because the FACTEX procedure is interactive, simply submit the following statements to generate the second replicate:

```
   model est=(A B C A*B*C);
   output out=Rep2
      blockname=block nvals=(3 4);
run;
```

The alias structure and the confounding scheme for the second replicate are listed in Output 8.12.2. The interaction A*B*C is free of any aliases, but now the two-factor interaction B*C is confounded with the block effect.

Output 8.12.2 Confounding Rule and Alias Structure for Replicate 2

The FACTEX Procedure

Aliasing Structure
A
B
C
A*B
A*C
[B] = B*C
A*B*C

To estimate the interaction B*C by using the third replicate, submit the following statements (immediately after the preceding statements):

```
    model est=(A B C A*B*C B*C);
    output out=Rep3 blockname=block nvals=(5 6);
run;
```

The alias structure and confounding rules are shown in Output 8.12.3. The interaction B*C is free of aliases, but the interaction A*C is confounded with the block effect.

Output 8.12.3 Confounding Rule and Alias Structure for Replicate 3

The FACTEX Procedure

Aliasing Structure
A
B
C
A*B
[B] = A*C
B*C
A*B*C

Finally, to estimate the interaction effect A*C by using the fourth replicate, submit the following statements:

```
    model est=(A B C A*B*C B*C A*C);
    output out=Rep4 blockname=block nvals=(7 8);
run;
```

The alias structure and confounding rules are displayed in Output 8.12.4.

Output 8.12.4 Confounding Rule and Alias Structure for Replicate 4

The FACTEX Procedure

Aliasing Structure
A
B
C
[B] = A*B
A*C
B*C
A*B*C

When combined, these four replicates provide full information about the main effects and three-quarter information about each of the interactions. The following statements combine the four replicates:

```
data Combine;
   set Rep1 Rep2 Rep3 Rep4;
run;
proc print data=Combine;
run;
```

The final design is saved in the data set Combine. A partial listing of this data set is shown in Output 8.12.5.

Output 8.12.5 Combined Design

Obs	block	A	B	C
1	1	-1	-1	-1
2	1	-1	1	1
3	1	1	-1	1
4	1	1	1	-1
5	2	-1	-1	1
6	2	-1	1	-1
7	2	1	-1	-1
8	2	1	1	1
9	3	-1	-1	1
10	3	-1	1	-1
11	3	1	-1	1
12	3	1	1	-1
13	4	-1	-1	-1
14	4	-1	1	1
15	4	1	-1	-1
16	4	1	1	1
17	5	-1	-1	1
18	5	-1	1	1
19	5	1	-1	-1
20	5	1	1	-1
21	6	-1	-1	-1
22	6	-1	1	-1
23	6	1	-1	1
24	6	1	1	1
25	7	-1	1	-1
26	7	-1	1	1
27	7	1	-1	-1
28	7	1	-1	1
29	8	-1	-1	-1
30	8	-1	-1	1
31	8	1	1	-1
32	8	1	1	1

Example 8.13: Incomplete Block Design

NOTE: See *Incomplete Block Design* in the SAS/QC Sample Library.

Several important series of balanced incomplete block designs can be derived from orthogonal factorial designs. One is the series of balanced lattices of Yates (1936); see page 396 of Cochran and Cox (1957). In a balanced lattice, the number of treatments v must be the square of a power of a prime number: $v = q^2, \quad q = p^k$, where p is a prime number. These designs are based on a complete set of $q - 1$ mutually orthogonal $q \times q$ Latin squares, which is equivalent to a resolution 3 design for $q + 1$ q-level factors in q^2 runs.

The balanced lattice designs include $q + 1$ replicates of the treatments. They are constructed by associating each treatment with a run in the factorial design, each replicate with one of the factors, and each block

with one of the q values of that factor. For example, the treatments in Block 3 within Replicate 2 are those treatments that are associated with runs for which factor 2 is set at value 3.

The following statements use this method to construct a balanced lattice design for 16 treatments in five replicates of four blocks each. The construction procedure is based on a resolution 3 design for five 4-level factors in 16 runs.

```
proc factex;
   factors x1-x5 / nlev=4;
   size design=16;
   model r=3;
   output out=a;
run;
```

In the following DATA step, the incomplete block design is built by using the design that PROC FACTEX saved in the data set a:

```
data b;
   keep Rep Block Plot t;
   array x{5} x1-x5;
   do Rep = 1 to 5;
      do Block = 1 to 4;
         Plot = 0;
         do n = 1 to 16;
            set a point=n;
            if (x{rep}=Block-1) then do;
               t = n;
               Plot = Plot + 1;
               output;
            end;
         end;
      end;
   end;
   stop;
run;
```

For each block within each replicate, the program loops through the run numbers in the factorial design and chooses those whose Repth factor is equal to Block–1. These run numbers are the treatments that go into the particular block.

The design is printed by using a DATA step. Each block of each replicate is built into the variables S1, S2, S3, and S4, and each block is printed with a PUT statement.

```
data _null_;
   array s{4} s1-s4;
   file print;
   n = 1;
   do r = 1 to 5;
      put "Replication " r 1.0 ":";
      do b = 1 to 4;
         do p = 1 to 4;
            set b point=n;
            s{Plot} = t;
            n = n+1;
         end;
         put "    Block " b 1.0 ":" (s1-s4) (3.0);
      end;
```

```
      put;
   end;
   stop;
run;
```

The ARRAY statement creates a buffer for holding each block, and the FILE statement directs the printing to output screen. The design is displayed in Output 8.13.1.

Output 8.13.1 A Balanced Lattice

```
Replication 1:
    Block 1:  1  2  3  4
    Block 2:  5  6  7  8
    Block 3:  9 10 11 12
    Block 4: 13 14 15 16

Replication 2:
    Block 1:  1  5  9 13
    Block 2:  2  6 10 14
    Block 3:  3  7 11 15
    Block 4:  4  8 12 16

Replication 3:
    Block 1:  1  6 11 16
    Block 2:  3  8  9 14
    Block 3:  4  7 10 13
    Block 4:  2  5 12 15

Replication 4:
    Block 1:  1  8 10 15
    Block 2:  3  6 12 13
    Block 3:  4  5 11 14
    Block 4:  2  7  9 16

Replication 5:
    Block 1:  1  7 12 14
    Block 2:  3  5 10 16
    Block 3:  4  6  9 15
    Block 4:  2  8 11 13
```

You can use the PLAN procedure to randomize the block design, as shown by the following statements:

```
proc plan seed=54321;
   factors Rep=5 Block=4 Plot=4 / noprint;
   output data=b out=c;
run;
proc sort;
   by Rep Block Plot;
run;
```

The variable Plot indexes the plots within each block. For a general discussion of randomizing block designs, see *SAS/STAT User's Guide*.

Finally, substitute **`set c`** for **`set b`** in the preceding DATA step. Running this DATA step creates the randomized design displayed in Output 8.13.2.

Output 8.13.2 Randomized Design

```
Replication 1:
    Block 1: 15  5  2 12
    Block 2:  3  8  9 14
    Block 3: 16  1 11  6
    Block 4:  7 10 13  4

Replication 2:
    Block 1:  2  4  3  1
    Block 2:  5  7  8  6
    Block 3:  9 11 10 12
    Block 4: 15 16 13 14

Replication 3:
    Block 1:  2 13  8 11
    Block 2: 14 12  7  1
    Block 3: 15  4  9  6
    Block 4:  5 16  3 10

Replication 4:
    Block 1: 13  1  5  9
    Block 2: 14  2 10  6
    Block 3: 11 15  3  7
    Block 4: 16 12  4  8

Replication 5:
    Block 1:  2 16  7  9
    Block 2: 15 10  8  1
    Block 3:  3 12  6 13
    Block 4:  5 11 14  4
```

Example 8.14: Design with Inner Array and Outer Array

NOTE: See *A Problem In Quality Improvement* in the SAS/QC Sample Library.

Byrne and Taguchi (1986) report the use of a fractional factorial design to investigate fitting an elastomeric connector to a nylon tube as tightly as possible. Their experiment applies the design philosophy of Genichi Taguchi, which distinguishes between control factors and noise factors. *Control factors* are typically those that the engineer is able to set under real conditions, while *noise factors* vary uncontrollably in practice (though within a predictable range).

The experimental layout consists of two designs, one for the control factors and one for the noise factors. The design for the control factors is called the *inner array*, and the design for noise factors is called the *outer array*. The outer array is replicated for each of the runs in the inner array, and a performance measure ("signal-to-noise ratio") is computed over the replicate. The performance measure thus reflects variation due to changes in the noise factors. You can construct such a crossproduct design by using the replication options in the OUTPUT statement of the FACTEX procedure, as shown in this example.

Researchers identified the following four control factors that were thought to influence the amount of force required to pull the connector off the tube:

- interference (Interference), defined as the difference between the outer width of the tubing and the inner width of the connector

- connector wall thickness (ConnectorWall)
- depth of insertion (InsertDepth) of the tubing into the connector
- amount of adhesive (Glue) in the connector before dipping

Researchers also identified the following three noise factors related to the assembly:

- amount of time (Time) allowed for assembly
- temperature (Temperature)
- relative humidity (Humidity)

Three levels were selected for each of the control factors, and two levels were selected for each of the noise factors.

The following statements construct the 72-run design used by Byrne and Taguchi (1986). First, an eight-run outer array for the three noise factors is created and saved in the data set OuterArray.

```
proc factex;
   factors Time Temperature Humidity;
   output out=OuterArray Time        nvals=(  24  120)
                         Temperature nvals=(  72  150)
                         Humidity    nvals=(0.25 0.75);
run;
```

Next, a nine-run inner array (design of resolution 3) is chosen for the control factors. The POINTREP= option in the OUTPUT statement replicates the eight-run outer array in the data set OuterArray for each of the nine runs in the inner array, and the final design (which contains 72 runs) is saved in the data set Design.

```
proc factex;
   factors Interference ConnectorWall InsertDepth Glue /
      nlev=3;
   size design=9;
   model resolution=3;
   output out=Design pointrep=OuterArray
        Interference  cvals=('Low'    'Medium' 'High'  )
        ConnectorWall cvals=('Thin'   'Medium' 'Thick' )
        InsertDepth   cvals=('Shallow' 'Deep' 'Medium')
        Glue          cvals=('Low'    'High'   'Medium');
run;
```

The final design is listed in Output 8.14.1. Main effects of each factor can be estimated free of each other, but they are confounded with two-factor interactions.

Output 8.14.1 Design for Control Factor and Noise Factors

Obs	Interference	ConnectorWall	InsertDepth	Glue	Time	Temperature	Humidity
1	Low	Thin	Shallow	Low	24	72	0.25
2	Low	Thin	Shallow	Low	24	72	0.75
3	Low	Thin	Shallow	Low	24	150	0.25
4	Low	Thin	Shallow	Low	24	150	0.75
5	Low	Thin	Shallow	Low	120	72	0.25
6	Low	Thin	Shallow	Low	120	72	0.75
7	Low	Thin	Shallow	Low	120	150	0.25
8	Low	Thin	Shallow	Low	120	150	0.75
9	Low	Medium	Medium	Medium	24	72	0.25
10	Low	Medium	Medium	Medium	24	72	0.75
11	Low	Medium	Medium	Medium	24	150	0.25
12	Low	Medium	Medium	Medium	24	150	0.75
13	Low	Medium	Medium	Medium	120	72	0.25
14	Low	Medium	Medium	Medium	120	72	0.75
15	Low	Medium	Medium	Medium	120	150	0.25
16	Low	Medium	Medium	Medium	120	150	0.75
17	Low	Thick	Deep	High	24	72	0.25
18	Low	Thick	Deep	High	24	72	0.75
19	Low	Thick	Deep	High	24	150	0.25
20	Low	Thick	Deep	High	24	150	0.75
21	Low	Thick	Deep	High	120	72	0.25
22	Low	Thick	Deep	High	120	72	0.75
23	Low	Thick	Deep	High	120	150	0.25
24	Low	Thick	Deep	High	120	150	0.75
25	Medium	Thin	Medium	High	24	72	0.25
26	Medium	Thin	Medium	High	24	72	0.75
27	Medium	Thin	Medium	High	24	150	0.25
28	Medium	Thin	Medium	High	24	150	0.75
29	Medium	Thin	Medium	High	120	72	0.25
30	Medium	Thin	Medium	High	120	72	0.75
31	Medium	Thin	Medium	High	120	150	0.25
32	Medium	Thin	Medium	High	120	150	0.75
33	Medium	Medium	Deep	Low	24	72	0.25
34	Medium	Medium	Deep	Low	24	72	0.75
35	Medium	Medium	Deep	Low	24	150	0.25
36	Medium	Medium	Deep	Low	24	150	0.75
37	Medium	Medium	Deep	Low	120	72	0.25
38	Medium	Medium	Deep	Low	120	72	0.75
39	Medium	Medium	Deep	Low	120	150	0.25
40	Medium	Medium	Deep	Low	120	150	0.75
41	Medium	Thick	Shallow	Medium	24	72	0.25
42	Medium	Thick	Shallow	Medium	24	72	0.75
43	Medium	Thick	Shallow	Medium	24	150	0.25
44	Medium	Thick	Shallow	Medium	24	150	0.75
45	Medium	Thick	Shallow	Medium	120	72	0.25
46	Medium	Thick	Shallow	Medium	120	72	0.75
47	Medium	Thick	Shallow	Medium	120	150	0.25
48	Medium	Thick	Shallow	Medium	120	150	0.75

Output 8.14.1 *continued*

Obs	Interference	ConnectorWall	InsertDepth	Glue	Time	Temperature	Humidity
49	High	Thin	Deep	Medium	24	72	0.25
50	High	Thin	Deep	Medium	24	72	0.75

Obs	Interference	ConnectorWall	InsertDepth	Glue	Time	Temperature	Humidity
51	High	Thin	Deep	Medium	24	150	0.25
52	High	Thin	Deep	Medium	24	150	0.75
53	High	Thin	Deep	Medium	120	72	0.25
54	High	Thin	Deep	Medium	120	72	0.75
55	High	Thin	Deep	Medium	120	150	0.25
56	High	Thin	Deep	Medium	120	150	0.75
57	High	Medium	Shallow	High	24	72	0.25
58	High	Medium	Shallow	High	24	72	0.75
59	High	Medium	Shallow	High	24	150	0.25
60	High	Medium	Shallow	High	24	150	0.75
61	High	Medium	Shallow	High	120	72	0.25
62	High	Medium	Shallow	High	120	72	0.75
63	High	Medium	Shallow	High	120	150	0.25
64	High	Medium	Shallow	High	120	150	0.75
65	High	Thick	Medium	Low	24	72	0.25
66	High	Thick	Medium	Low	24	72	0.75
67	High	Thick	Medium	Low	24	150	0.25
68	High	Thick	Medium	Low	24	150	0.75
69	High	Thick	Medium	Low	120	72	0.25
70	High	Thick	Medium	Low	120	72	0.75
71	High	Thick	Medium	Low	120	150	0.25
72	High	Thick	Medium	Low	120	150	0.75

Note that the levels of InsertDepth and Glue are listed in the OUTPUT statement in a nonstandard order so that the design produced by the FACTEX procedure matches the design of Byrne and Taguchi (1986). The order of assignment of levels does not affect the properties of the resulting design. Furthermore, the design can be randomized by specifying the RANDOMIZE option in the OUTPUT statement.

Byrne and Taguchi (1986) indicate that a smaller outer array with only four runs would have been sufficient. You can generate this design (not shown here) by modifying the statements in this example; specifically, add the following SIZE and MODEL statements:

```
size design=4;
model resolution=3;
```

In their analysis of the data from the experiment based on the smaller design, Byrne and Taguchi (1986) note several interesting interactions between control and noise factors. However, because the inner array is of resolution 3, it is impossible to say whether interesting interactions exist between the control factors. In other words, you cannot determine whether an effect is due to an interaction or to the main effect with which it is confounded.

One alternative is to begin with a design of resolution 4. Two-factor interactions remain confounded with one another, but they are free of main effects. Moreover, further experimentation can be carried out to distinguish

between confounded interactions that seem important. To determine the optimal size of this design, submit the following statements interactively:

```
proc factex;
   factors Interference ConnectorWall InsertDepth Glue /
      nlev=3;
   model resolution=4;
   size design=minimum;
run;
```

This causes the following message to appear in the SAS log:

```
NOTE: Design has 27 runs, resolution = 4.
```

In other words, the smallest resolution 4 design for four 3-level factors has 27 runs, which together with the eight-run outer array requires 216 runs. Even the smaller four-run outer array requires 108 runs. Both of these designs are substantially larger than the design originally reported, but the larger designs protect against the effects of unsuspected interactions.

A second alternative is to begin with only two levels of the control factors. Further experimentation can then be directed toward exploring the effects of factors that are determined to be important in this initial stage of experimentation. Submit the following additional statements (NLEV=2 is the default in the FACTORS statement):

```
   factors Interference ConnectorWall InsertDepth Glue;
   model resolution=4;
   size design=minimum;
run;
```

This causes the following message to appear in the SAS log:

```
NOTE: Design has 8 runs, resolution = 4.
```

Thus, as few as eight runs can be used for the inner array. This design is amenable to blocking, whereas the proposed nine-run design is not. Blocking is an important consideration whenever experimental conditions can vary over the course of conducting the experiment.

Now, submit the following statements:

```
   size design=8;
   blocks size=minimum;
run;
```

This causes the following message to appear in the SAS log:

```
NOTE: Design has 8 runs in 4 blocks of size 2,
      resolution = 4.
```

Thus the experiment can be run in blocks as small as two runs.

Example 8.15: Fractional Factorial Split-Plot Designs

NOTE: See *Fractional Factorial Split-Plot Design* in the SAS/QC Sample Library.

In split-plot designs, not all factor levels can change from plot to plot. In the simplest split-plot structure, runs are grouped into whole plots; certain factors (*whole-plot factors*) are applied to all plots in the whole plot,

and others (*subplot factors*) are applied to individual plots within a whole plot. Split-plot designs are very common in chemical and process industries, where factors of interest are often applied at different stages of the production process and the final measurements of interest are made on the finished product. In this case, the different stages of production might give rise to multiple whole-plot effects.

Suppose you are designing an experiment to measure six factors that affect characteristics of metal wires that are sheathed with a certain material. Three of the factors (W1, W2, W3) apply to how the wires themselves are made, and the other three (S1, S2, S3) apply to the sheathing material. You propose to first prepare eight different batches of wire, making two wires from each batch, and then to prepare the sheathing material for each wire individually. This describes a standard split-plot experiment, in which batches of wires form whole plots and sheathed wires form subplots. The following code constructs a resolution 4 design for this experiment, specifying the Wire unit effect in the BLOCKS statement, and then in the UNITEFFECT statement specifying that W1, W2, and W3 should be constant within Wire and that S1, S2, and S3 should change within Wire. The resulting design is printed, sorted by Wire.

```
proc factex;
   factors W1 W2 W3
           S1 S2 S3;
   size design=16;
   blocks units=(Wire=8);
   model r=4;
   uniteffect Wire / whole=(W1 W2 W3)
                     sub  =(S1 S2 S3);
   examine aliasing(units);
   output out=WireExperiment1;
run;

proc sort data=WireExperiment1;
   by Wire W1-W3 S1-S3;
run;
proc print data=WireExperiment1;
run;
```

Output 8.15.1 shows the aliasing structure for the design, which indicates that the main effects of the wire factors are indeed estimated on the Wire whole plots and the main effects of the sheath factors are estimated on the subplots. Interestingly, some of the sheath factor interactions are also confounded with whole plots.

Output 8.15.1 A Split-Plot Design

The FACTEX Procedure

Aliasing Structure	
Units	
Wire	W1
Wire	W2
Wire	W3
Wire	W1*W2 = S1*S2
Wire	W1*W3 = S1*S3
Wire	W2*W3 = S2*S3
Residual	S1
Residual	S2
Residual	S3
Residual	W1*S1 = W2*S2 = W3*S3
Residual	W1*S2 = W2*S1
Residual	W1*S3 = W3*S1
Residual	W2*S3 = W3*S2

The final design is listed in Output 8.15.2. Notice that the factors W1, W2, and W3 are constant within Wire, whereas S1, S2, and S3 change within Wire.

Output 8.15.2 A Split-Plot Design

Obs	_1_	_2_	_3_	_4_	W1	W2	W3	S1	S2	S3	Wire
1	-1	-1	-1	1	-1	1	1	-1	1	1	1
2	-1	-1	-1	-1	-1	1	1	1	-1	-1	1
3	-1	-1	1	-1	1	-1	-1	-1	1	1	2
4	-1	-1	1	1	1	-1	-1	1	-1	-1	2
5	-1	1	-1	-1	1	-1	1	-1	1	-1	3
6	-1	1	-1	1	1	-1	1	1	-1	1	3
7	-1	1	1	1	-1	1	-1	-1	1	-1	4
8	-1	1	1	-1	-1	1	-1	1	-1	1	4
9	1	-1	-1	-1	1	1	-1	-1	-1	1	5
10	1	-1	-1	1	1	1	-1	1	1	-1	5
11	1	-1	1	1	-1	-1	1	-1	-1	1	6
12	1	-1	1	-1	-1	-1	1	1	1	-1	6
13	1	1	-1	1	-1	-1	-1	-1	-1	-1	7
14	1	1	-1	-1	-1	-1	-1	1	1	1	7
15	1	1	1	-1	1	1	1	-1	-1	-1	8
16	1	1	1	1	1	1	1	1	1	1	8

To see why the Wire factors are constant within wire and the sheath factors change, examine the confounding rules for the design. The following statements produce the table of confounding rules listed in Output 8.15.3:

```
proc factex;
   factors W1 W2 W3
           S1 S2 S3;
   size design=16;
```

```
   blocks units=(Wire=8);
   model r=4;
   uniteffect Wire / whole=(W1 W2 W3)
                     sub  =(S1 S2 S3);
   examine confounding;
run;
```

Output 8.15.3 Split-Plot Confounding Rules

The FACTEX Procedure

Factor Confounding Rules
W1 = [1]*[2]*[3]
W2 = [2]*[3]
W3 = [1]*[3]
S1 = [1]*[2]*[3]*[4]
S2 = [2]*[3]*[4]
S3 = [1]*[3]*[4]

The terms [*i*] on the right-hand side of these rules denote plot-indexing pseudofactors, as discussed in the section "Split-Plot Designs" on page 655. Note that the wire factors W1, W2, and W3 are confounded only with interactions between the first three pseudofactors, the ones identified with the eight levels of the Wire unit factor. This guarantees that these factors are constant within levels of Wire. By contrast, the confounding rules for the sheath factors S1, S2, and S3 each involve the fourth pseudofactor, so they must change within levels of Wire.

There are only eight different combinations of the sheath factors, but the previous design requires you to produce batches of sheath material 16 times, once for each of the two wires to be made from each wire batch. If instead you propose to make just four batches of sheath material and apply part of each batch to parts of different batches of wires, the design becomes a row-column design instead of a split-plot design. Furthermore, suppose that the number of batches rather than the size of each batch is the main cost, so that you can prepare eight batches of wire and four batches of sheathing material in sufficient quantity to make 64 different sheathed wires. Because there can be only four different combinations of the three sheathing factors, each sheathing factor interaction is aliased with a main effect, and thus the design necessarily has resolution 3. All other interactions are estimable free of main effects. The following statements create the design and display the two unit effects with their respective whole-unit factor levels:

```
proc factex;
   factors W1 W2 W3
           S1 S2 S3;
   size design=64;
   blocks units=(Wire=8 Sheath=4);
   model r=3;
   uniteffect Wire   / whole=(W1 W2 W3);
   uniteffect Sheath / whole=(S1 S2 S3);
   examine aliasing(units);
   output out=WireExperiment2;
proc freq data=WireExperiment2;
   table Wire  *W1*W2*W3 / list nocum nopct;
   table Sheath*S1*S2*S3 / list nocum nopct;
run;
```

The results, listed in Output 8.15.4 and Output 8.15.5, indicate that W1, W2, and W3 are constant within Wire and S1, S2, and S3 are constant within Sheath.

Output 8.15.4 A Split-Lot Design: Wire Units

The FREQ Procedure

Wire	W1	W2	W3	Frequency
1	-1	1	1	8
2	1	-1	-1	8
3	1	-1	1	8
4	-1	1	-1	8
5	1	1	-1	8
6	-1	-1	1	8
7	-1	-1	-1	8
8	1	1	1	8

Output 8.15.5 A Split-Lot Design: Sheath Units

The FREQ Procedure

Sheath	S1	S2	S3	Frequency
1	1	-1	-1	16
2	-1	1	-1	16
3	-1	-1	1	16
4	1	1	1	16

Example 8.16: Design for a Three-Step Process

NOTE: See *A Design for a Three-Step Process* in the SAS/QC Sample Library.

Ramirez and Weisz (2009) discuss an experiment on a multistep milling process that has 16 processing factors, with a single factor applied at the first stage, seven more factors at the second stage, and eight more at the final stage. The experiment involves eight first-stage runs, eight second-stage runs within each of those, and again, two to four third-stage runs within each of those, for a total of 128 to 256 total experimental units. This example explores several different ways to design this experiment, depending on what types of effects are most important.

The following statements request a design of maximum resolution for this split-plot structure.

```
%let F1 = Z;
%let F2 = A B C D E F G;
%let F3 = P Q R S T U V W;
proc factex;
   factors &F1 &F2 &F3;
   model r=max;
   size design=128;
   blocks units=(Step1=8 Step2=8);
   uniteffect Step1       / whole=(&F1) sub=(&F2 &F3);
   uniteffect Step1*Step2 / whole=(&F2) sub=(    &F3);
```

```
   examine aliasing(units) summary;
quit;
```

The factors are listed in macro variables, for ease in specifying them in UNITEFFECT statements. The BLOCKS statement defines the unit factors for the first two processing stages, with eight runs of each. The two UNITEFFECT statements then use these unit factors to specify which unit effects correspond to which factors. Finally, the EXAMINE statement requests that the aliasing structure and the overall modeling summary be displayed to see how many effects of different orders are estimable and clear. The UNITS suboption of the ALIASING option includes the unit effect confounding for each alias string in the alias structure.

The resulting design has resolution 4, which means that main effects are clear of two-factor interactions but interactions are aliased with each other. Output 8.16.1 shows which interactions are aliased and also shows which units are used to estimate them. Note that several interactions between Step2 and Step3 factors are estimated with Step2 units.

Output 8.16.1 Aliasing for Default 128-Run Three-Step Design

The FACTEX Procedure

Aliasing Structure	
Units	
Step1	Z
Step1	A*B = C*D = E*F = P*Q = R*S = T*U = V*W
Step1	A*C = B*D = E*G = P*R = Q*S = T*V = U*W
Step1	A*D = B*C = F*G = P*S = Q*R = T*W = U*V
Step1*Step2	A
Step1*Step2	B
Step1*Step2	C
Step1*Step2	D
Step1*Step2	E
Step1*Step2	F
Step1*Step2	G
Step1*Step2	Z*A
Step1*Step2	Z*B
Step1*Step2	Z*C
Step1*Step2	Z*D
Step1*Step2	Z*E
Step1*Step2	Z*F
Step1*Step2	Z*G
Step1*Step2	A*E = B*F = C*G = P*T = Q*U = R*V = S*W
Step1*Step2	A*F = B*E = D*G = P*U = Q*T = R*W = S*V
Step1*Step2	A*G = C*E = D*F = P*V = Q*W = R*T = S*U
Step1*Step2	B*G = C*F = D*E = P*W = Q*V = R*U = S*T
Residual	P
Residual	Q
Residual	R
Residual	S
Residual	T
Residual	U
Residual	V
Residual	W
Residual	Z*P
Residual	Z*Q
Residual	Z*R
Residual	Z*S
Residual	Z*T
Residual	Z*U
Residual	Z*V
Residual	Z*W
Residual	A*P = B*Q = C*R = D*S = E*T = F*U = G*V
Residual	A*Q = B*P = C*S = D*R = E*U = F*T = G*W
Residual	A*R = B*S = C*P = D*Q = E*V = F*W = G*T
Residual	A*S = B*R = C*Q = D*P = E*W = F*V = G*U
Residual	A*T = B*U = C*V = D*W = E*P = F*Q = G*R
Residual	A*U = B*T = C*W = D*V = E*Q = F*P = G*S

Output 8.16.1 *continued*

The FACTEX Procedure

Aliasing Structure	
Units	
Residual	A*V = B*W = C*T = D*U = E*R = F*S = G*P
Residual	A*W = B*V = C*U = D*T = E*S = F*R = G*Q

As Output 8.16.2 shows, only 30/120 = 25% of the two-factor interactions (2FI) are estimable and only 15/120 = 13% of them are clear.

Output 8.16.2 Modeling Summary for Default 128-Run Three-Step Design

Modeling Summary		
	Effects	
	Main	2FI
Total	16	120
Estimable	16	30
Clear	16	15

If simply protecting the main-effects estimates against potential two-factor interactions is sufficient, then this design suffices. However, if you want to estimate as many of the two-factor interactions as possible, then you should look for a MaxClear design. The following statements use the MAXCLEAR option in the MODEL statement to request a MaxClear design, and they also use the ORDER=RANDOM(RESTART) option in the PROC FACTEX statement to improve the chances that the best design is found. For more information about MaxClear designs, see the section "MaxClear Designs" on page 655.

```
%let F1 = Z;
%let F2 = A B C D E F G;
%let F3 = P Q R S T U V W;
proc factex order=random(restart seed=1);
   factors &F1 &F2 &F3;
   model r=max / maxclear;
   size design=128;
   blocks units=(Step1=8 Step2=8);
   uniteffect Step1       / whole=(&F1) sub=(&F2 &F3);
   uniteffect Step1*Step2 / whole=(&F2) sub=(    &F3);
   examine summary;
quit;
```

The modeling summary results for the MaxClear design are shown in Output 8.16.3. Now 87/120 = 73% of the 2FI are estimable and 69/120 = 58% of them clear.

Output 8.16.3 Modeling Summary for MaxClear 128-Run Three-Step Design

The FACTEX Procedure

Modeling Summary	Effects	
	Main	2FI
Total	16	120
Estimable	16	87
Clear	16	69

This is a great improvement over the default design, but more than 128 runs are necessary if complete estimability of all two-factor interactions is required. The following statements construct a design in 256 runs, effectively doubling the number of third-stage runs from two to four:

```
%let F1 = Z;
%let F2 = A B C D E F G;
%let F3 = P Q R S T U V W;
proc factex;
   factors &F1 &F2 &F3;
   model r=max;
   size design=256;
   blocks units=(Step1=8 Step2=8);
   uniteffect Step1       / whole=(&F1) sub=(&F2 &F3);
   uniteffect Step1*Step2 / whole=(&F2) sub=(    &F3);
   examine aliasing(units);
quit;
```

The aliasing structure (not shown) shows that the resulting design has resolution 5, which means that all main effects and two factor interactions are estimable free of each other. Even though the required 256 runs mean that this is a relatively large experiment, they are still only a tiny fraction of the 65,536 runs required for a complete factorial design.

Example 8.17: Strip-Split-Split-Plot Design

NOTE: See *A Strip-Split-Split-Plot Design* in the SAS/QC Sample Library.

Suppose you are designing an experiment for a three-step process that runs on different machines. One way to model this is with a row × column strip-split-split-plot structure, with one type of unit, Machine, crossed with a process that has a split-split-plot structure. The following statements create a resolution 4 design in 11 factors for this situation, with one Machine factor (MSetting) and three, three, and five whole plot, split-plot, and split-split-plot process factors, respectively. The statements also request that the design's aliasing structure and modeling summary be displayed, with the unit effect confounding for each alias string included in the alias structure.

```
%let FR = X11-X13;
%let FC = X21-X23;
%let FX = X31-X35;
proc factex;
   factors MSetting &FR &FC &FX;
   model r=4;
```

```
   blocks units=(Machine=2 Step1=8 Step2=4 Step3=2);
   uniteffect Machine           / whole=(MSetting);
   uniteffect Step1             / whole=(&FR) sub=(&FC &FX);
   uniteffect Step1*Step2       / whole=(&FC) sub=(    &FX);
   uniteffect Step1*Step2*Step3 / whole=(&FX);
   size design=128;
   examine aliasing(units) summary;
run;
```

The UNITEFFECT statements define a triply nested split-plot structure for the process on each machine, including the Step1*Step2*Step3 split-split units for the process, in order to ensure that process effects are crossed with Machine.

As Output 8.17.1 shows, 36/66 = 55% of the 2FI are estimable and 21/66 = 32% of them are clear. The aliasing structure (not shown) indicates that the main effect of MSetting is the only thing that is estimated with the Machine units; all interactions between MSetting and the process factors are estimated with the experimental units, labeled "Residual" in the alias structure.

Output 8.17.1 A Strip-Split-Split-Plot Design

The FACTEX Procedure

Modeling Summary		
	Effects	
	Main	2FI
Total	12	66
Estimable	12	36
Clear	12	21

If simply protecting the main-effects estimates against potential two-factor interactions is the reason for requiring a resolution 4 design, then the design of Output 8.17.1 suffices. However, if you want to estimate as many of the two-factor interactions as possible, then you should use the MAXCLEAR option in the MODEL statement to construct a MaxClear design, as shown in the following statements:

```
%let FR = X11-X13;
%let FC = X21-X23;
%let FX = X31-X35;
proc factex order=random(restart seed=230501);
   factors MSetting &FR &FC &FX;
   model r=4 / maxclear;
   blocks units=(Machine=2 Step1=8 Step2=4 Step3=2);
   uniteffect Machine           / whole=(MSetting);
   uniteffect Step1             / whole=(&FR) sub=(&FC &FX);
   uniteffect Step1*Step2       / whole=(&FC) sub=(    &FX);
   uniteffect Step1*Step2*Step3 / whole=(&FX);
   size design=128;
   examine summary;
run;
```

As Output 8.17.2 shows, now 55/66 = 83% of the 2FI are estimable and 45/66 = 68% of them are clear—more than twice as many clear interactions as before.

Output 8.17.2 A Strip-Split-Split-Plot Design

The FACTEX Procedure

Modeling Summary	Effects	
	Main	2FI
Total	12	66
Estimable	12	55
Clear	12	45

For more information about MaxClear designs, see the section "MaxClear Designs" on page 655.

Example 8.18: Design and Analysis of a Complete Factorial Experiment

NOTE: See *Complete Factorial Experiment* in the SAS/QC Sample Library.

Yin and Jillie (1987) describe an experiment on a nitride etch process for a single-wafer plasma etcher. The experiment has four factors: cathode power (Power), gas flow (Flow), reactor chamber pressure (Pressure), and electrode gap (Gap). A single replicate of a 2^4 design is run, and the etch rate (Rate) is measured. You can use the following statements to construct a 16-run design in the four factors:

```
proc factex;
   factors Power Flow Pressure Gap;
   output out=EtcherDesign
          Power    nvals=(0.80 1.20)
          Flow     nvals=(4.50 550)
          Pressure nvals=(125  200)
          Gap      nvals=(275  325);
run;
```

The design that includes the actual (decoded) factor levels is saved in the data set EtcherDesign. The experiment that uses the 16-run design is performed, and the etch rate is measured. The following DATA step updates the data set EtcherDesign with the values of Rate:

```
data EtcherDesign;
   set EtcherDesign;
   input Rate @@;
   datalines;
   550   669  604   650  633   642  601   635
   1037  749  1052  868  1075  860  1063  729
   ;

title 'Nitride Etch Process Experiment';
proc print;
run;
```

The data set EtcherDesign is listed in Output 8.18.1.

Output 8.18.1 A 2^4 Design with Responses

Nitride Etch Process Experiment

Obs	Power	Flow	Pressure	Gap	Rate
1	0.8	4.5	125	275	550
2	0.8	4.5	125	325	669
3	0.8	4.5	200	275	604
4	0.8	4.5	200	325	650
5	0.8	550.0	125	275	633
6	0.8	550.0	125	325	642
7	0.8	550.0	200	275	601
8	0.8	550.0	200	325	635
9	1.2	4.5	125	275	1037
10	1.2	4.5	125	325	749
11	1.2	4.5	200	275	1052
12	1.2	4.5	200	325	868
13	1.2	550.0	125	275	1075
14	1.2	550.0	125	325	860
15	1.2	550.0	200	275	1063
16	1.2	550.0	200	325	729

To perform an analysis of variance on the responses, you can use the GLM procedure, as follows:

```
proc glm data=EtcherDesign;
   class Power Flow Pressure Gap;
   model rate=Power|Flow|Pressure|Gap@2 / ss1;
run;
```

The factors are listed in both the CLASS and MODEL statements, and the response as a function of the factors is modeled by using the MODEL statement. The MODEL statement requests Type I sum of squares (SS1) and lists all effects that contain two or fewer factors. It is assumed that three-factor and higher interactions are not significant.

Part of the output from the GLM procedure is shown in Output 8.18.2. The main effect of the factors Power and Gap and the interaction between Power and Gap are significant (their p-values are less than 0.01).

Output 8.18.2 Analysis of Variance for the Nitride Etch Process Experiment

Nitride Etch Process Experiment

The GLM Procedure

Dependent Variable: Rate

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Power	1	374850.0625	374850.0625	183.99	<.0001
Flow	1	217.5625	217.5625	0.11	0.7571
Power*Flow	1	18.0625	18.0625	0.01	0.9286
Pressure	1	10.5625	10.5625	0.01	0.9454
Power*Pressure	1	1.5625	1.5625	0.00	0.9790
Flow*Pressure	1	7700.0625	7700.0625	3.78	0.1095
Gap	1	41310.5625	41310.5625	20.28	0.0064
Power*Gap	1	94402.5625	94402.5625	46.34	0.0010
Flow*Gap	1	2475.0625	2475.0625	1.21	0.3206
Pressure*Gap	1	248.0625	248.0625	0.12	0.7414

References

Addelman, S. (1962). "Orthogonal Main-Effects Plans for Asymmetrical Factorial Experiments." *Technometrics* 4:21–46.

Bose, R. C. (1947). "Mathematical Theory of the Symmetrical Factorial Design." *Sankhyā* 8:107–166.

Box, G. E. P., Hunter, W. G., and Hunter, J. S. (1978). *Statistics for Experimenters*. New York: John Wiley & Sons.

Butler, N. A. (2004). "Construction of Two-Level Split-Plot Fractional Factorial Designs for Multistage Processes." *Technometrics* 46:445–451.

Byrne, D. M., and Taguchi, S. (1986). "The Taguchi Approach to Parameter Designs." *Quality Congress Transactions* 177:168–177.

Chakravarti, I. M. (1956). "Fractional Replication in Asymmetrical Factorial Designs and Partially Balanced Arrays." *Sankhyā* 17:143–164.

Cochran, W. G., and Cox, G. M. (1957). *Experimental Designs*. 2nd ed. New York: John Wiley & Sons.

Dehnad, K., ed. (1989). *Quality Control, Robust Design, and Taguchi Method*. Pacific Grove, CA: Wadsworth & Brooks/Cole.

Fries, A., and Hunter, W. G. (1980). "Minimum Aberration 2^{k-p} Designs." *Technometrics* 22:601–608.

Huang, P., Chen, D., and Voelkel, J. O. (1998). "Minimum-Aberration Two-Level Split-Plot Designs." *Technometrics* 40:314–326.

Kempthorne, O. (1975). *The Design and Analysis of Experiments*. Huntington, NY: Robert E. Krieger Publishing.

Margolin, B. H. (1967). "Systematic Methods of Analyzing $2^n \times 3^m$ Factorial Experiments with Applications." *Technometrics* 11:431–444.

Montgomery, D. C. (1991). *Design and Analysis of Experiments*. 3rd ed. New York: John Wiley & Sons.

Phadke, M. (1989). *Quality Engineering Using Robust Design*. Englewood Cliffs, NJ: Prentice-Hall.

Ramirez, J. G., and Weisz, J. T. (2009). "Designing Multi-step Fractional Factorial Split-Plots: A Combined JMP and SAS User Application." In *Proceedings of the SAS Global Forum 2009 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings09/254-2009.pdf.

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

Williams, E. J. (1949). "Experimental Designs Balanced for the Estimation of Residual Effects of Treatments." *Australian Journal of Scientific Research, Series A* 2:149–168.

Wu, C. F. J., and Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. New York: John Wiley & Sons.

Yates, F. (1936). "Incomplete Randomized Blocks." *Annals of Eugenics* 7:121–140.

Yin, G. Z., and Jillie, D. W. (1987). "Orthogonal Design for Process Optimization and Its Application in Plasma Etching." *Solid State Technology* 30:127–132.

Chapter 9
The ISHIKAWA Procedure

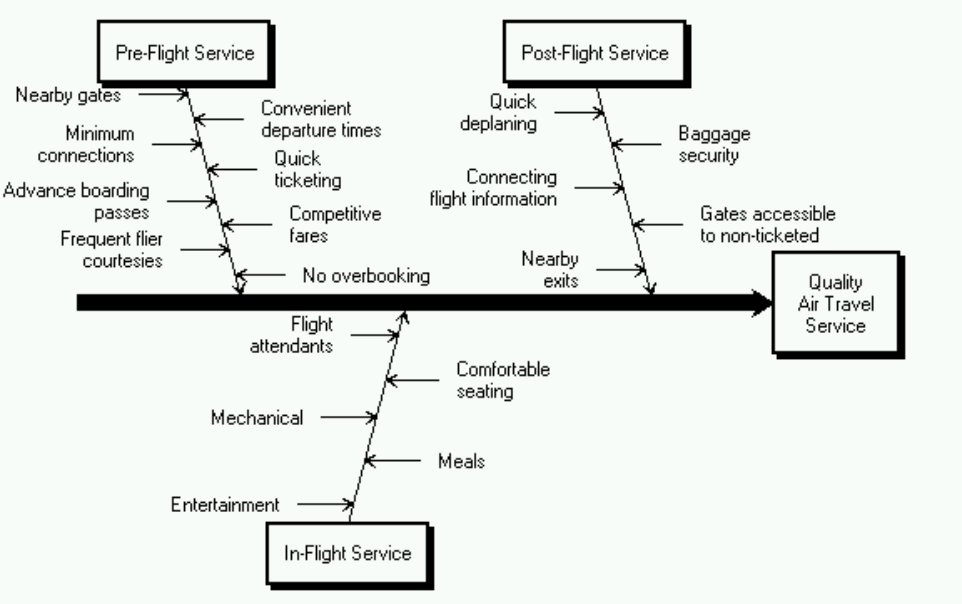
Contents

Overview **704**
Terminology 705
Getting Started: ISHIKAWA Procedure **707**
Syntax: ISHIKAWA Procedure **716**
Details: ISHIKAWA Procedure **716**
Summary of Operations 717
Operations 720
Adding Arrows 720
Labeling Arrows 724
Moving Arrows 727
Deleting Arrows 733
Resizing Arrows 736
Swapping Arrows 738
Balancing Arrows 741
Notepads 748
Managing Complexity 750
Zooming Arrows 752
Isolating Arrows 753
Merging Diagrams 755
Creating Graphics Output Using SAS/GRAPH Software 758
Creating Bitmap Graphics Output 761
Modifying Fonts 763
Modifying Box Colors 764
Modifying Arrow Colors and Line Styles 764
Modifying Text Colors 772
Modifying Arrow Heads 774
Modifying Environmental Attributes 774
Saving an Ishikawa Diagram for Future Editing 776
Reading an Existing Ishikawa Diagram 777
Displaying Multiple Ishikawa Diagrams 778
Input and Output Data Sets 782
Examples: ISHIKAWA Procedure **784**
Example 9.1: Quality of Air Travel Service 784
Example 9.2: Integrated Circuit Failures 785
Example 9.3: Photographic Development Process 786
References **786**

Overview: ISHIKAWA Procedure

The Ishikawa diagram,[1] also known as a cause-and-effect diagram or fishbone diagram, is one of the seven basic tools for quality improvement in Japanese industry. It is used to display the factors that affect a particular quality characteristic or problem. For example, the following Ishikawa diagram shows factors affecting the quality of air travel service:

Figure 9.1 Ishikawa Diagram



In this example, the factors are organized into three categories of service (Pre-flight, In-flight, and Post-flight), which are represented as branches. The factors affecting each of these areas are represented as stems.

An Ishikawa diagram is typically the result of a brainstorming session to improve a product, process, or service. The main goal is represented by a main arrow or trunk, and primary factors are represented as sub-arrows or branches. Secondary factors are then added as stems, tertiary factors as leaves, and so on.

Creating the diagram stimulates discussion and often leads to an increased understanding of a complex problem. Japanese QC Circle members often post Ishikawa diagrams in a display area where they will be accessible to managers and other groups; refer to Rodriguez (1991). In the United States, Ishikawa diagrams are often included in presentations by plant personnel to management or customers.

Traditionally, Ishikawa diagrams have been prepared by hand on paper or chalk boards. This limits the amount of detail that can be added and makes it awkward to update the diagram as an understanding of the process evolves. Manual preparation also restricts the collection and display of data on the diagram, as advocated by Ishikawa (1982).

The ISHIKAWA procedure was designed to overcome these limitations by providing a highly interactive graphics environment (referred to in this section as the *ISHIKAWA environment*) for creating and modifying Ishikawa diagrams.

[1] The Ishikawa diagram is named after its developer, Kaoru Ishikawa (1915-1989), a leader in Japanese quality control; refer to Karabatsos (1989), Kume (1985) and Sarazen (1990).

In the ISHIKAWA environment you can

- add and delete arrows with a mouse. You can also swap, copy, and so forth.
- highlight special problems or critical paths with line styles and color
- display additional data for each of the arrows in a popup notepad
- display portions of the diagram in separate windows for increasing or isolating detail. You can also divide sections of the diagram into separate Ishikawa diagrams.
- merge multiple Ishikawa diagrams into a single, master diagram
- display any number of arrows and up to ten levels of detail
- foliate and defoliate diagrams dynamically
- save diagrams for future editing
- save diagrams in graphics catalogs or export them to host clipboards or graphics files
- customize graphical features such as fonts, arrow types, and box styles
- obtain online help at any time

If you are using the ISHIKAWA procedure for the first time, the tutorial at the end of this chapter demonstrates some of the basic operations used in the ISHIKAWA procedure. A summary of these operations (and others) can be found in the section "Summary of Operations" on page 717.

For a detailed discussion of each of the operations, see "Details: ISHIKAWA Procedure" on page 716. This chapter includes many tools not presented in the tutorial.

Terminology

This section introduces basic operations used in the ISHIKAWA environment and defines terms used to describe the ISHIKAWA procedure. Some details depend on your *host*, which is the specific system of computing hardware and software you use. For example, all hosts present the ISHIKAWA environment in a system of *windows* on the host's *display*, but the appearance of your windows may differ from the figures in this book. You can find more information in the SAS companion for your host and in your host system documentation.

Using a Mouse

On most hosts you can use a *mouse* to point to objects on the display. A mouse is a physical device that controls the location of a *cursor*, which is a small, movable symbol on the display. Due to the precision required, you must use a mouse to perform tasks in the ISHIKAWA environment.

Text is placed relative to the *text* cursor and not the *mouse* cursor (↖). The mouse cursor is always visible, while the text cursor is displayed only when text can be entered (for example, when an arrow is being added).

The mouse also has *buttons* that work like keys on the keyboard. On most hosts, you *select* an object by pointing to it with the mouse and clicking the left button on the mouse. To *click*, press the button down and

release it quickly without moving the mouse. To *double click*, click *twice* quickly without moving the mouse. To *drag*, move the mouse while holding down the left mouse button.

Popup menus appear to *pop up* on the display when you press a button—usually the right mouse button. Popup menus are convenient to use, since they always appear at the cursor location. Selecting an item from the popup menu, however, is host specific.

For details about using the mouse on your system, consult the SAS companion for your host.

Using Context-Sensitive Operations

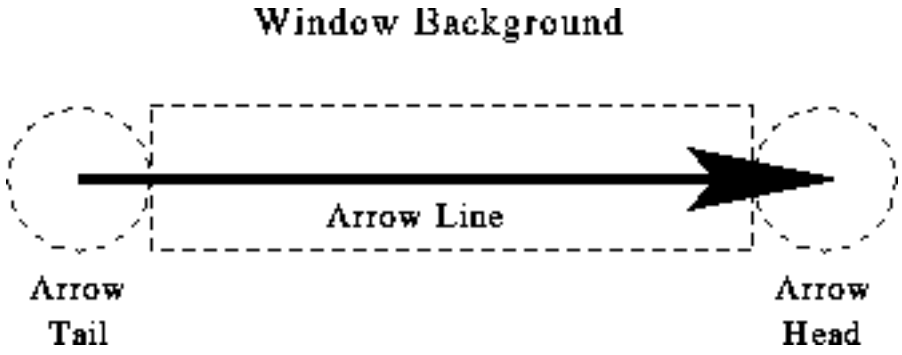
Basic operations such as add, edit, delete, and move are invoked by activating the mouse near various *hotspots* along the arrows rather than selecting tools from a tools palette. The hotspots are the following context-sensitive areas in the Ishikawa diagram:

- arrow heads, tails, lines, and labels
- window background

Given such evident features, and a rigidly defined structure, the hotspots are not highlighted.

The hotspot areas are illustrated in the following figure:

Figure 9.2 Context-Sensitive Locations (Hotspots)



The dotted, circular region at the right end of the arrow is the arrow head hotspot. Arrows attach to other arrows at the head of the arrow. The dotted, circular region at the left end of the arrow is the arrow tail hotspot. The region that encompasses the arrow line is also hot. Every arrow in the diagram has these hotspots.

The window background is any area inside the window and outside the dotted lines. You use the window background to cancel pending operations (such as adds and moves) and to control global or environment-specific operations (such as decreasing detail and tagging arrows).

When you activate the mouse, the ISHIKAWA environment uses the mouse event (click, double click, drag, or popup) and the hotspot type (head, tail, line, label, or background) to infer the intended operation. The ISHIKAWA environment responds differently depending upon which hotspot you select and how you select it. This is often referred to as *context-sensitive* behavior.

Context sensitivity allows the ISHIKAWA environment to operate without modes. In a modeless environment like the ISHIKAWA environment, context-sensitive operations reduce the amount of mouse travel (the time and distance spent moving the cursor from the drawing area to the tools palette and back). For example, you do not go to a tools palette to change from *add mode* to *delete mode*. This allows you to focus on the diagram rather than on the diagramming tool.

In the ISHIKAWA environment, the primary operations such as add, edit, delete, and move are all operations associated with a specific hotspot and the mouse button. Secondary operations such as zoom, copy, highlight, and so forth operate from *context-sensitive* popup menus (typically activated using the right mouse button.) Other, less frequently used operations are available from the command bar.

The relationship between these context-sensitive areas, the mouse actions, and the basic ISHIKAWA tools is introduced in the tutorial that follows. A comprehensive discussion of each operation is given in "Details: ISHIKAWA Procedure" on page 716. In addition, the tables in the section "Summary of Operations" on page 717 provide a good overview of how to function inside the ISHIKAWA environment.

Using the Command Bar

In addition to the editing tools, the ISHIKAWA environment provides a number of file management, printing, and help facilities. These facilities are located on the *pull-down* menu associated with the window. The appearance and location of the command bar are host specific. On most hosts, you choose these operations by *pulling down* a menu from a *menu bar* using the mouse button. For more details about using the command bar on your system, consult the SAS companion for your host.

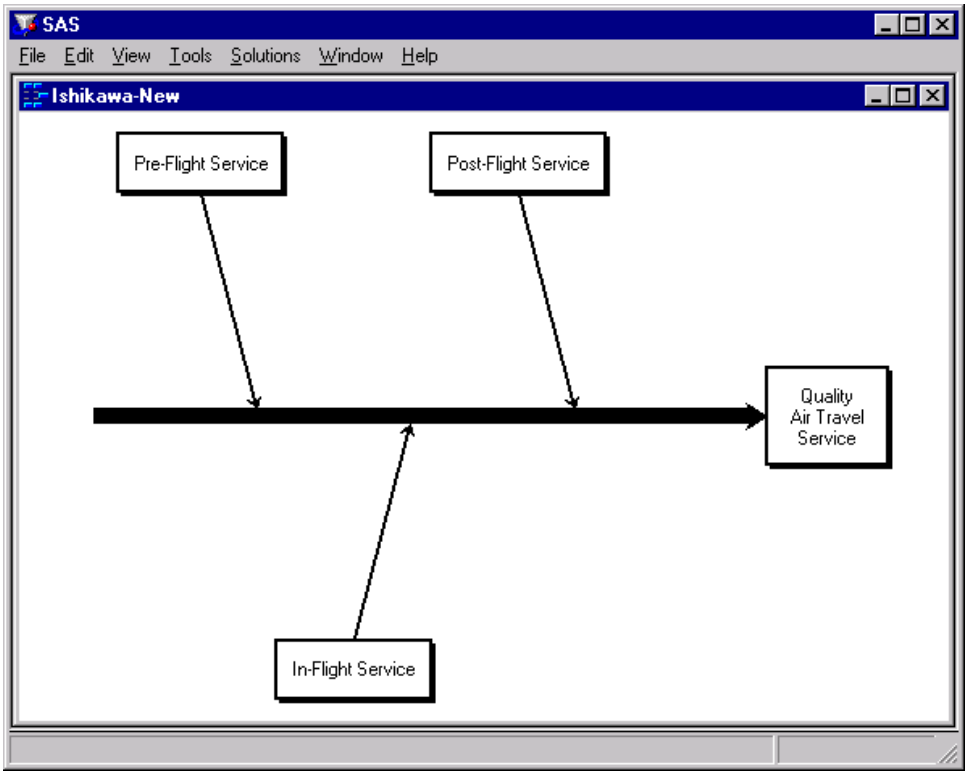
Getting Started: ISHIKAWA Procedure

NOTE: See *Airline Data* in the SAS/QC Sample Library.

The following example is used throughout the ISHIKAWA chapters. Later examples illustrate how to add to and modify this diagram. If you are not familiar with the ISHIKAWA procedure, you may want to complete this tutorial before proceeding to "Details: ISHIKAWA Procedure" on page 716. In this tutorial you will learn to create and save a simple Ishikawa diagram.

A task force is studying ways to improve the quality of passenger service for a major airline. After a preliminary discussion, the team concludes that three major areas should be considered: pre-flight service, in-flight service, and post-flight service. This result is to be displayed with the following preliminary Ishikawa diagram:

Figure 9.3 Preliminary Ishikawa Diagram

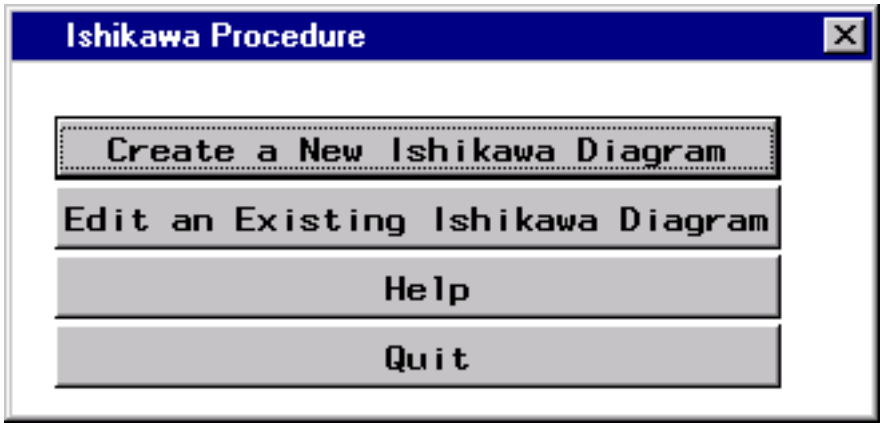


1. To begin using the ISHIKAWA environment, submit the following SAS statements:

```
proc ishikawa;
run;
```

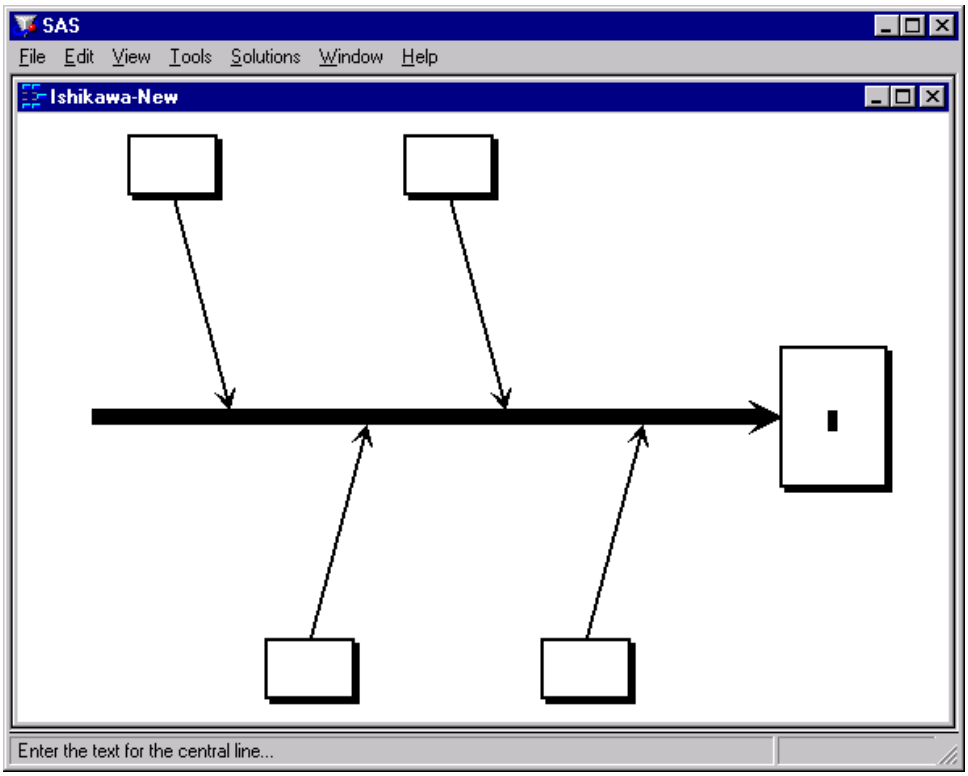
An initial menu appears on your display, as follows:

Figure 9.4 Initial Menu



2. Select **Create a New Ishikawa Diagram** to open a window containing a template for a new Ishikawa diagram.

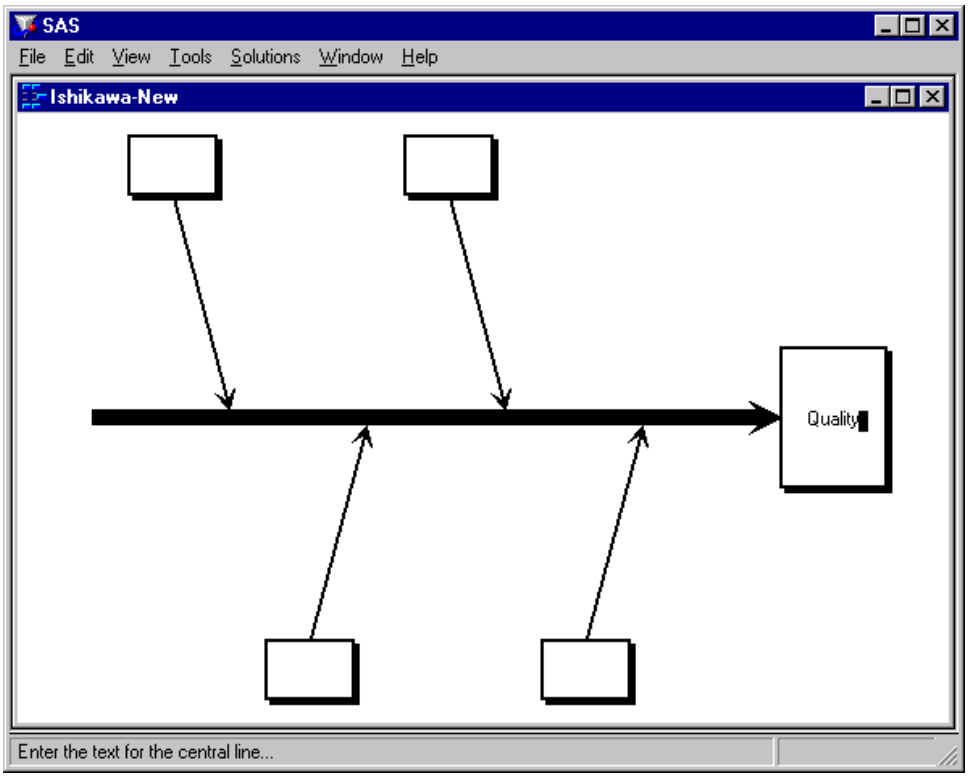
Figure 9.5 Starting a New Ishikawa Diagram



The ISHIKAWA environment guides you through the first steps of the diagramming process by prompting you to enter the text for the central line and then the upper left branch. During each step, a message indicating the action required is displayed in the message area for this window. Once you have completed these preliminary steps, you can proceed in any order you want.

3. Initially, the text cursor is positioned inside the box for the trunk. A message is displayed directing you to enter the first line of text for the trunk. Type the word *Quality*.[2]

Figure 9.6 Labeling the Trunk

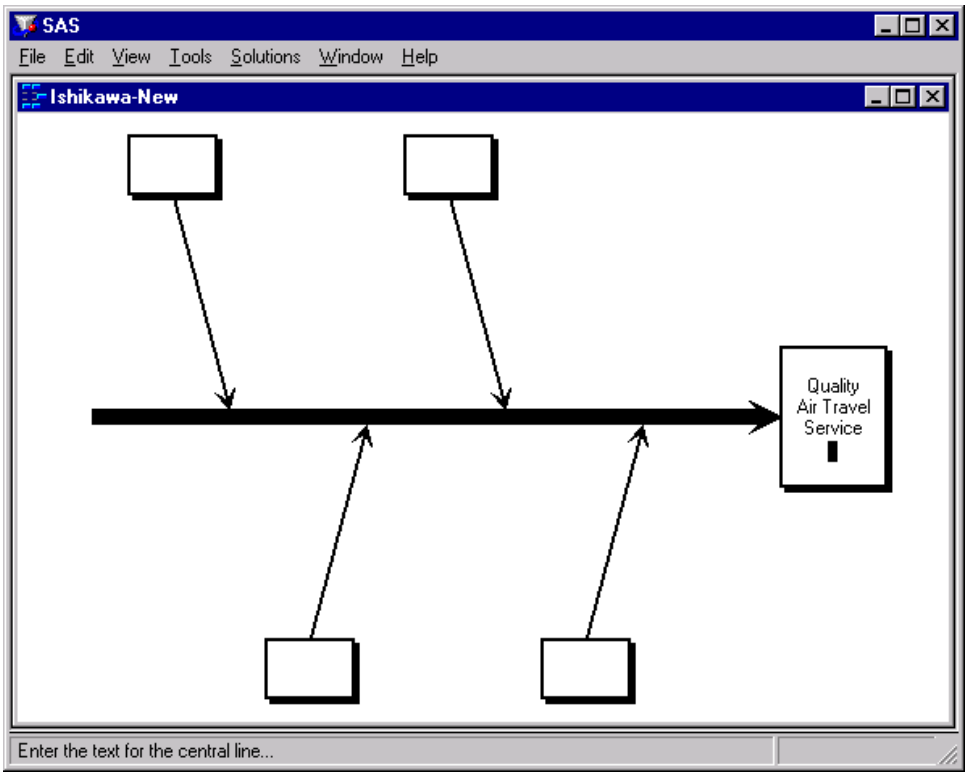


Note that the text is placed relative to the text cursor. You can correct mistakes by using any of the keyboard editing keys or cursor navigation keys (for instance, BACK SPACE and ←).

[2] You can skip this step, in future diagrams, by pressing RETURN before entering any text.

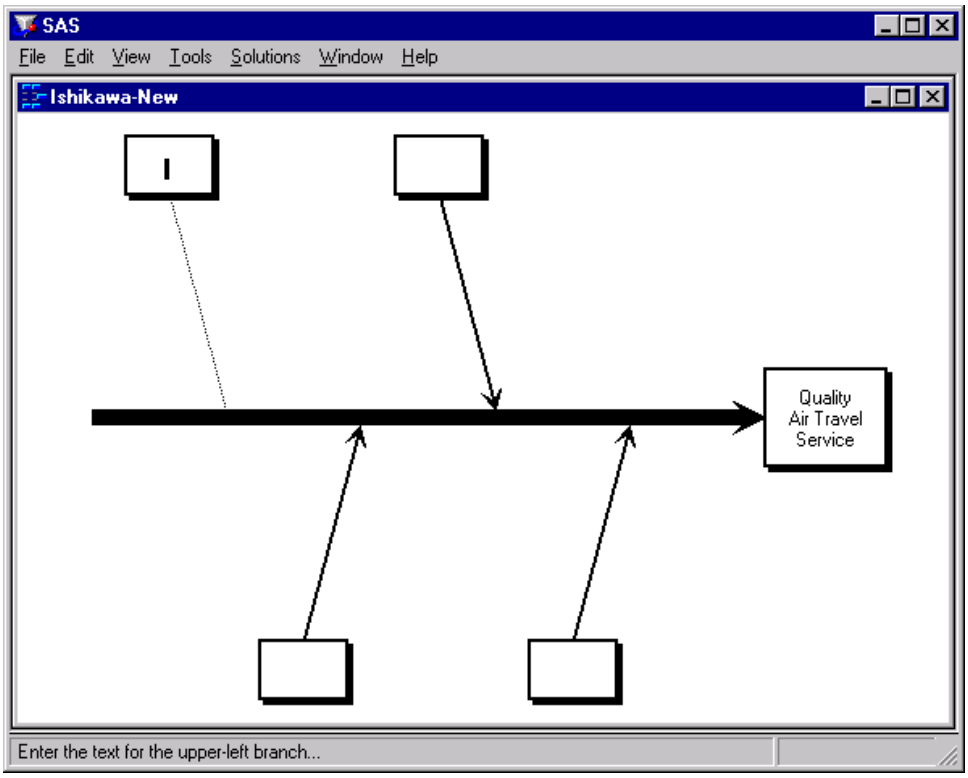
4. Advance to the next line by pressing RETURN. Now complete the label by entering *Air Travel* and *Service* on separate lines.

Figure 9.7 Labeling the Trunk *(continued)*



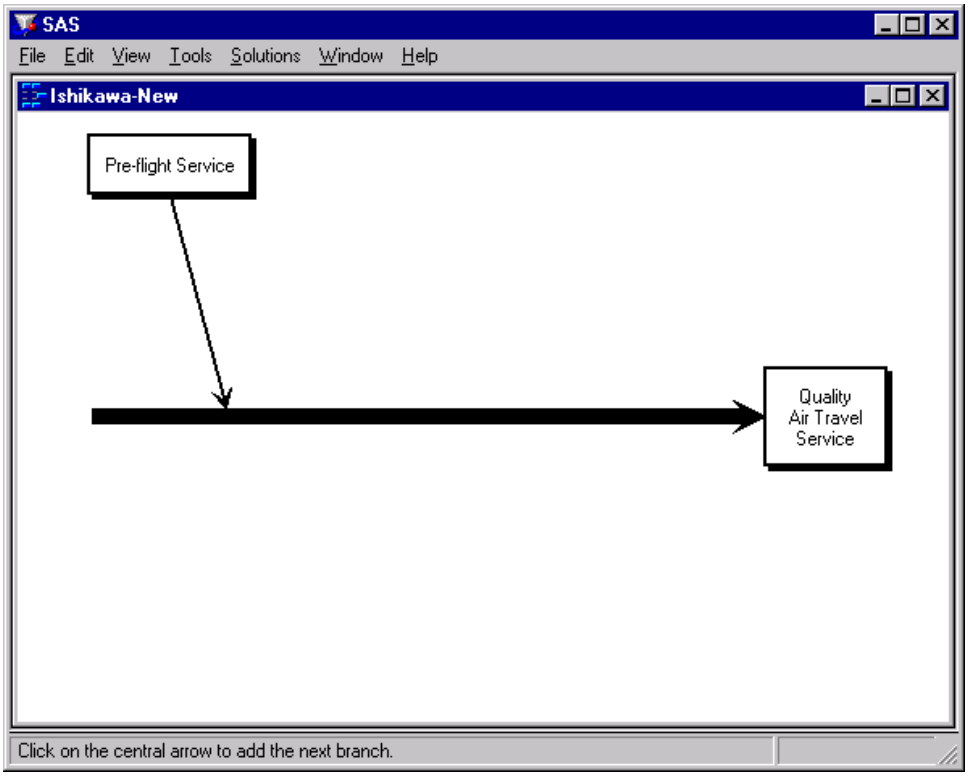
5. To terminate text entry, press RETURN a second time. The ISHIKAWA environment automatically moves the cursor to the upper left branch. If you made a mistake labeling the trunk, continue with the example. You cannot return to the trunk until you have finished the branch.

Figure 9.8 Labeling the First Branch



6. Enter the label *Pre-Flight Service*. Press RETURN twice to terminate text entry for this branch.[3] Your window should now look like this:

Figure 9.9 Completed Branch Label

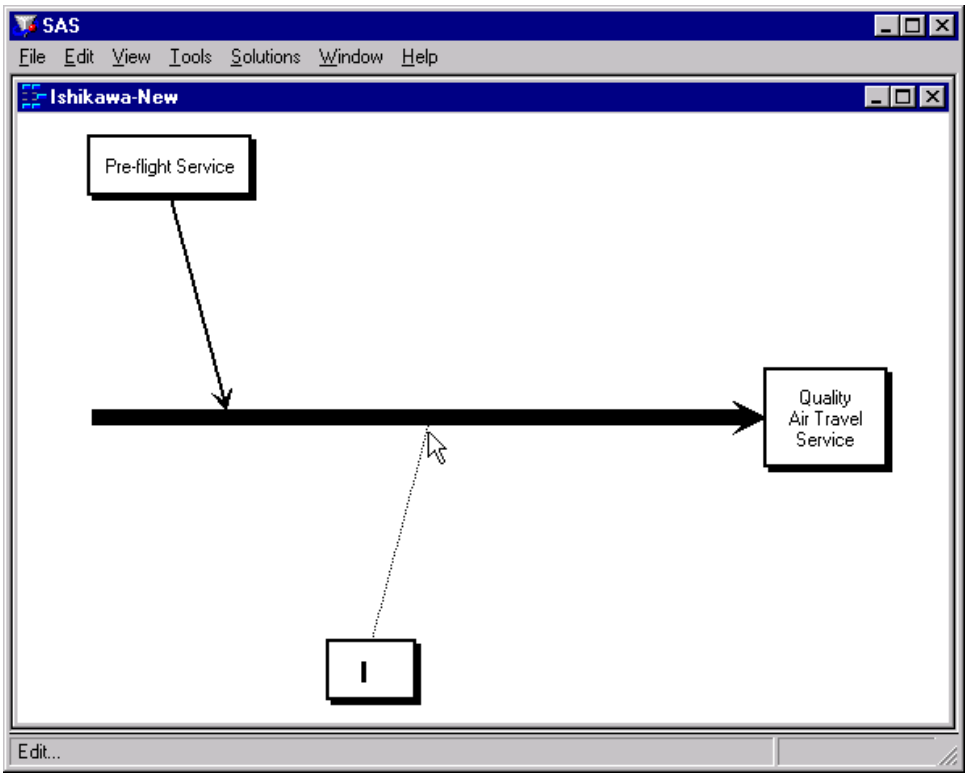


Note that when you finish entering text for the upper left branch, the other branches are deleted. These were temporarily displayed as visual cues, and now it is up to you to decide where to add the remaining branches.

[3] You can skip this step, in future diagrams, by pressing RETURN before entering any text.

7. To add the branch labeled *In-Flight Service* to the lower half of the diagram, position the cursor slightly below the point where you want the branch to attach to the trunk and click the mouse button. The branch appears with the text cursor centered inside the box. Enter the first line of text.

Figure 9.10 Adding a New Branch



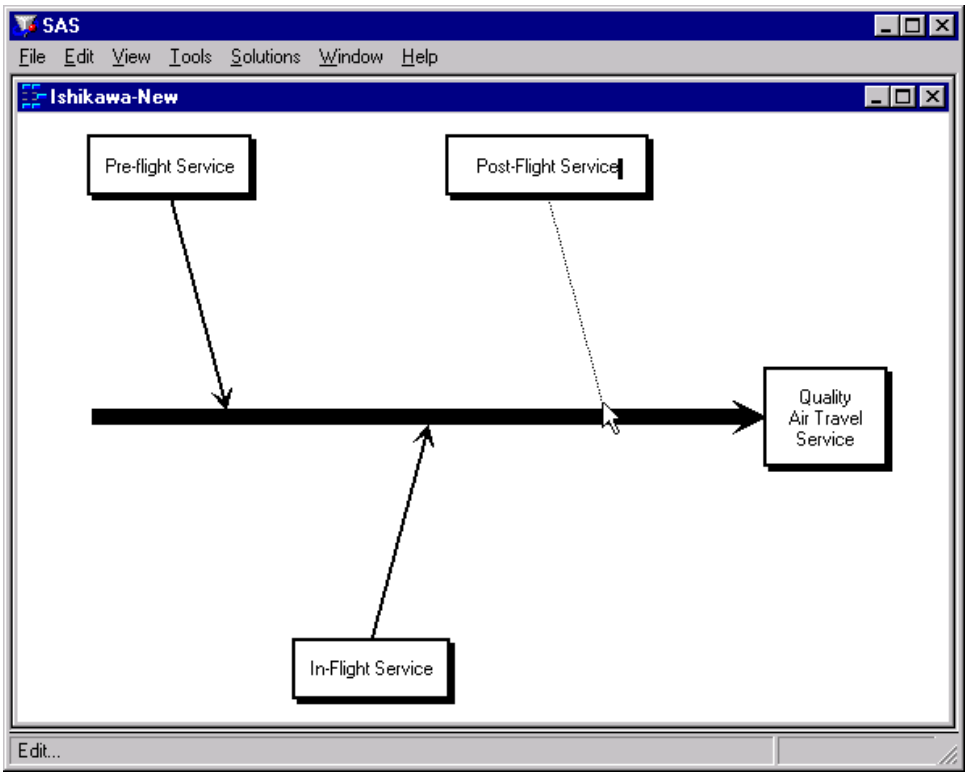
If your branch is not positioned where you want it, move the cursor to the appropriate position along the trunk and click. Each time you click, the branch is moved to the new location.

If, on the other hand, the branch is not drawn at all, the cursor was probably too far away from the trunk to be recognized. Move the cursor closer to the trunk and try again.

Enter the label *In-Flight Service* and press RETURN twice to terminate text entry.

8. Next add the branch *Post-Flight Service* to the upper half of the diagram. Position the cursor so that it is just above the point where you want the branch to attach to the trunk and click.

Figure 9.11 Adding the Last Branch



9. Press RETURN twice to terminate text entry.

Congratulations. You have just completed your first Ishikawa diagram using the ISHIKAWA procedure. In the process you learned to add branches to a diagram using context-sensitive mouse clicks. Future examples will illustrate other context sensitive areas, tools, and popup menus.

The examples that follow will, for the most part, expand on this diagram. To save the diagram, select **File ► Save as ► Data Set** from the command bar. Use SASUSER for the library name and AIRLINE for the data set name. Then select **Save**.

To leave the ISHIKAWA environment and return to the SAS Display Manager, select **File ► Close** from the command bar.

Syntax: ISHIKAWA Procedure

There are only three options that can be specified in the PROC ISHIKAWA statement, since the ISHIKAWA procedure is primarily a user-driven procedure.

DATA=*SAS-data-set*
identifies the name of a SAS data set that specifies an existing Ishikawa diagram. By default, the procedure will prompt you to edit an existing Ishikawa diagram or start a new one. When you specify the DATA= option, the procedure bypasses this initial menu. For example, the following statements simplify editing an existing Ishikawa diagram saved in a SAS data set:

```
proc ishikawa data=work.airline;
run;
```

NEW
starts a new Ishikawa diagram. By default, the procedure will prompt you to edit an existing Ishikawa diagram or start a new one. When you specify the NEW option, the procedure bypasses this initial menu and starts with a new diagram. Do not specify any other options when using the NEW option. For example, the following statements simplify starting a new Ishikawa diagram:

```
proc ishikawa new;
run;
```

NOFS
allows you to create hard copies of Ishikawa diagrams saved as SAS data sets without invoking the interactive features of the procedure. You must specify the DATA= option when you use the NOFS option. For example, the following statements create a hard copy of the Ishikawa diagram saved in the SAS data set *work.airline*:

```
goptions dev=psl noprompt;
proc ishikawa data=work.airline nofs;
run;
```

Details: ISHIKAWA Procedure

This chapter presents detailed information about and examples of all the operations available in the ISHIKAWA environment. Some of the examples build upon the diagram created in the tutorial.

Summary of Operations

To invoke the following context-sensitive operations, apply the specified action (mouse event) to the appropriate hotspot, using the left mouse button:

Table 9.1 Primary Operations

Operation	Mouse Event	Hotspot	Section
Add	Click	Near the intended attachment point	"Adding Arrows"
Edit	Click	Arrow tail	"Labeling Arrows"
Move	Click *(to pick)*	Arrow head	
	Click *(to drop)*	Near the intended attachment point	"Moving Arrows"
Delete	Double click	Arrow head	"Deleting Arrows"
Resize	Drag	Arrow tail	"Resizing Arrows"
Notepad	Double click	Arrow tail	"Notepads"

To invoke the following operations, make the specified selection from the appropriate context-sensitive popup menu using the right mouse button:

Table 9.2 Secondary Operations

Operation	Menu	Selection	Section
Swap	Head or tail	**Swap**	"Swapping Arrows"
Balance	Head or tail	**Balance**	"Balancing Arrows"
Hide Detail	Background	**< Detail**	"Managing Complexity"
Show Detail	Background	**> Detail**	"Managing Complexity"
Zoom	Head or tail	**Zoom**	"Zooming Arrows"
Isolate	Head or tail	**Isolate**	"Isolating Arrows"
Print	Pull-down	**File ► Save as ► Graph**	"Creating Graphics Output Using SAS/GRAPH Software"
Save	Pull-down	**File ► Save as ► Data Set**	"Saving an Ishikawa Diagram for Future Editing"
Save	Pull-down	**File ► Save as ► Image**	
Subset	Head or tail	**Subset**	"Modifying Arrow Colors and Line Styles"
Copy	Head or tail	**Copy**	"Merging Diagrams"
Refresh	Background	**Refresh**	
Unsubset	Background	**Unsubset**	"Modifying Arrow Colors and Line Styles"
Unbalance	Background	**Unbalance**	"Balancing Arrows"
Undelete	Background	**Undelete**	"Deleting Arrows"

When applied to the appropriate hotspots, the following actions (mouse events) invoke these context-sensitive operations:

Table 9.3 Context-Sensitive Tools

Hotspot	Mouse Event	Operation	Section
Arrow Head	Click	Begin move	"Moving Arrows"
	Double click	Delete	"Deleting Arrows"
	Drag	Resize	"Resizing Arrows"
	Popup menu	Subset	"Modifying Arrow Colors and Line Styles"
		Balance	"Balancing Arrows"
		Swap	"Swapping Arrows"
		Copy	"Merging Diagrams"
		Zoom	"Zooming Arrows"
		Isolate	"Isolating Arrows"
Arrow Tail	Click	Edit	"Labeling Arrows"
	Double click	Notepad	"Notepads"
	Drag	Resize	"Resizing Arrows"
	Popup menu	Subset	"Modifying Arrow Colors and Line Styles"
		Balance	"Balancing Arrows"
		Swap	"Swapping Arrows"
		Copy	"Merging Diagrams"
Arrow	Click	Add new arrow or	"Adding Arrows"
		complete move operation	"Moving Arrows"
Window	Click	Drop (finish) pending action	
Background	Drag	Drop (finish) pending action	
	Popup menu	Undelete	"Deleting Arrows"
		Unsubset	"Modifying Arrow Colors and Line Styles"
		Unbalance	"Balancing Arrows"
		Show Detail	"Managing Complexity"
		Hide Detail	"Managing Complexity"
		Refresh	

The File menu on the command bar can be used to control the following operations:

Table 9.4 File Menu

File ►	Description	Section
New…	Start a new diagram	"Getting Started: ISHIKAWA Procedure"
Open…	Open an existing diagram	"Reading an Existing Ishikawa Diagram"
Close	Close the current window	
Merge	Merge in an existing diagram	"Merging Diagrams"
Save as ►		
Data Set	Save as a SAS data set	"Saving an Ishikawa Diagram for Future Editing"
Graph	Print using SAS/GRAPH software	"Creating Graphics Output Using SAS/GRAPH Software"
Image	Save as an IMAGE, catalog entry	"Save as an IMAGE"
Export as Bitmap ►		
File…	Copy to a bitmap file	"Creating Bitmap Graphics Output"
Customize…	Export options	"Export as Bitmap"

The Edit menu on the command bar can be used to control the following operations:

Table 9.5 Edit Menu

Edit ►	Description	Section
Copy	Copy the diagram to host clipboard	"Creating Bitmap Graphics Output"
Clear…	Clear the window	

The View menu on the command bar can be used to control the following operations:

Table 9.6 View Menu

View ►	Description	Section
Ishikawa Settings ►		
Palettes	Line and color palettes	"Modifying Arrow Colors and Line Styles"
Background Color	Change window background color	
Save Attributes	Save window attributes: size, fonts and background color	
Balance Method ►	Select a balancing style	"Balancing Arrows"
Resize Method ►	Select a resizing method	"Resizing Arrows"
Primary Fonts...	Font dialog for first 3 levels	"Modifying Fonts"
Secondary Fonts...	Font dialog for levels 4-10	"Modifying Fonts"
Colors...	Color dialog	"Modifying Box Colors"
Arrows...	Arrow style dialog	"Modifying Arrow Heads"
Other...	Style dialog	"Modifying Environmental Attributes"
Refresh	Refresh the window	

The Help menu on the command bar can be used to control the following operations:

Table 9.7 Help Menu

Help ►	Description
SAS System Help	SAS help system
Using This Window	Ishikawa specific help

Operations

This section provides details concerning the operations available in the ISHIKAWA environment. The order in which the topics appear is the order in which the operations are typically encountered. Some of the examples in this section build upon the diagram created in the tutorial.

Adding Arrows

You add an arrow by pointing with the mouse to the intended attachment point along an existing arrow and clicking the mouse. You control the direction of the new arrow by offsetting the mouse cursor a small distance away from the parent arrow on the side where the new arrow is to appear.

For example, to add upper branches, you offset the cursor slightly above the trunk. To add lower branches, you offset the cursor slightly below the trunk. Likewise, you offset the cursor to the right of the branch to add a right-hand stem and slightly left for a left-hand stem.

If a new arrow is not drawn as you intended (either positionally or directionally), you can easily move or delete it. To delete a new arrow before you have entered any text, click in the background. To move a new arrow before you have entered any text, move the cursor to a new attachment point and click.

Once an arrow is drawn, you are immediately prompted for its label (note the hint, *Edit...*, displayed on the message line and the appearance of the text cursor at the end of the arrow). See "Labeling Arrows" on page 724, for details on the text editing features of the ISHIKAWA environment.

A diagram can contain up to ten levels of detail, but the number of arrows is limited only by the resolution and size of your graphics display.

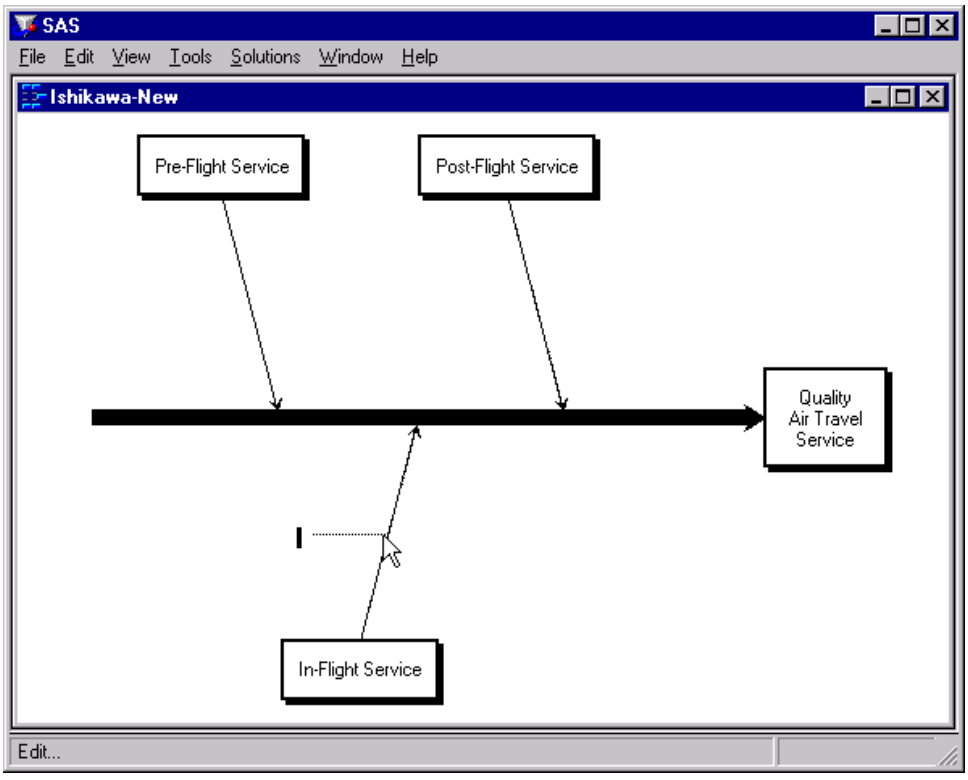
Example

Continuing with the tutorial example from "Getting Started: ISHIKAWA Procedure" on page 707, suppose that you have obtained detailed information for each of the three major service areas, which you want to display by adding stems to the branches of the diagram you previously created. If you closed the ISHIKAWA environment after saving the data set, SASUSER.AIRLINE, you can easily restore the diagram by submitting:

```
proc ishikawa data=sasuser.airline;
run;
```

To add a stem to the left side of the branch labeled *In-Flight Service*, position the cursor so that it is just to the left of the point where you want the stem to attach. Click the mouse. The new arrow (pending text) appears as follows:

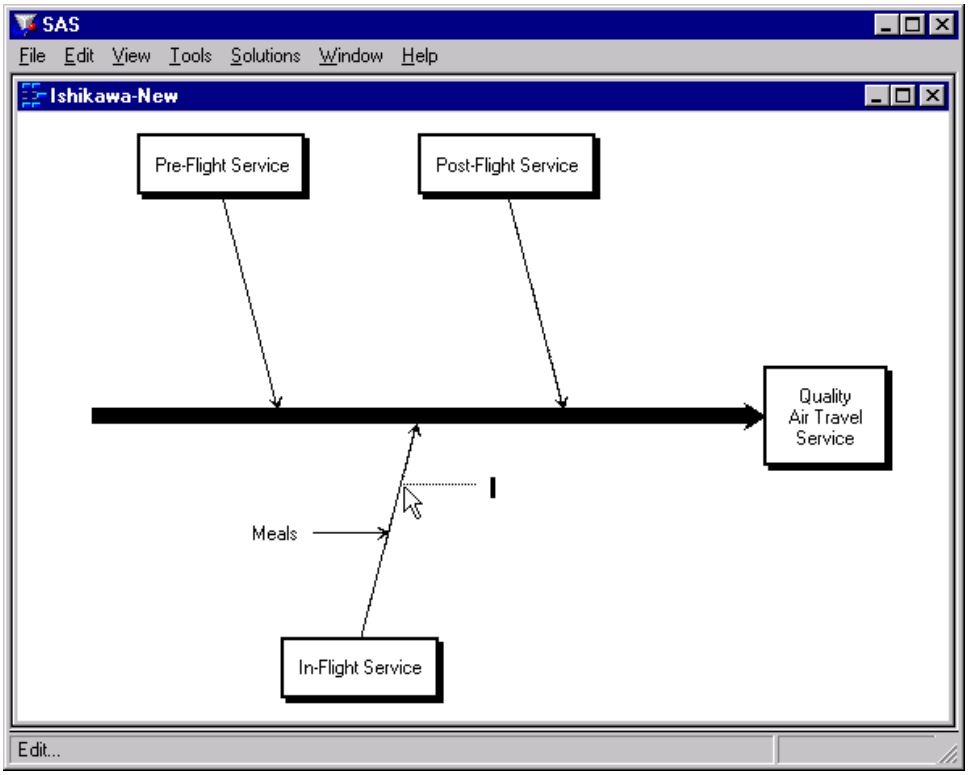
Figure 9.12 Adding the Left Stem



Type the label *Meals* and press RETURN twice.

To add a stem to the right side of the same branch, position the cursor so that it is just to the right of the attachment point. When you click the mouse, your window will appear as follows:

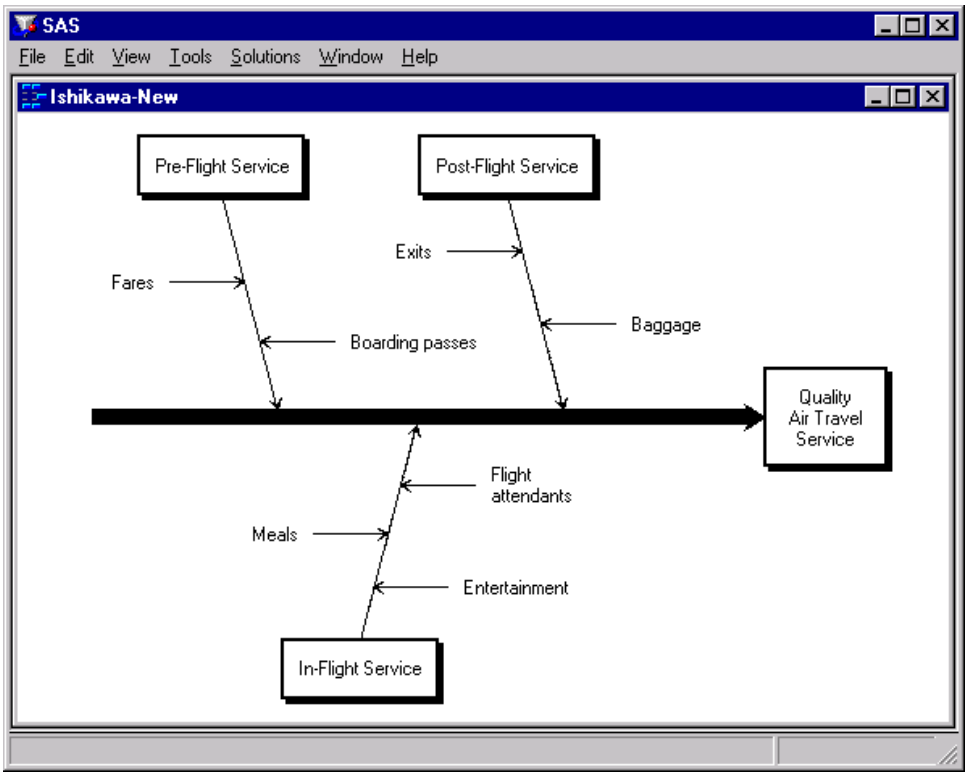
Figure 9.13 Adding the Right Stem



Type the label *Flight attendants* on two lines and press RETURN to terminate text entry.

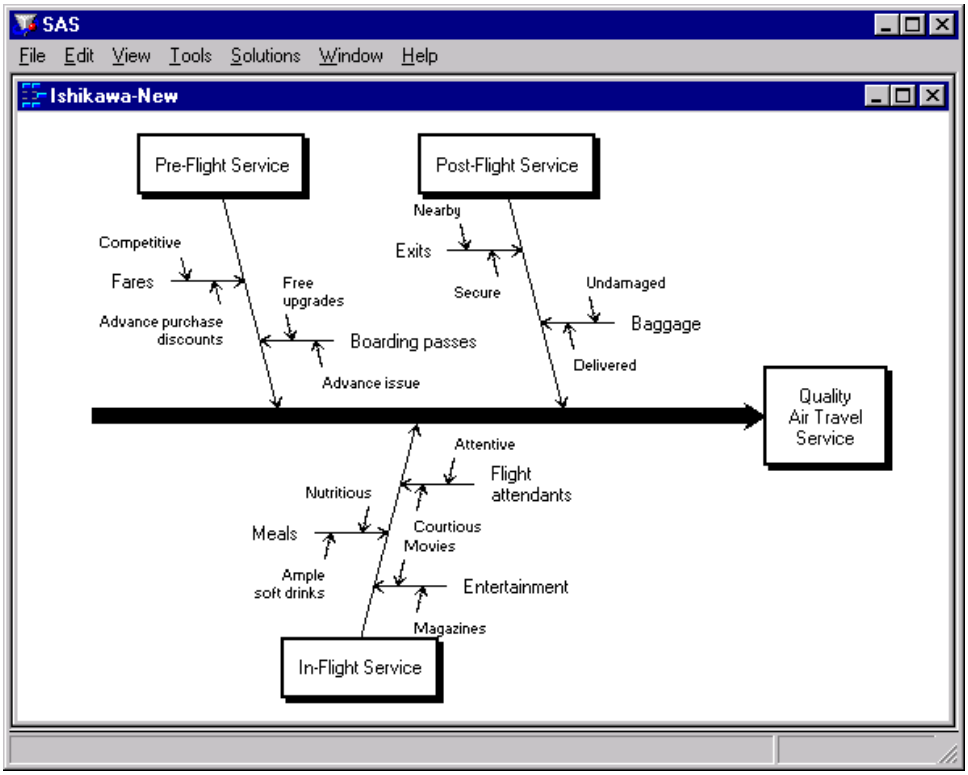
Complete the diagram by adding the remaining stems shown in the following window:

Figure 9.14 Stem-Level Diagram



Experiment further by adding several of the leaves shown in the following window. Don't be concerned if some of the labels collide with each other. Later, you will learn how to move and resize arrows.

Figure 9.15 Leaf-Level Diagram



Labeling Arrows

To edit the label of an existing arrow, click on one of the following areas:

- the label
- the arrow tail (if the arrow does not have a label)
- inside the box for trunk and branch labels

Use your keyboard to enter the text.

On hosts that support direct graphical text entry,[4] the following functions are supported:

- edit keys such as BACK SPACE, DELETE CHAR, DELETE LINE, and RETURN
- cursor navigation keys such as ↑, ↓, → and ←
- the INSERT key to toggle between insert and overstrike modes
- buffers to copy, cut, or paste text into and from external sources

Text entry is terminated whenever you press RETURN on an empty line or exceed the maximum line limit for a label. Text entry is also terminated whenever you click the mouse. This shifts focus away from the editing operation and to the new location.

Labels are restricted to 40 characters per line. The trunk label can have up to five lines, and labels for other levels are limited to two lines.

You can split a line of text into two lines by pressing the RETURN key anywhere inside the line. Likewise, flow a line with the previous line of text by pressing the BACK SPACE key at the beginning of the line.

You can copy the contents of the paste buffer into a label using the PASTE command. This can be helpful when the information for your diagram is available from another source (a flat file, for example). Use the paste buffer to copy the information from that source to your Ishikawa diagram.

Some hosts designate the right mouse button for pasting, some use control keys (like ctrl-p), while others use a designated function key. For more details about using paste buffers with the SAS System, consult the SAS companion for your host.

To paste text into a label, you must first select the label. For existing arrows, select the arrow, position the cursor where you want the text to appear, and then issue the PASTE command. For new arrows pending text entry, simply issue the PASTE command. Any text in the paste buffer that causes the label to exceed its limits is truncated.

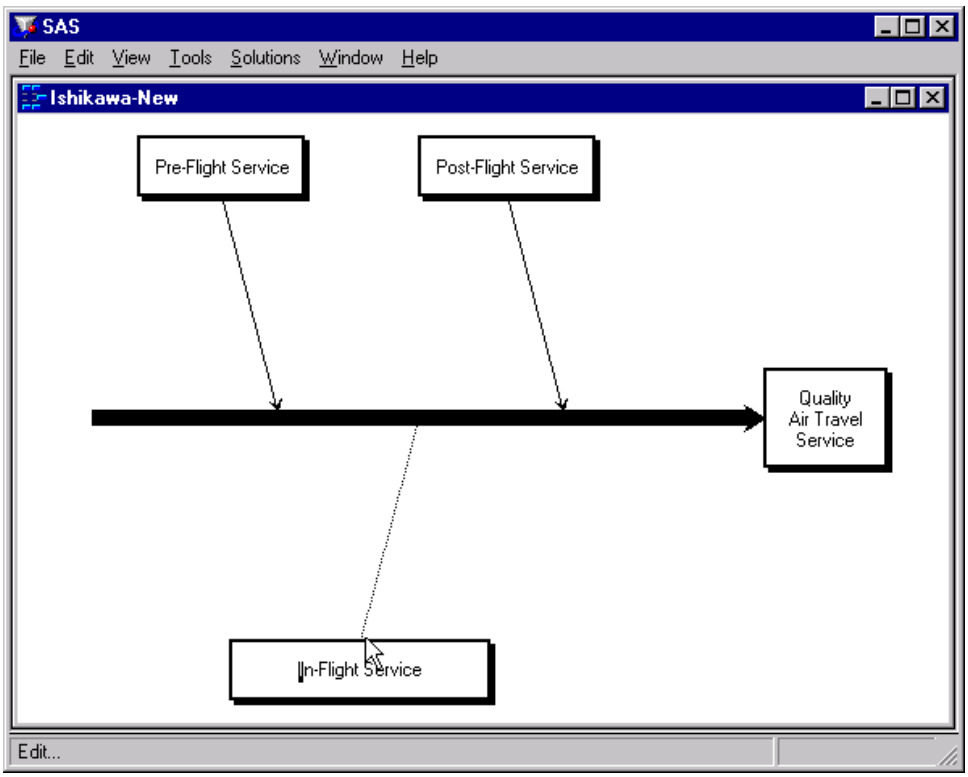
When your mouse has a paste key defined, instead of adding an arrow and pasting the text in two operations, use the *right* mouse button to add the arrow. This action adds a new arrow, automatically copies the label from the paste buffer, and terminates text entry, in a single operation.

Example

In the following diagram, the branch labeled *In-Flight Service* has been selected by clicking on the arrow tail. The arrow is highlighted with a narrow dotted line, and the text cursor is positioned over the first character in the label.

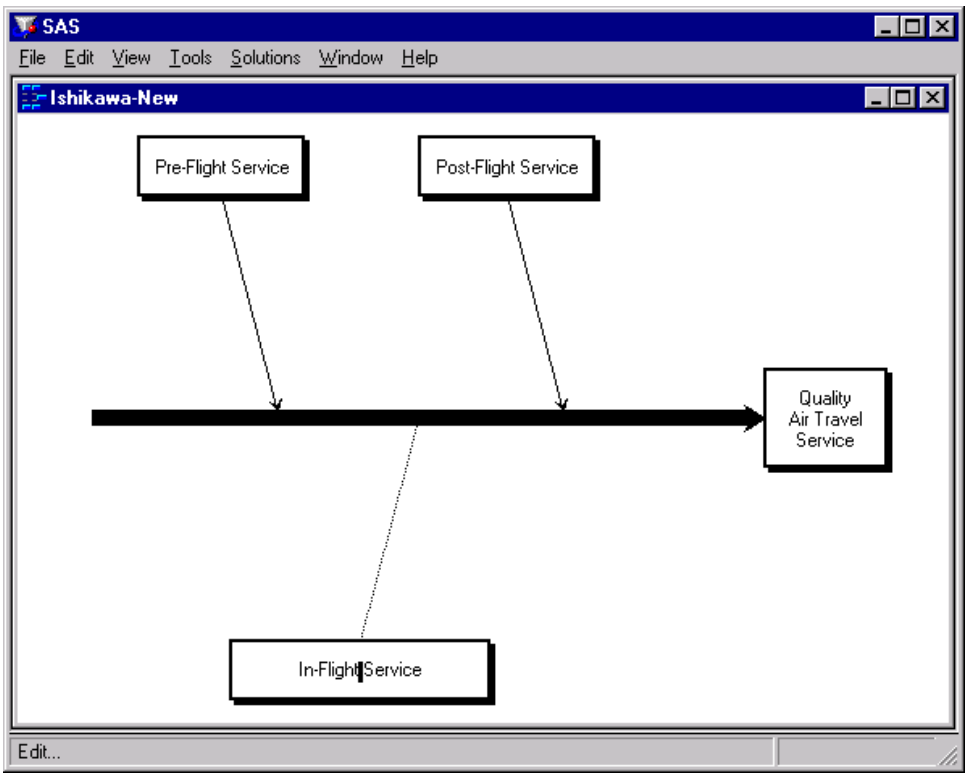
[4] Devices such as the IBM3179 do not support the direct graphical text entry mechanism described in these examples. Instead, a text entry window pops up whenever you select an arrow for editing. You must edit the text for the arrow from the dialog box and close the text entry window before the diagram is updated.

Figure 9.16 Selecting an Arrow for Editing



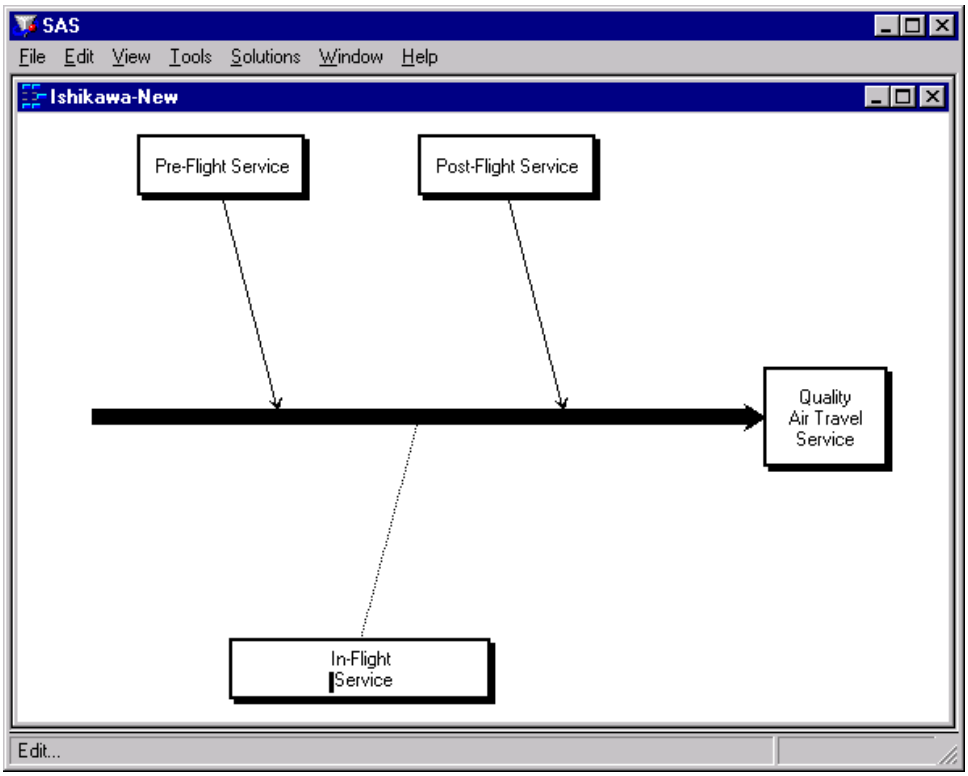
To change the label so that the word *Service* appears on a separate line, use the → or ← key to move the cursor to the space before the word *Service*, as shown in the following:

Figure 9.17 Using Cursor Keys



Now press RETURN to split the text into two lines.

Figure 9.18 Splitting Text



Remember to delete the space preceding *Service* before pressing RETURN to terminate text entry.

Moving Arrows

You move an arrow by picking up the arrow and dropping it at a new location:

- To *pick up* an arrow, position the cursor over the arrow head and click the mouse. The arrow you selected will be highlighted with a narrow dotted line. If the arrow is not highlighted, move the cursor closer to the arrow head and repeat the click.
- To *drop* an arrow, move the cursor slightly to one side of the new attachment point and click (just as though you are adding a new arrow).

When you move an arrow, all its descendants move with it.

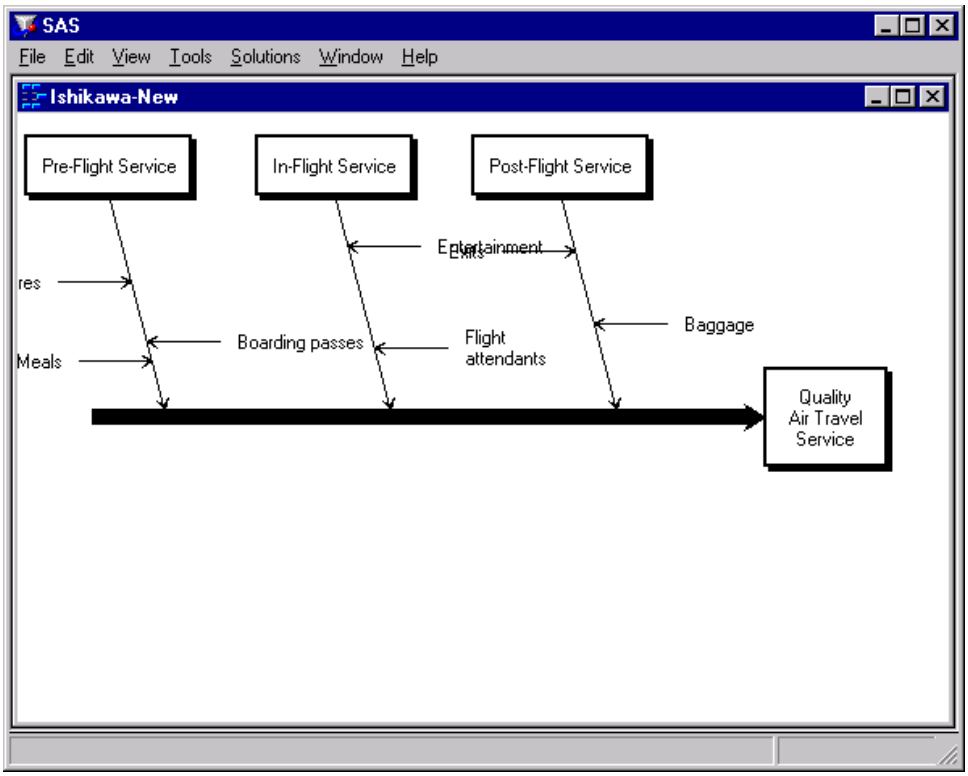
To cancel a move after picking up an arrow, click in the background area of the ISHIKAWA window.

Do not try to drop the arrow back into place by clicking on the arrow head a second time. A double click on (or near) the arrow head deletes the arrow. To move an arrow a short distance, move the cursor away from the arrow head before clicking to drop the arrow. On some systems the cursor will change shape when you have moved outside the context-sensitive area.

Example

As your diagrams develop, you will want to reposition arrows, either because of errors or for aesthetic reasons. The following is an example of an Ishikawa diagram that needs to be modified:

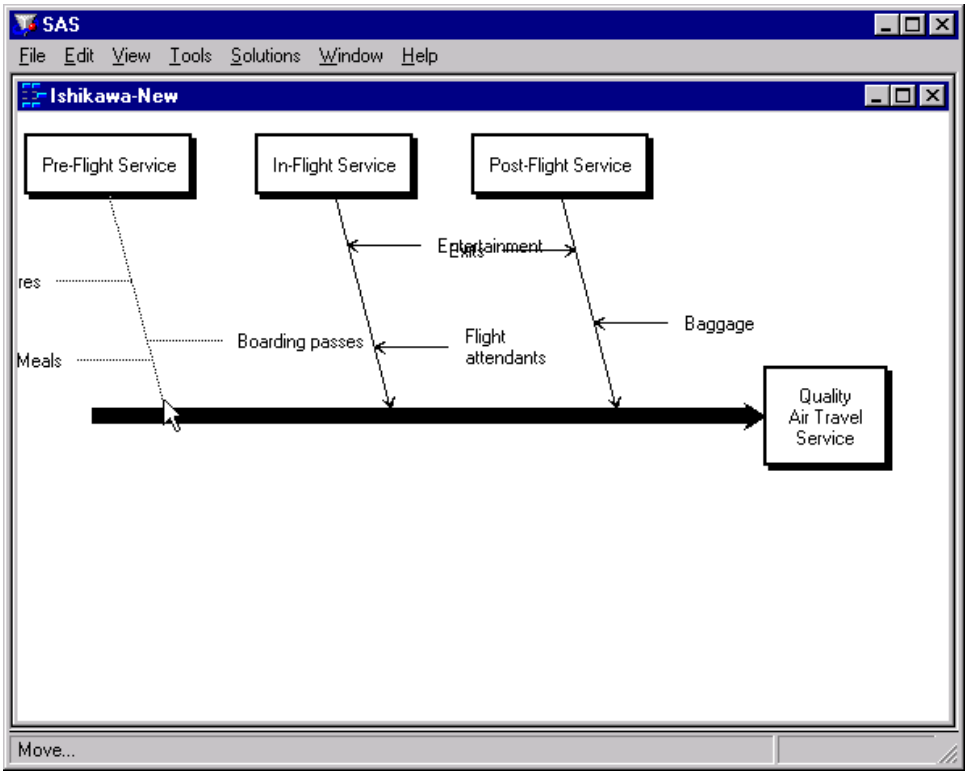
Figure 9.19 An Inelegantly Arranged Ishikawa Diagram



The diagram lacks balance, and some of the branches are too close, resulting in collisions and clipping.

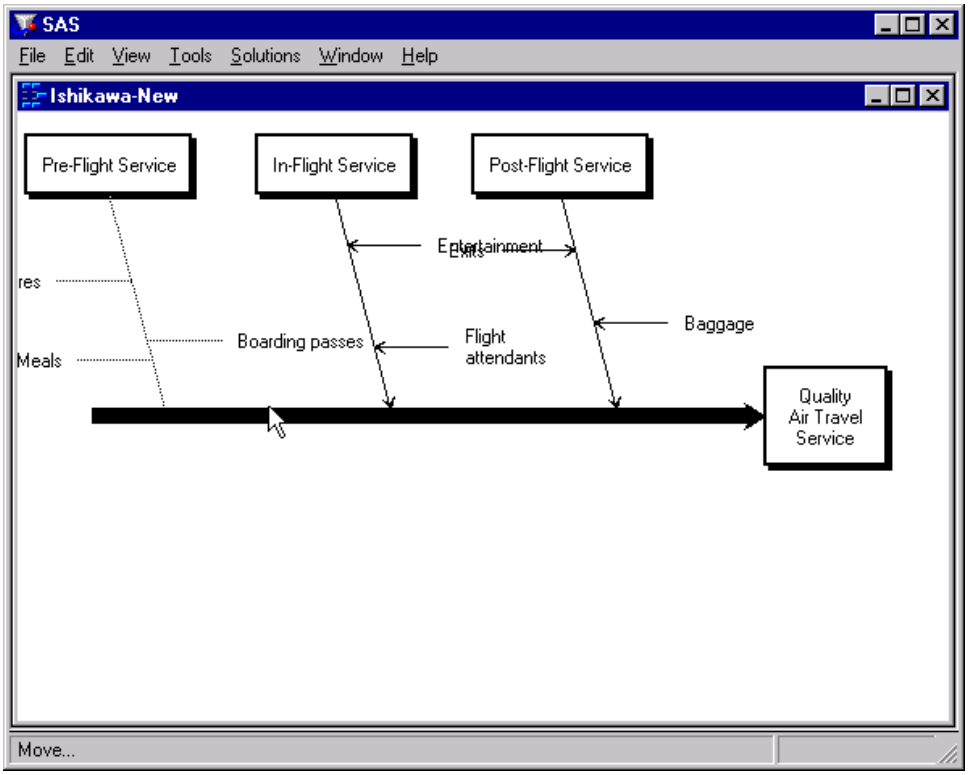
One way to improve the diagram is to move the branch for *Pre-Flight Service* toward the center of the trunk. First select the arrow head for this branch.

Figure 9.20 Selecting an Arrow to Move



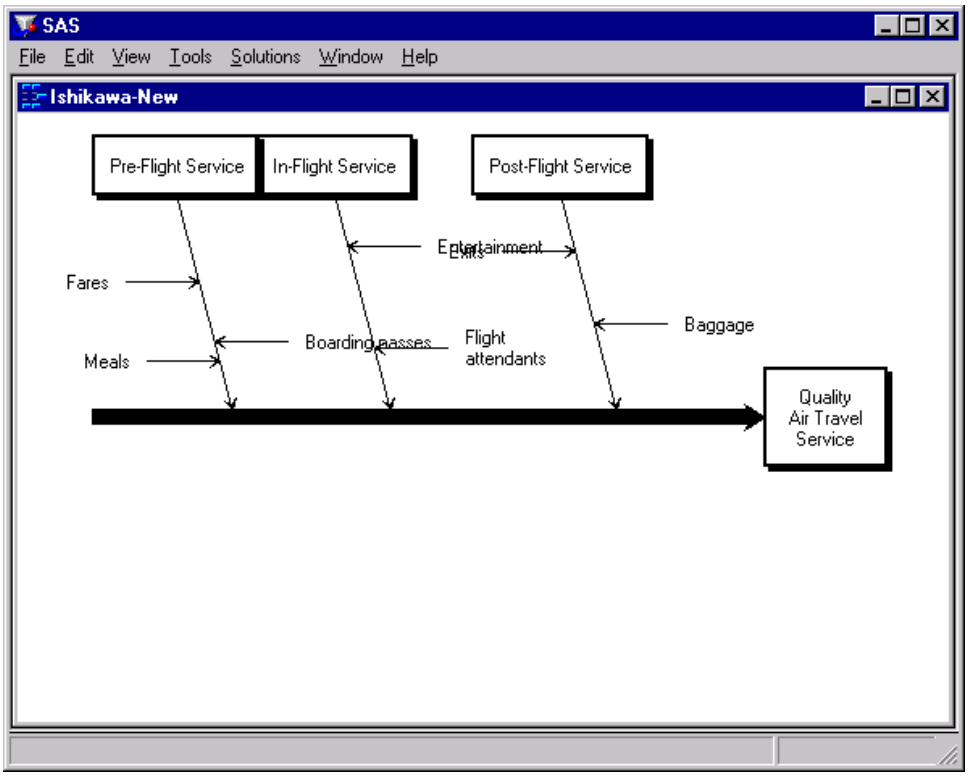
Then move the cursor to a point just slightly above the trunk near the desired new attachment point.

Figure 9.21 Locating the New Attachment Point



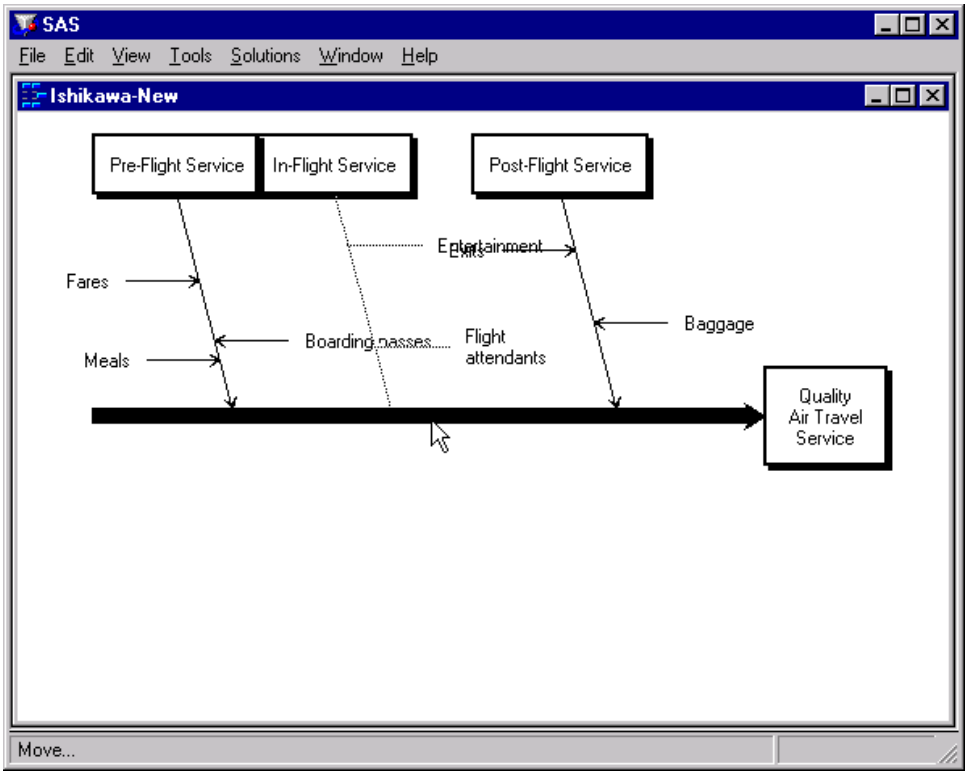
Drop the arrow in place by clicking the mouse.

Figure 9.22 Dropping an Arrow into Position



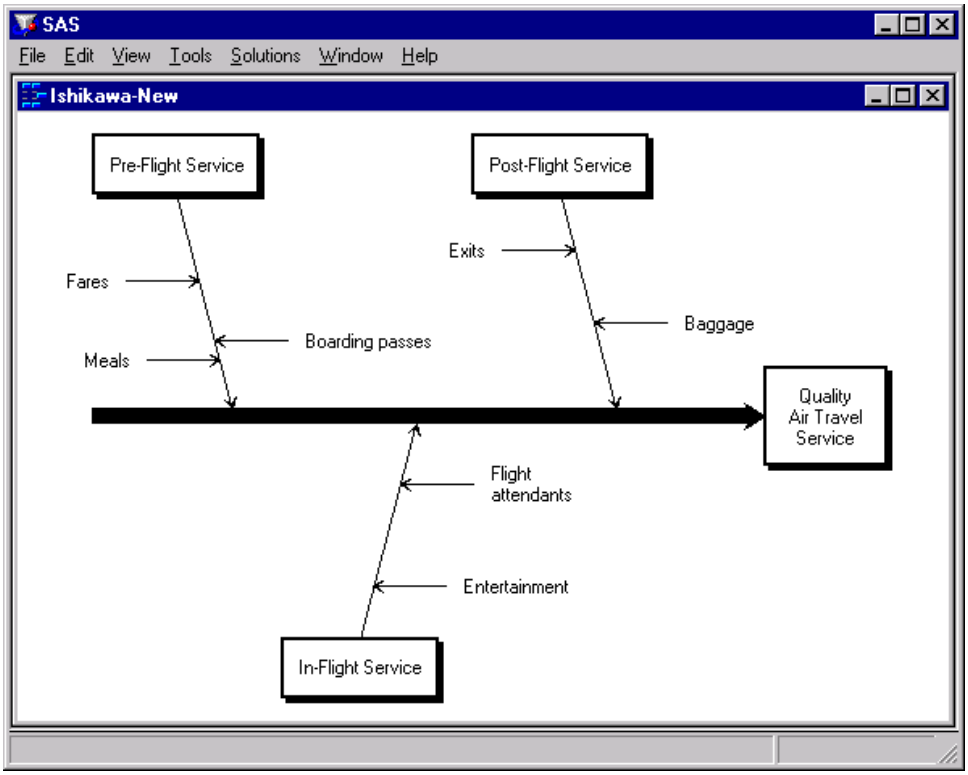
Next, you should reflect the middle branch to the lower half of the diagram to balance the diagram and eliminate the remaining collisions. Once you have selected the branch, position the cursor slightly below the trunk near the desired new attachment point.

Figure 9.23 Selecting an Arrow for Reflecting



Click the mouse to complete the reflection.

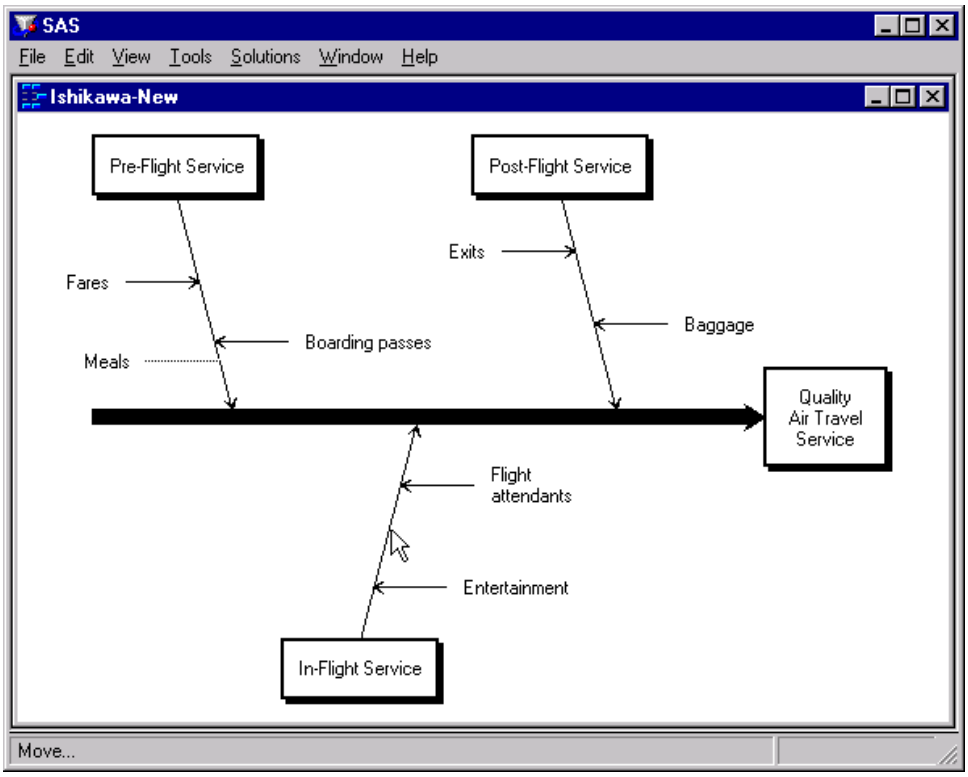
Figure 9.24 Reflecting an Arrow



Note that the stems are reflected with the branch and that their positions (relative to the trunk) are preserved.

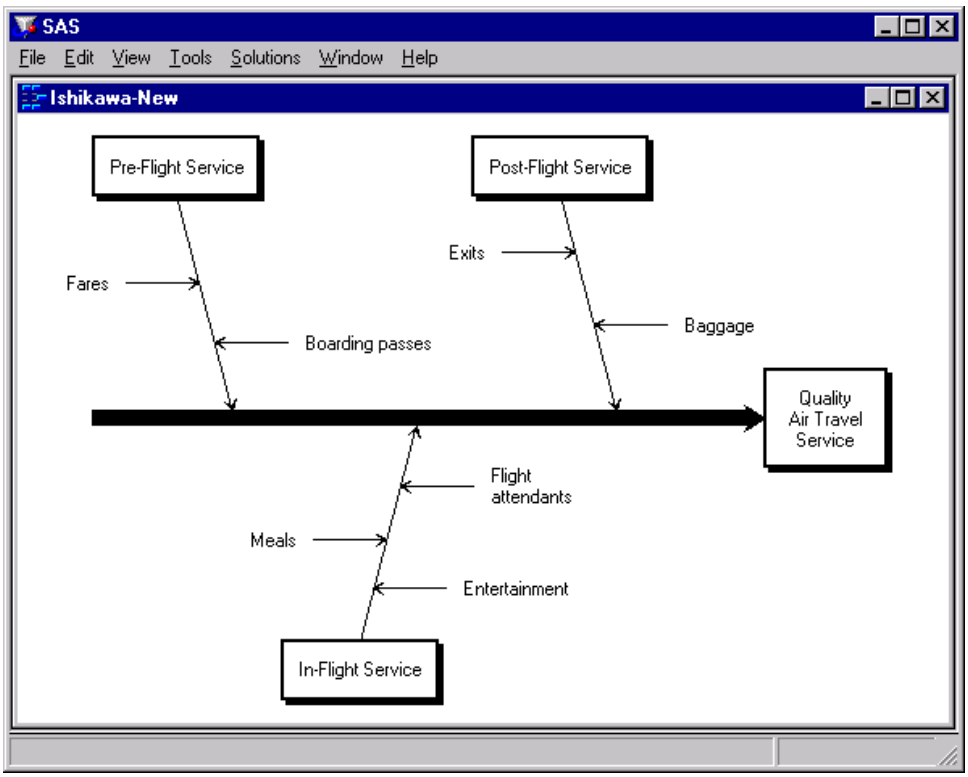
Finally, the stem labeled *Meals* is incorrectly attached to the branch labeled *Pre-Flight Service* and should be moved to the branch labeled *In-Flight Service*. Once you have selected the stem, move the cursor slightly left of the new attachment point.

Figure 9.25 Locating the New Attachment Point



To complete the move, click the mouse.

Figure 9.26 Moving a Stem



Apply the same principles when moving an arrow to a new level (for example, to elevate a stem to a branch) or a new diagram (when you have multiple ISHIKAWA windows open).

Deleting Arrows

You can delete an arrow (with all its descendants) by moving the cursor over the arrow head (attachment point) and double clicking. If you accidentally move the cursor while double clicking, it is possible that the arrow will be moved instead of being deleted. In that case, double click on the arrow head again.

You can undo a deletion by moving the cursor to a background area of the window and using the right mouse button to select **Undelete** from the background popup menu. Repeat the operation when you want to undo several deletions.

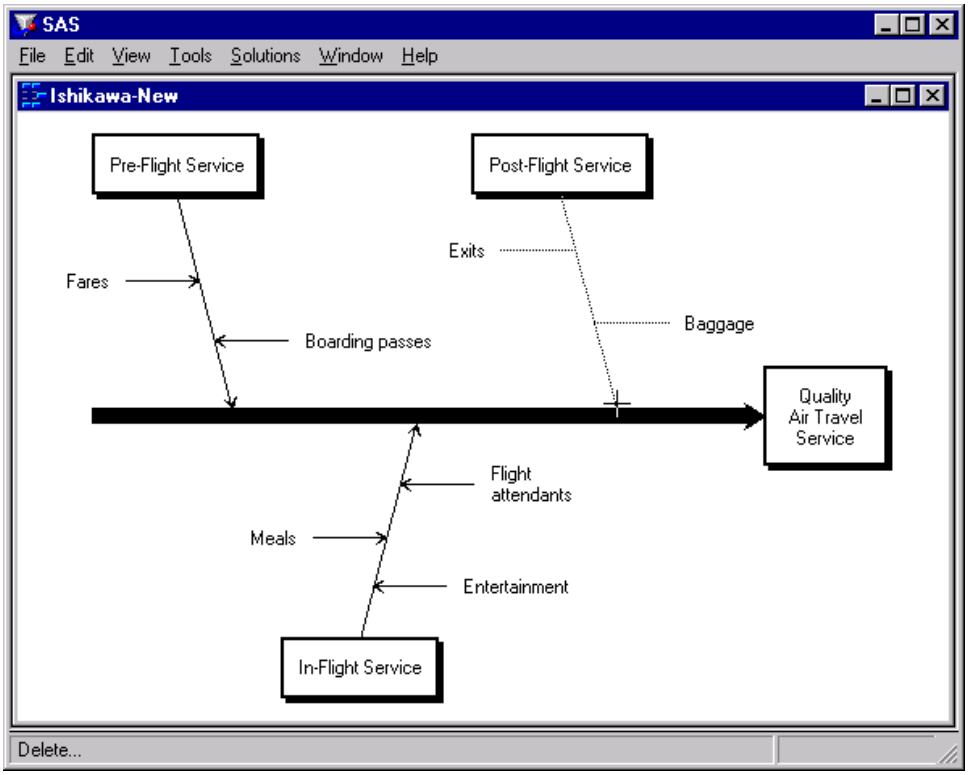
Once an arrow has been selected for deletion, you can cancel the pending operation by moving the cursor to a background area of the diagram and clicking the mouse.

The ISHIKAWA environment does not allow you to delete the trunk. To clear the window, select **Edit ► Clear...** from the command bar. Then start a new diagram by selecting **File ► New...** or **File ► Open...**.

Example

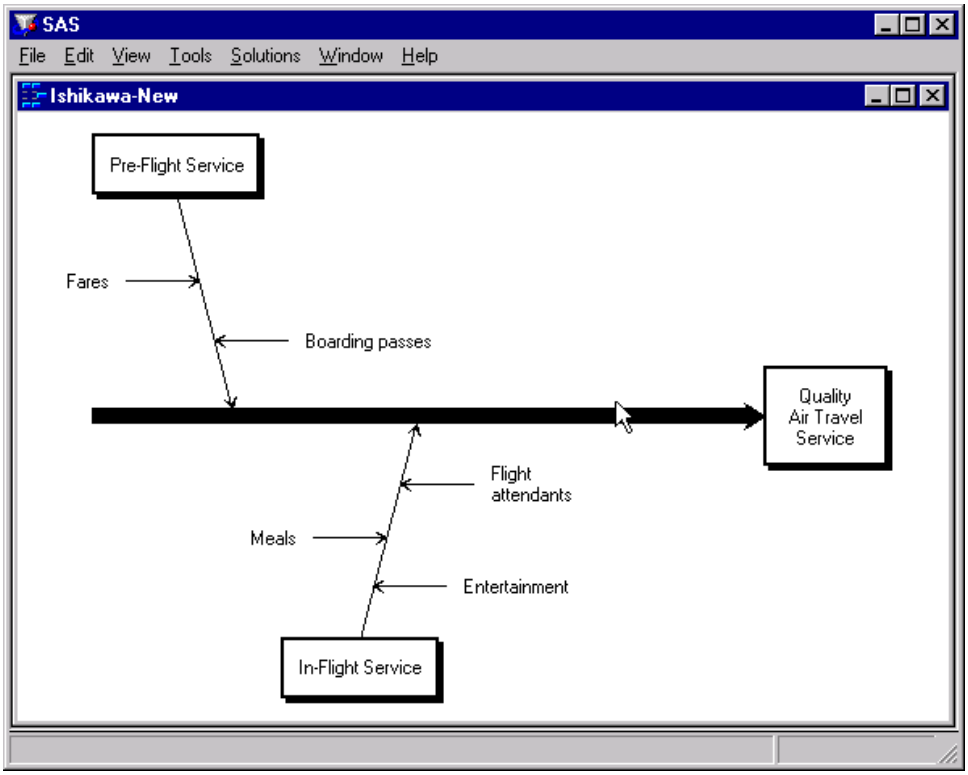
In the following diagram, the branch labeled *Post-Flight Service* has been selected for deletion (note that the branch is highlighted):

Figure 9.27 Selecting a Branch for Deletion



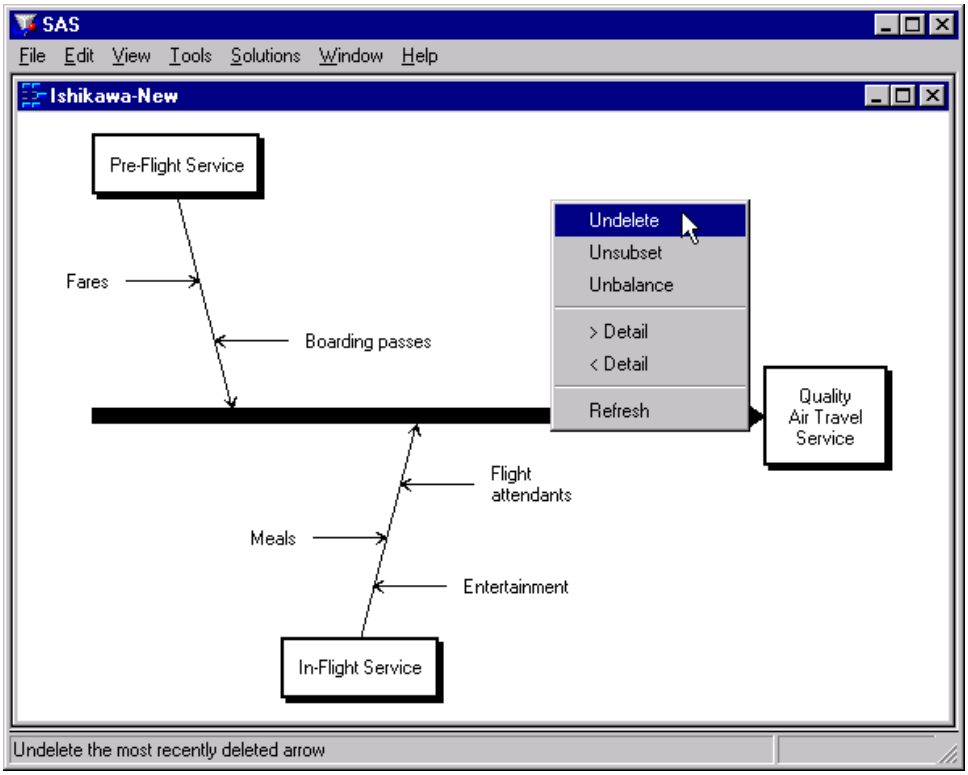
Without moving the cursor, click on the arrow head a second time to delete the branch.

Figure 9.28 Deleting a Branch



To undelete the previous deletion, move the cursor to a background area of the window and use the right mouse button to select **Undelete** from the background popup menu.

Figure 9.29 Undeleting a Branch



Resizing Arrows

You can resize an arrow by holding the mouse button down over the tail end of the arrow and dragging the mouse.[5] As you move the mouse, the arrow is represented by a rubberband line, and a plus sign (+) is drawn to indicate the original position of the arrow tail. The new length is determined by the position of the cursor when you release the mouse.

To cancel a resize operation once you have depressed the mouse button, release the button outside the ISHIKAWA window.

All non-horizontal arrows are constrained to have the same angle. You control the angle by resizing a branch. That is to say, when you resize a leaf, its angle does not change.

Use **View ► Ishikawa Settings ► Resize Method ►**to control the scope of the resizing operation.

- **Local** resizes only the arrow being dragged.
- **Global** resizes all the arrows at that level to lengths that are proportional to the arrow being dragged. This is the default.
- **Uniform** resizes all arrows at that level to the length of the arrow being dragged.

When you resize an arrow, you also update the default size for all new arrows at that level.

By default, global and uniform resizing applies to all the arrows at the level of the arrow being resized. To restrict resizing to a specific subset of arrows, you can subset them as follows:

- Move the cursor over the arrow head of an arrow to subset that arrow and all its descendants.
- Move the cursor over the arrow tail of an arrow to subset only that arrow (and not its descendants).
- Use the right mouse button to activate the popup menu.
- Select **Subset**.

On some hosts, shift-clicking on the arrow head or tail also subsets an arrow.

Subsetted arrows are indicated by underlined labels. Subsetting is a toggle operation, so to *unsubset* an arrow, repeat the preceding steps.

To unsubset all the arrows in the diagram, do the following:

- Move the cursor to a background area of the window.
- Use the right mouse button to activate the background popup menu.
- Select **Unsubset**.

Be sure to remove all subsets after you have finished modifying the diagram, since remaining subsets can alter the focus of other operations.

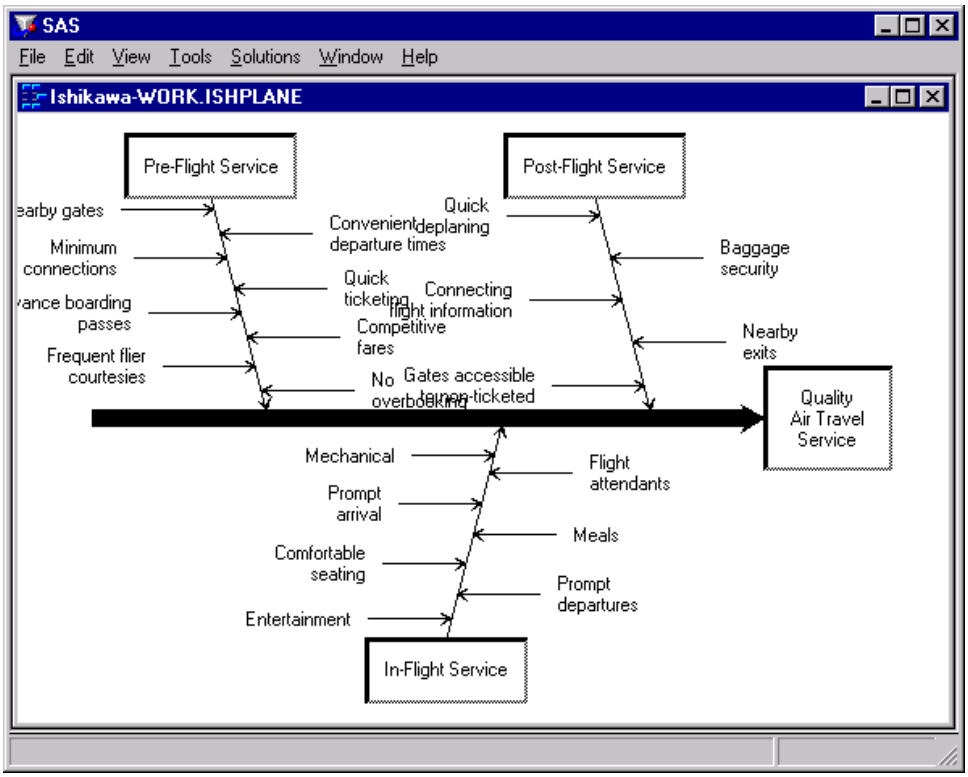
See “Modifying Arrow Colors and Line Styles” on page 764, for more examples of how subsets are used.

[5] Some devices (such as the IBM3179) require you to define a drag key. For more details about dragging on your system, consult the SAS companion for your host.

Example

Arrows that are too long can cause clipping and collisions, as illustrated in the following diagram:

Figure 9.30 Before Resizing the Diagram

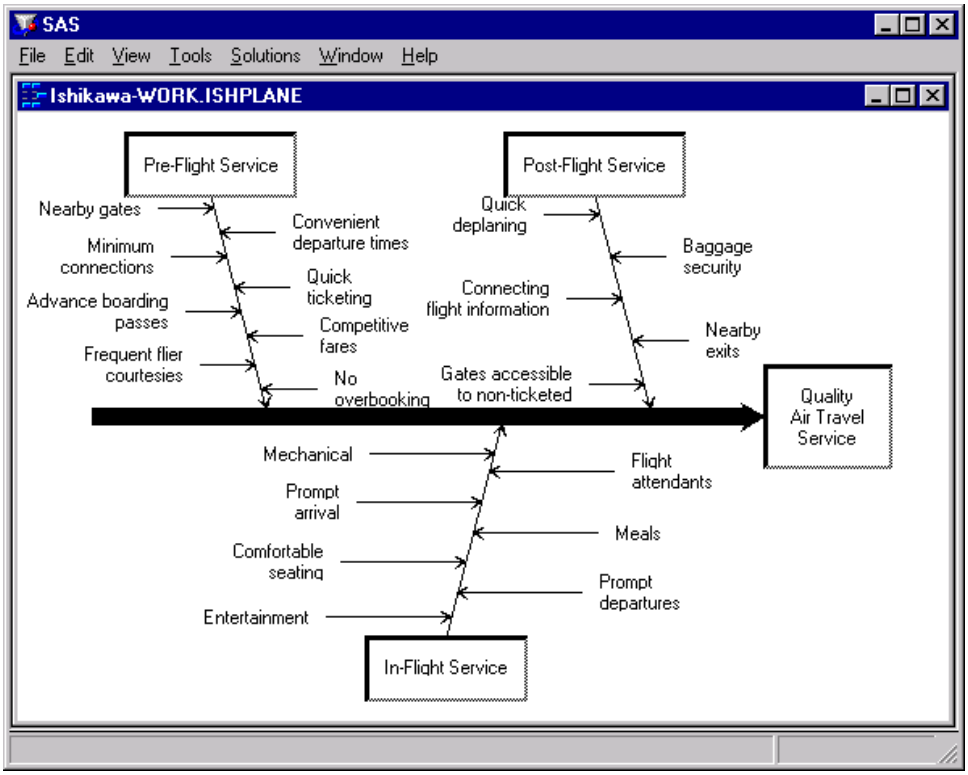


To resize the stems in the upper half of the diagram, proceed as follows:

- Subset the branch for *Pre-Flight Service* by moving the cursor over its arrow head and selecting **Subset**.
- Do the same to *Post-Flight Service*.
- Shorten one of the subsetted stems by dragging its tail to the desired length.
- Remove all subsets by selecting **Unsubset**.

The results are as follows:

Figure 9.31 After Resizing the Diagram



Swapping Arrows

Use the swap operation to interchange two arrows in a single operation instead of using two move operations. Swapping has all the flexibility of the move operation; you can swap arrows that have different parents, different levels, or arrows from different diagrams.

Like moving, the results depend upon whether you select the arrow from the arrow head or the arrow tail. When you select the arrow head, the arrow and all its descendants are moved. When you select the arrow tail, only the labels of the selected arrows are interchanged.

Swapping is a two step operation.

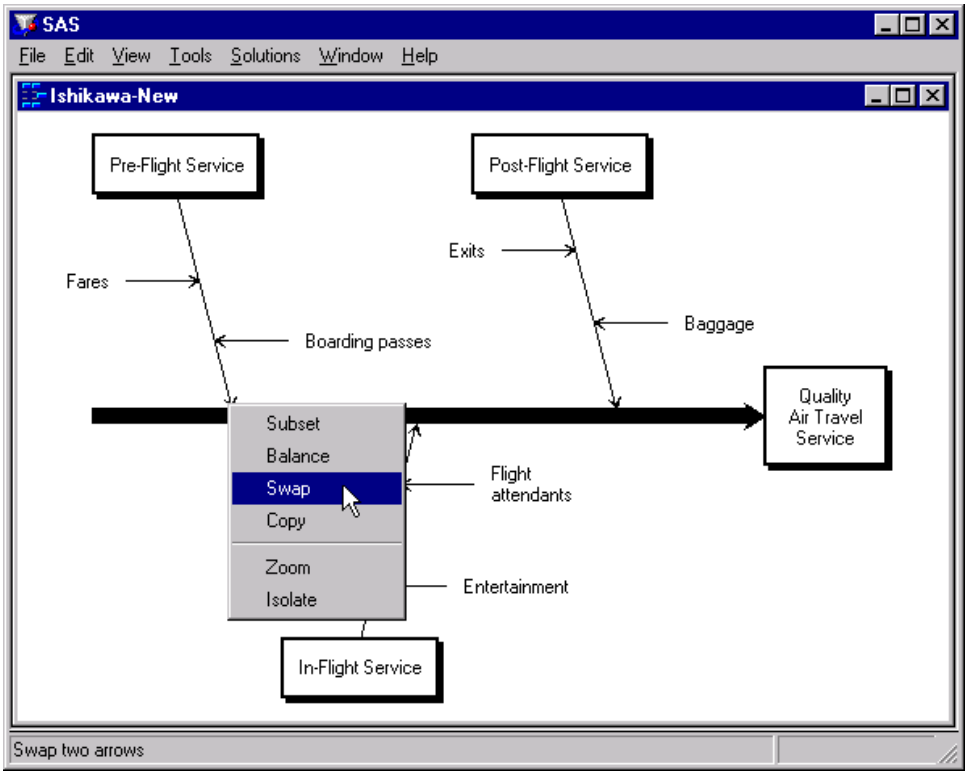
- Move the cursor over the arrow head (tail) of one of the arrows to be swapped and select **Swap** from the context-sensitive popup menu.
- Complete the swap by using the mouse to select the comparable end (head or tail) of the second arrow.

To cancel a swap after you have selected the first arrow, click in a background area of the diagram.

Example

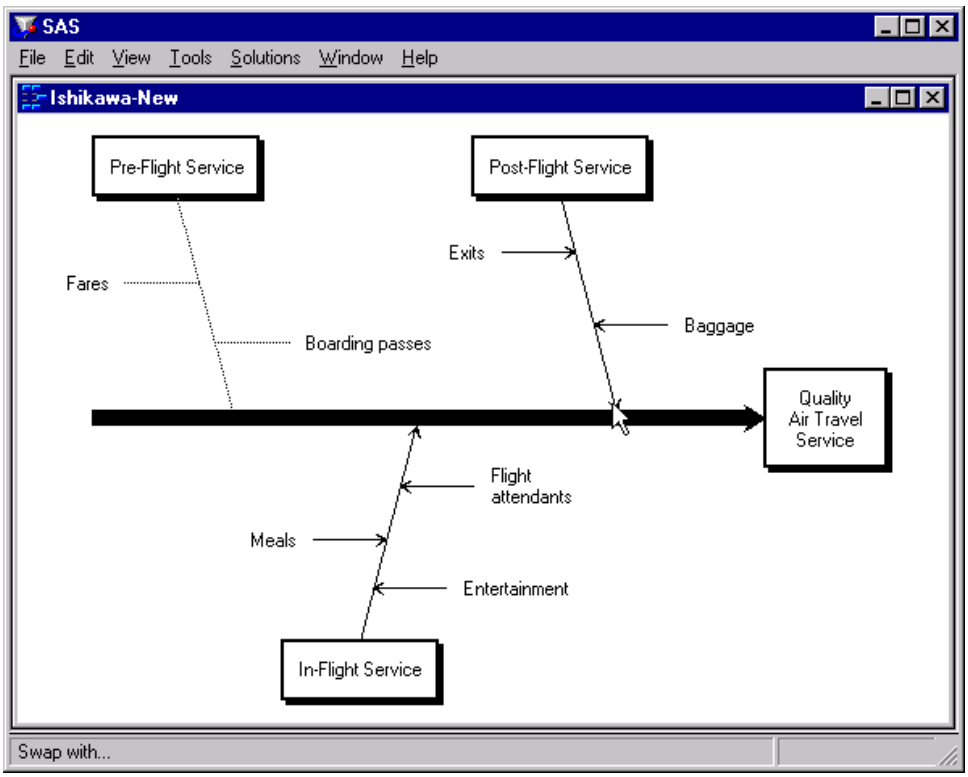
To swap the branch labeled *Pre-Flight Service* (and all its descendants) with the branch labeled *Post-Flight Service* in the following diagram, move your cursor over the arrow head of the *Pre-Flight Service* branch and activate the popup menu using the right mouse button. Select **Swap** to begin the operation.

Figure 9.32 Swapping Two Arrows



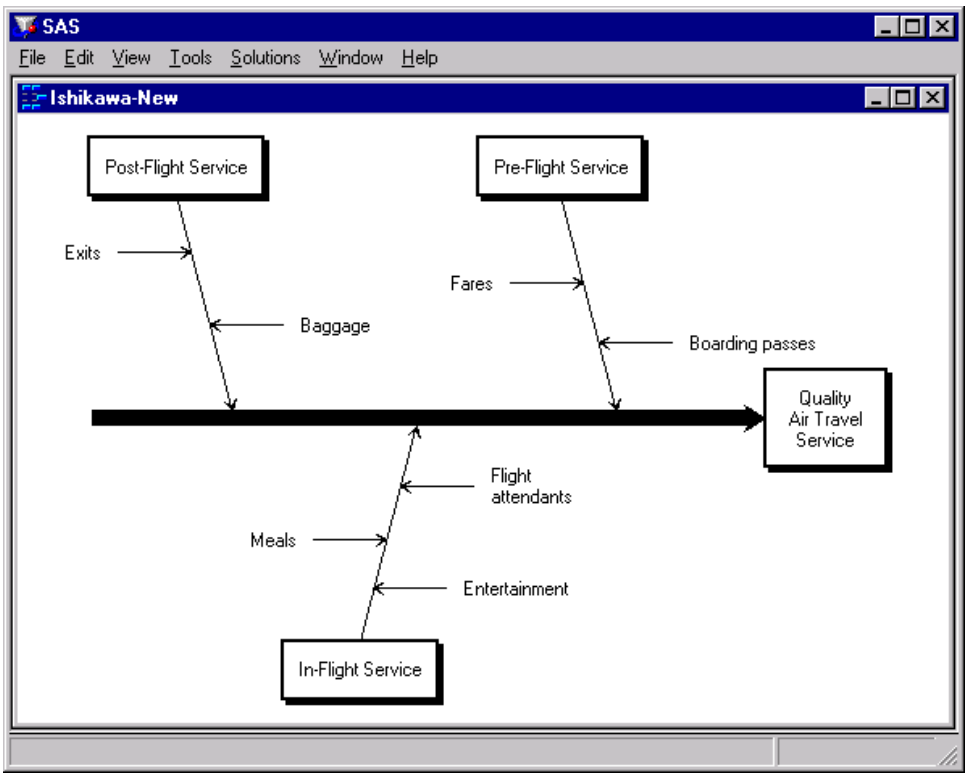
To complete the swap, select the arrow head of the *Post-Flight Service* branch.

Figure 9.33 Swapping Two Arrows *(continued)*



The completed diagram illustrates how the swap operation simplifies interchanging two arrows.

Figure 9.34 Completing a Swap



An alternative to swapping the arrows is to move them. However, moving arrows in this situation requires more steps and tends to be more cumbersome than swapping.

Balancing Arrows

An Ishikawa diagram is said to be *balanced* if the sub-arrows attached to each arrow are equally spaced.

To balance the immediate descendants of an arrow *and all its descendants*, proceed as follows:

- Move the cursor over the arrow head.
- Activate the popup menu using the right mouse button.
- Select **Balance**.

To balance only the immediate descendants of an arrow, select **Balance** from the popup menu for the arrow tail.

You can restore the arrows to their original positions by doing the following:

- Activate the background popup menu using the right mouse button.
- Select **Unbalance**.

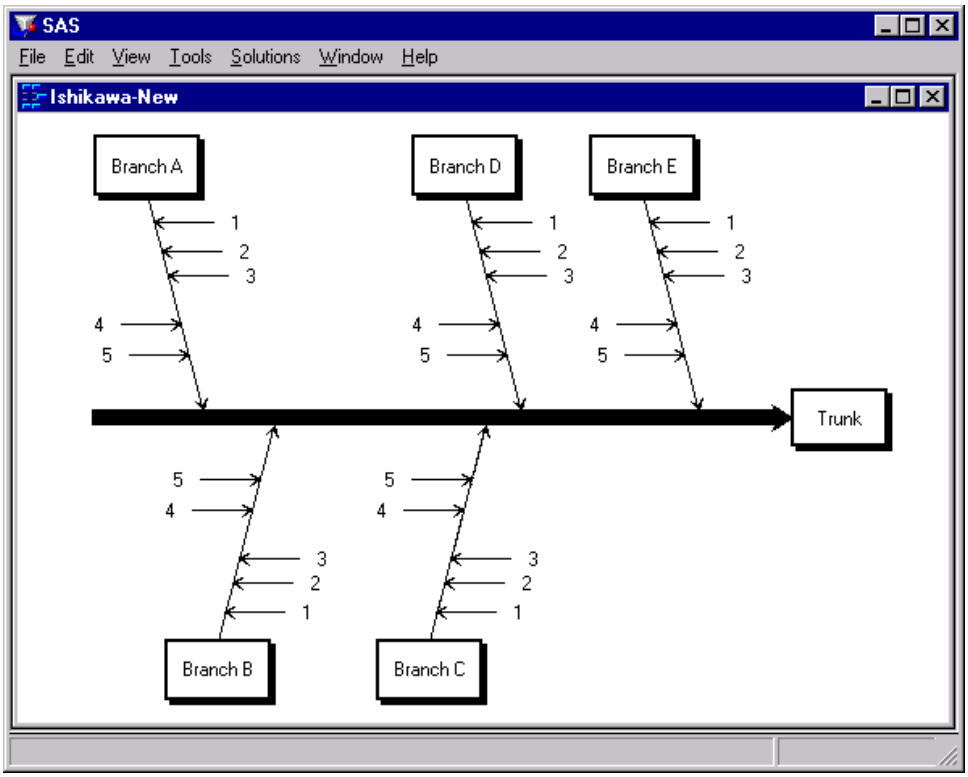
The ISHIKAWA environment provides three alternative methods for balancing arrows. Select one of the following choices from the **View ► Ishikawa Setting ► Balance Method ►**menu:

- **Preserve order/sides** maintains the order and directions of the sub-arrows but repositions them so they are evenly spaced.
- **Preserve order/alternate sides** maintains the ordering of the arrows but repositions adjacent arrows so that they appear on opposite sides. This is the default.
- **Preserve sides** maintains the side on which the sub-arrows are attached then spaces each side of the arrow independently.

Example

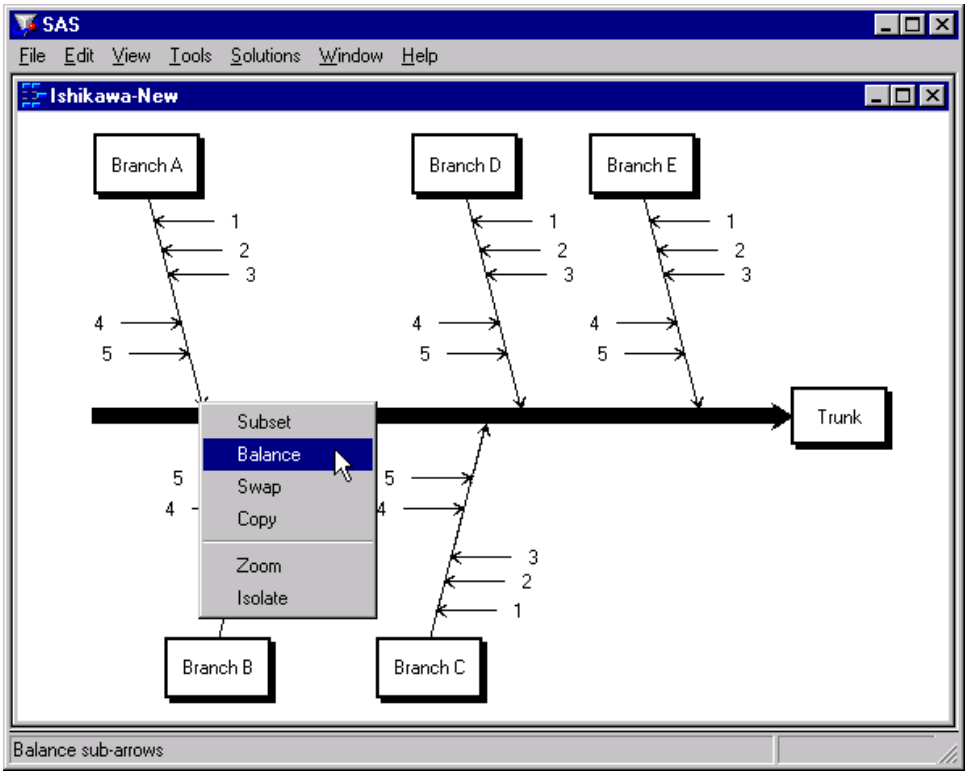
Consider the following unbalanced diagram:

Figure 9.35 An Unbalanced Ishikawa Diagram



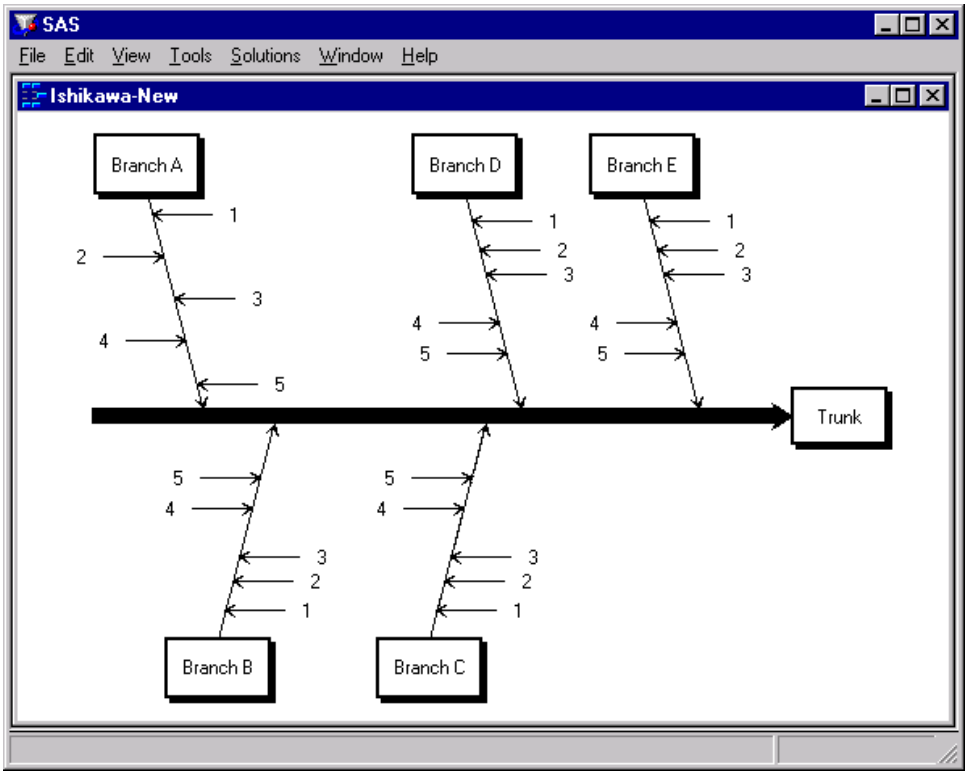
To balance only the stems of the branch labeled *Branch A*, move the cursor over the arrow head and press the right mouse button.

Figure 9.36 Balancing a Branch



Select **Balance** from the arrow head popup menu.

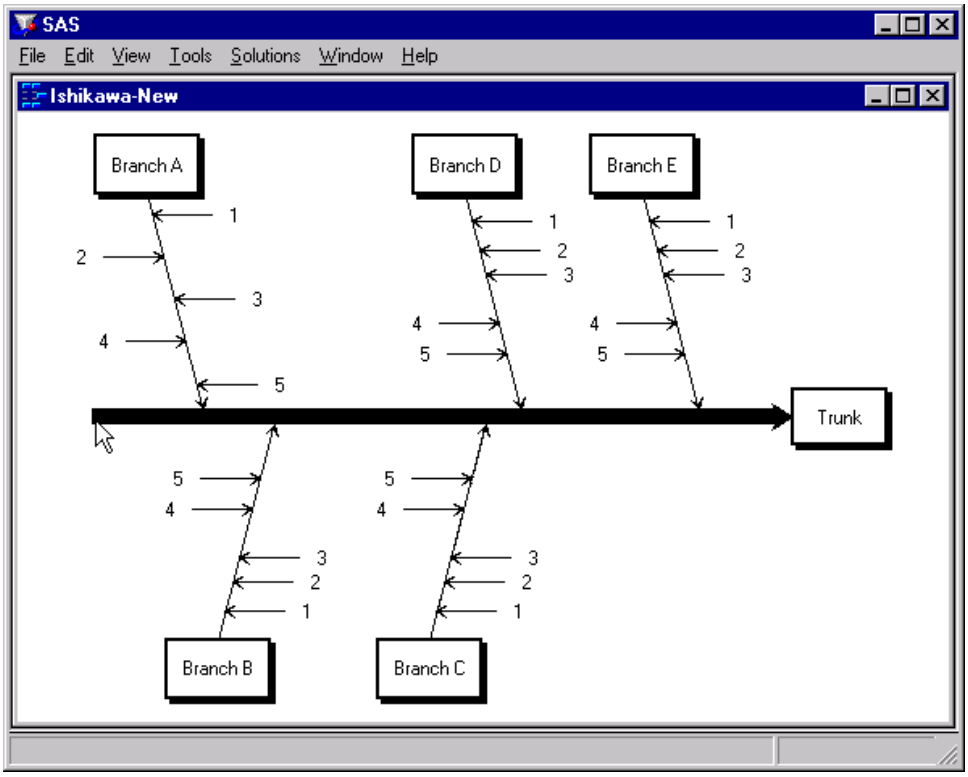
Figure 9.37 A Balanced Branch



Note that since the stems are without leaves, selecting either the head or the tail has the same result.

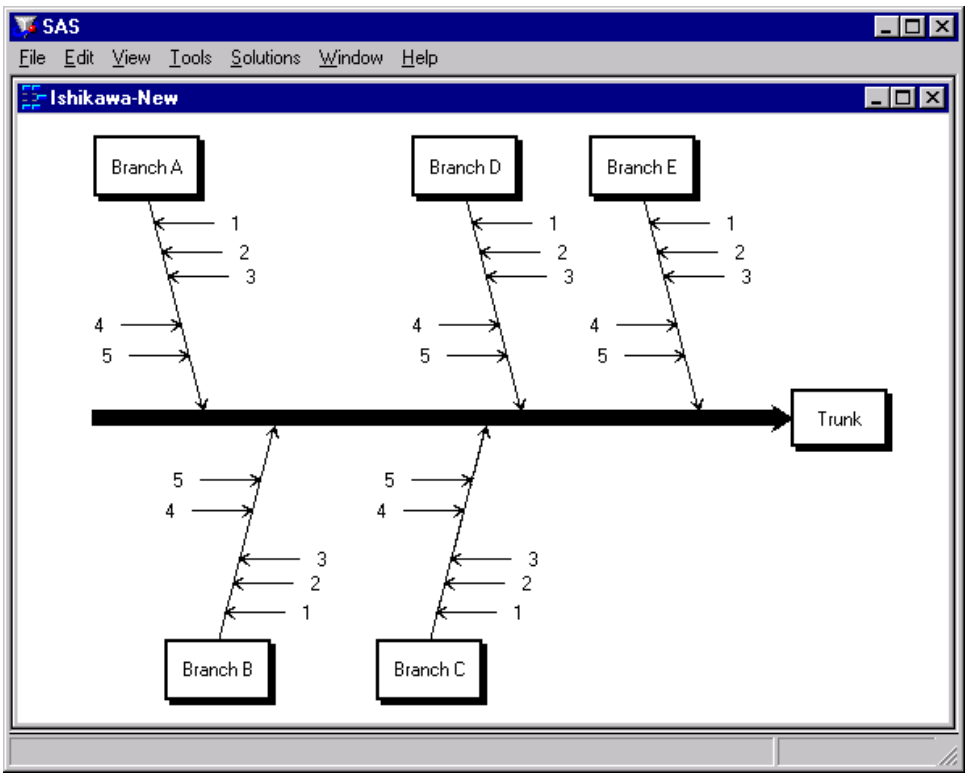
To balance only the five major branches in the preceding diagram without affecting their stems, move the cursor to the tail end of the trunk and select **Balance** from the popup menu.

Figure 9.38 Balancing Only the Branches



To balance the entire diagram (from head to tail, so to speak), move the cursor to the head of the trunk and select **Balance** from the popup menu.

Figure 9.39 Balancing the Entire Diagram

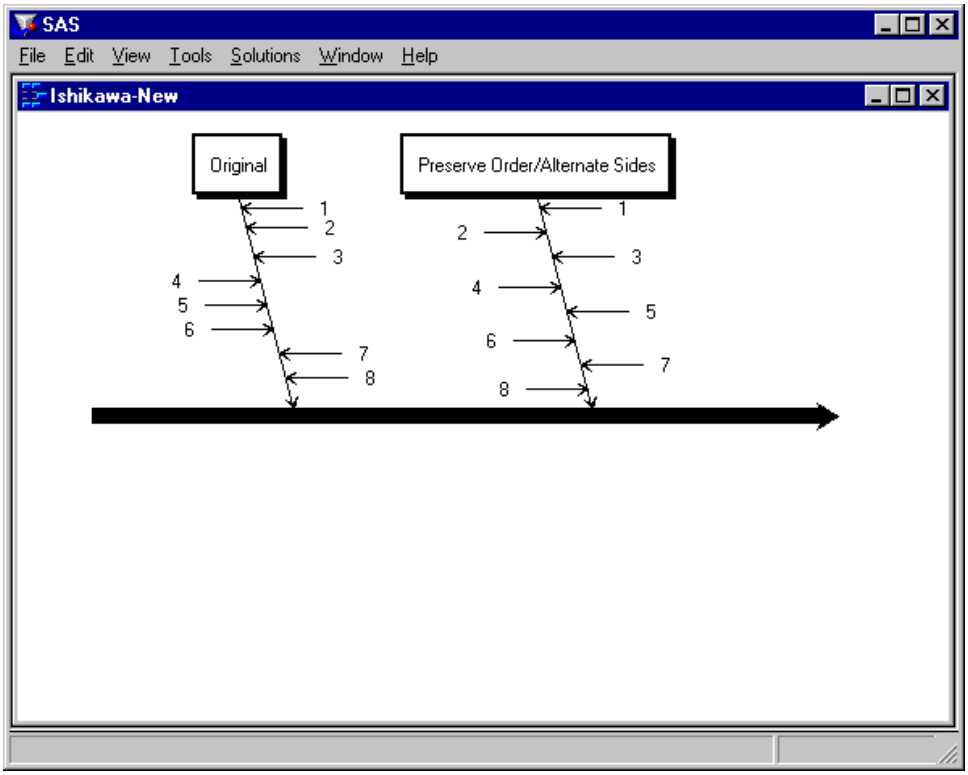


Note that the balancing method used here not only changes the spacing of the stems but reflects them as needed to achieve a balanced appearance. You can control this by specifying a balancing method, as illustrated by the next example.

Example

The following diagram displays an unbalanced branch and a copy of that branch after it was balanced using the **Preserve order/alternate sides** balancing method:

Figure 9.40 Preserving Order But Alternating Sides

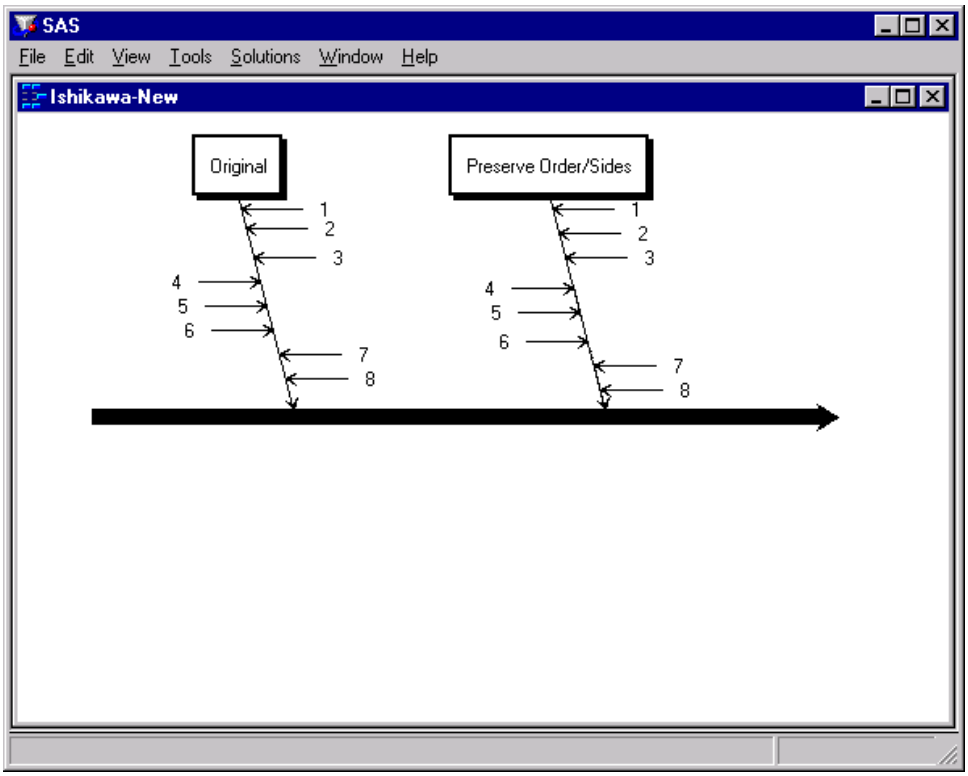


Note that the stems remain in order (*1-8*) from tail to head, but they now alternate evenly across both sides of the branch. This is the default method used for balancing arrows.

Example

The following diagram displays an unbalanced branch and a copy of that branch after it was balanced using the **Preserve order/sides** method:

Figure 9.41 Preserving Order and Sides

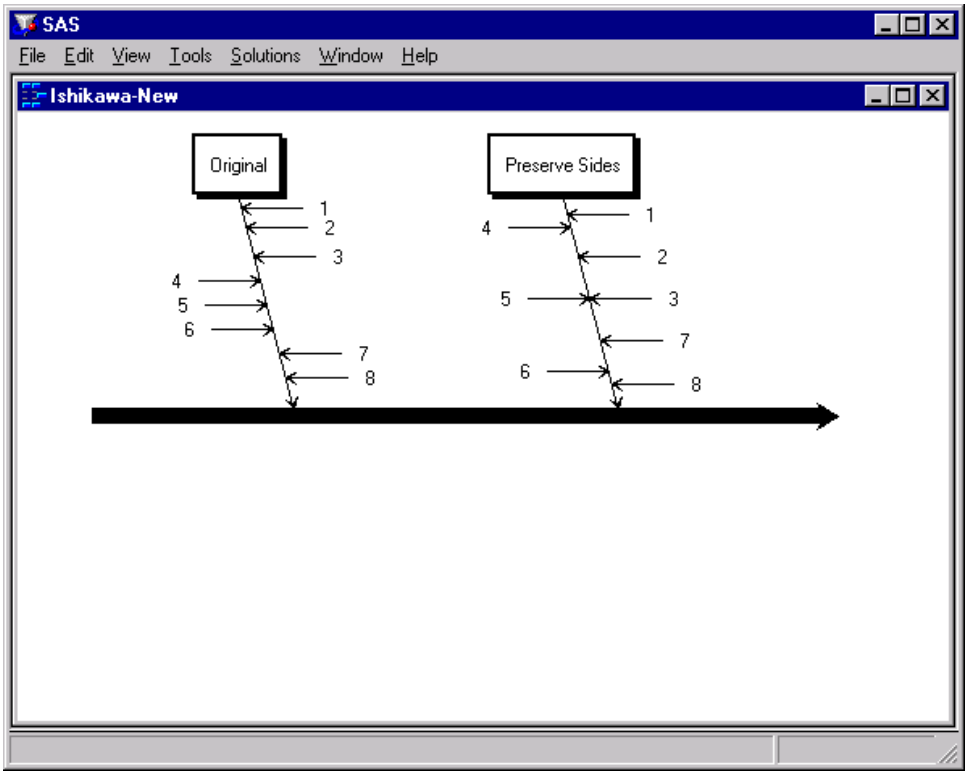


Note that stems *4-6* remain on the left, stems *1-3* and *7-8* remain on the right, and the order from tail to head is still *1-8*. However, the stems are now spaced uniformly.

Example

The following diagram displays an unbalanced branch and a copy of that branch after it was balanced using the **Preserve sides** balancing method:

Figure 9.42 Preserving Sides



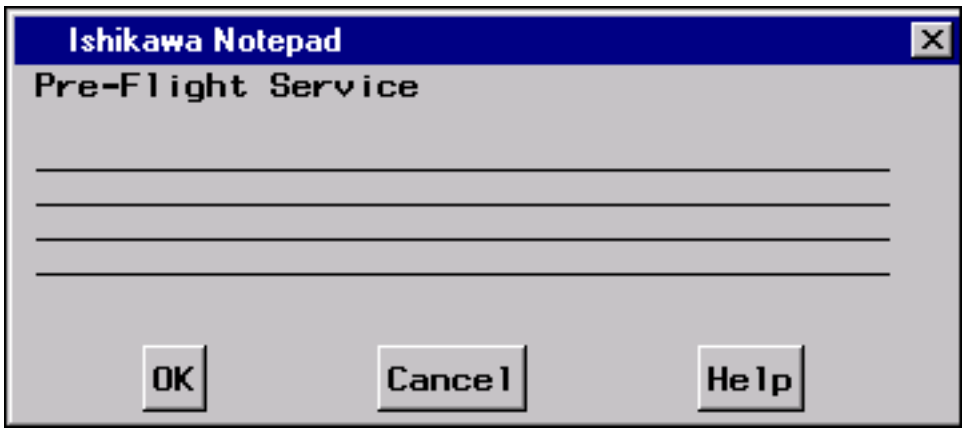
Note that the stems on the left (*4-6*) are spaced uniformly, and the stems on the right (*1-3* and *7-8*) are spaced uniformly. The two sides are spaced independently of each other.

Notepads

Ishikawa (1982) and Kume (1985) advocate the display of quantitative information with the arrows in an Ishikawa diagram.

In the ISHIKAWA environment, you can use *Notepad* windows to record or display information associated with each arrow. To open the Notepad window, move the cursor over the arrow tail and double click.

Figure 9.43 Ishikawa Notepad



Notes are limited to four lines of text with no more than 40 characters per line.

When you save your Ishikawa diagram, your notes are saved with the SAS data set.

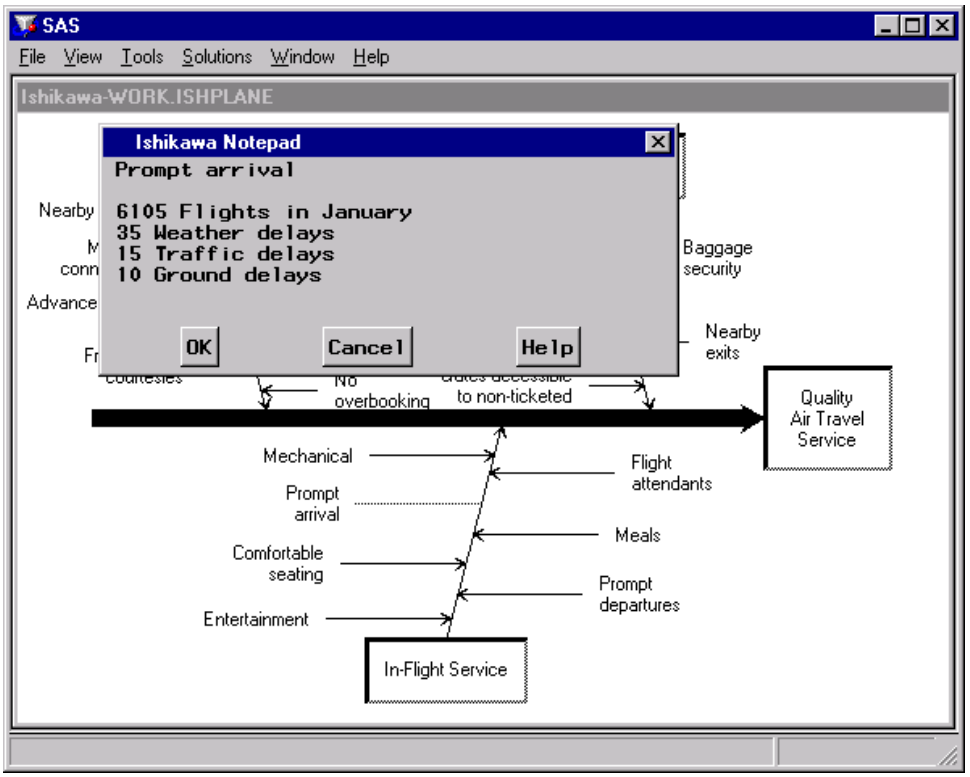
Later, when you retrieve your diagram, all the notes are restored.

You must close the *Notepad* window before you continue working in the ISHIKAWA environment.

Example

In the following figure, double clicking on the *Prompt arrival* stem reveals details about prompt arrival times:

Figure 9.44 Using Notepads to Organize Details



Managing Complexity

A major advantage of the ISHIKAWA environment is that you can quickly organize a highly complex diagram. However, not everyone may be interested in seeing all the details—at least initially.

To increase the level of detail by one level, do the following:

- Move the cursor to a background area of the window, and use the right mouse button to activate the background popup menu.
- Select **> Detail**. On some hosts, you can press the > key instead of using the popup menu (as long as you are not editing text).

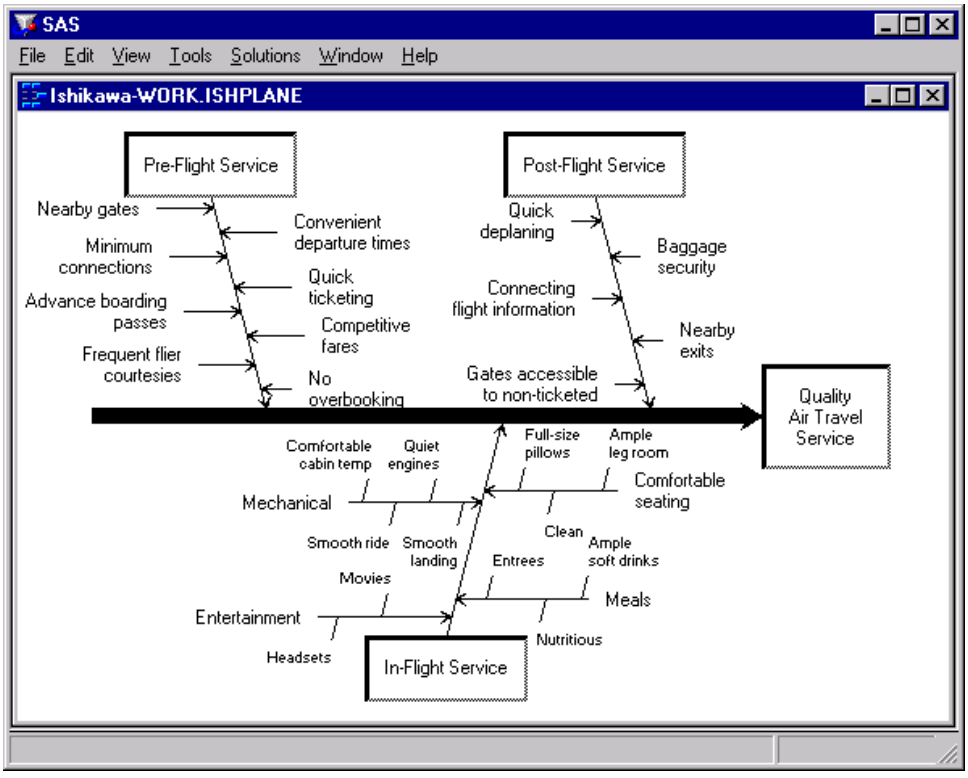
Each time you select **> Detail** from the background popup menu, the detail increases by one level.

To reverse the process and decrease the level of detail, select **< Detail** from the popup menu, or press the < key.

Example

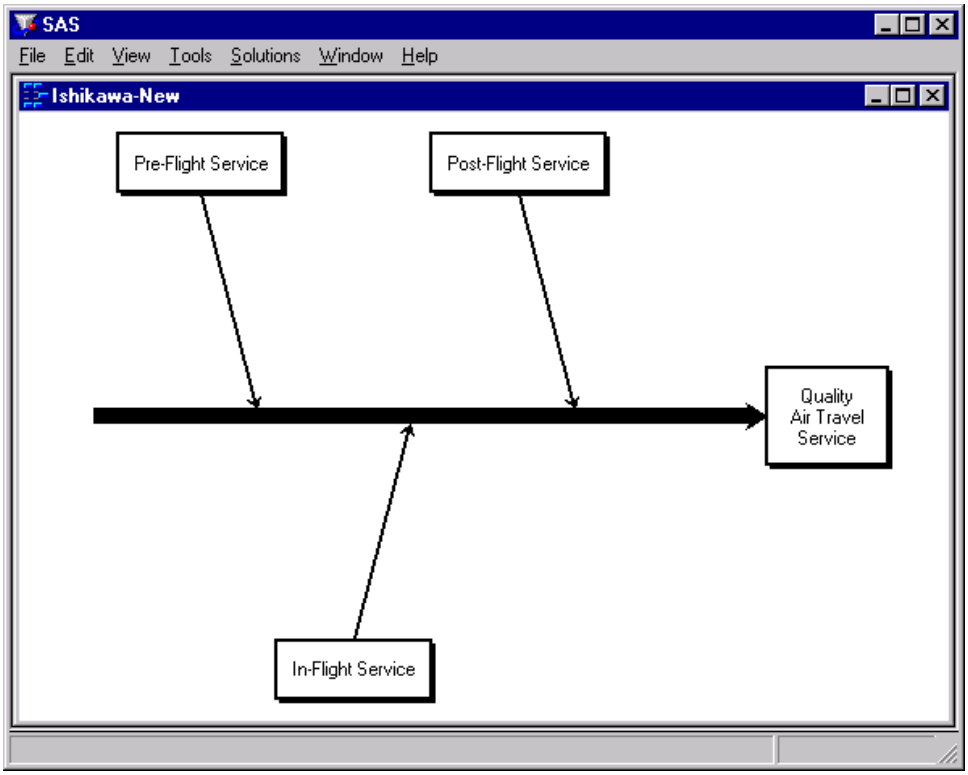
You are making an online presentation about factors that influence the quality of air travel service. The following diagram presents too many details to be a good starting point for your audience:

Figure 9.45 Highly Detailed Ishikawa Diagram



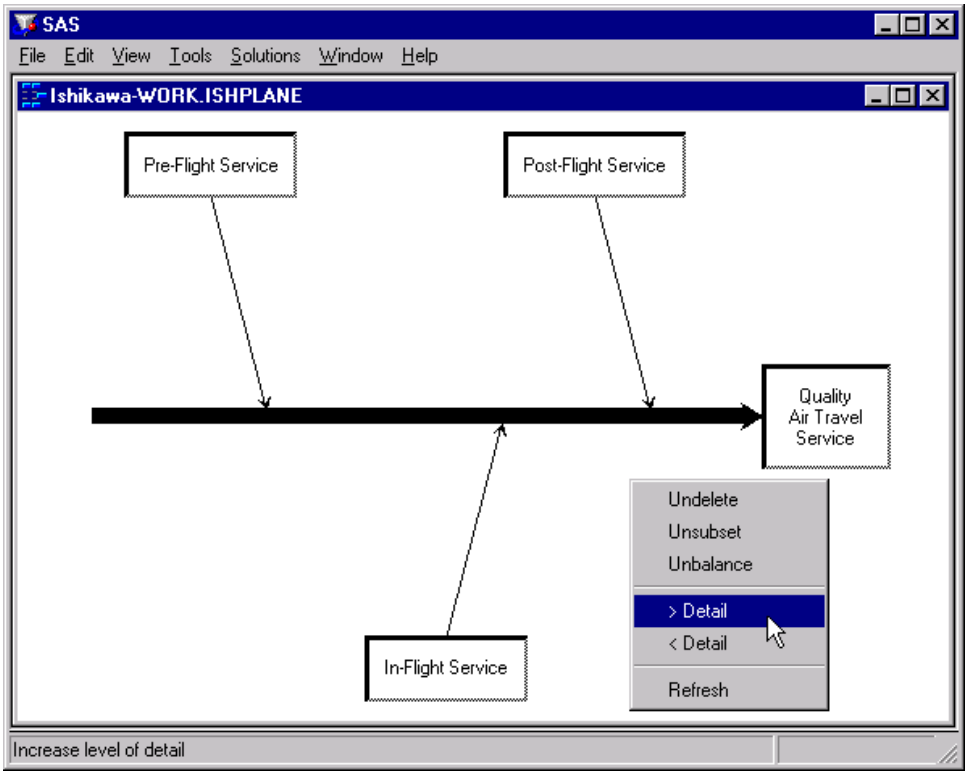
A better way to begin is by displaying only the trunk and branches.

Figure 9.46 Branch-Level Diagram



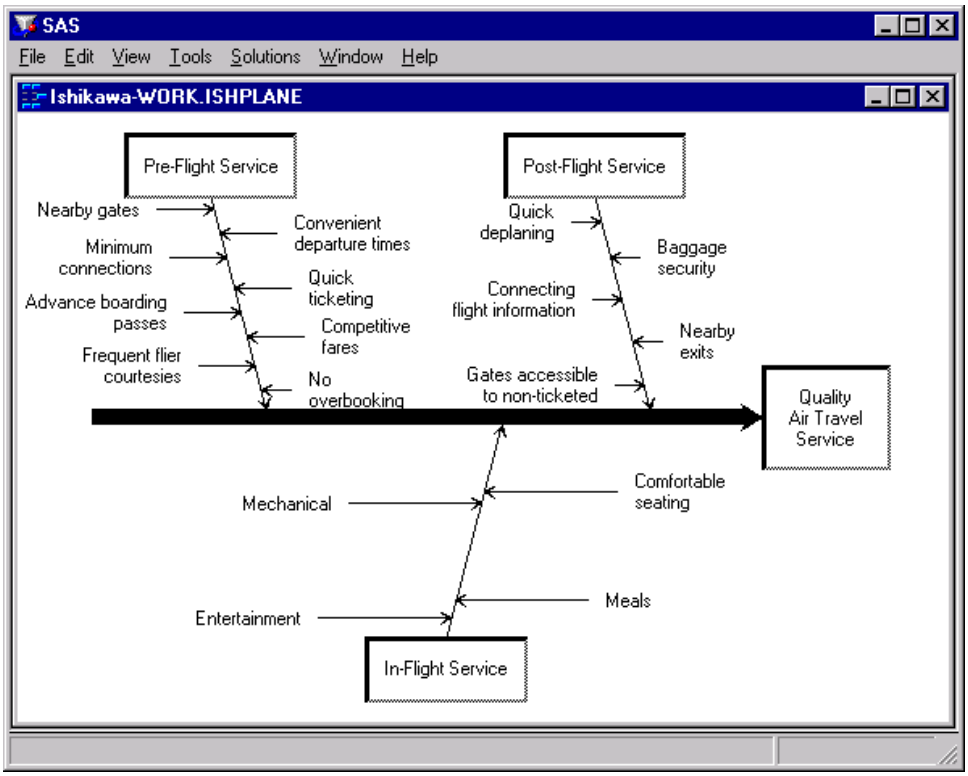
Then, at the next stage of your presentation, dynamically foliate the branches with stems, as follows:

Figure 9.47 Increasing the Level of Detail



The amount of detail is increased by one level.

Figure 9.48 Increasing the Level of Detail



Zooming Arrows

A second method for managing a highly detailed Ishikawa diagram is to work with a subsection of the diagram in a separate window. The window and the sub-arrows inside it can be resized independently of the parent window. In all other respects, the information in the two diagrams is linked dynamically. Changes in one window (for instance, moving, adding, and editing arrows) are reflected in the other window.

To zoom an arrow, proceed as follows:

- Move the cursor over the arrow head.
- Activate the popup menu using the right mouse button.
- Select **Zoom**.

To return or *unzoom*, select **File ► Close**.

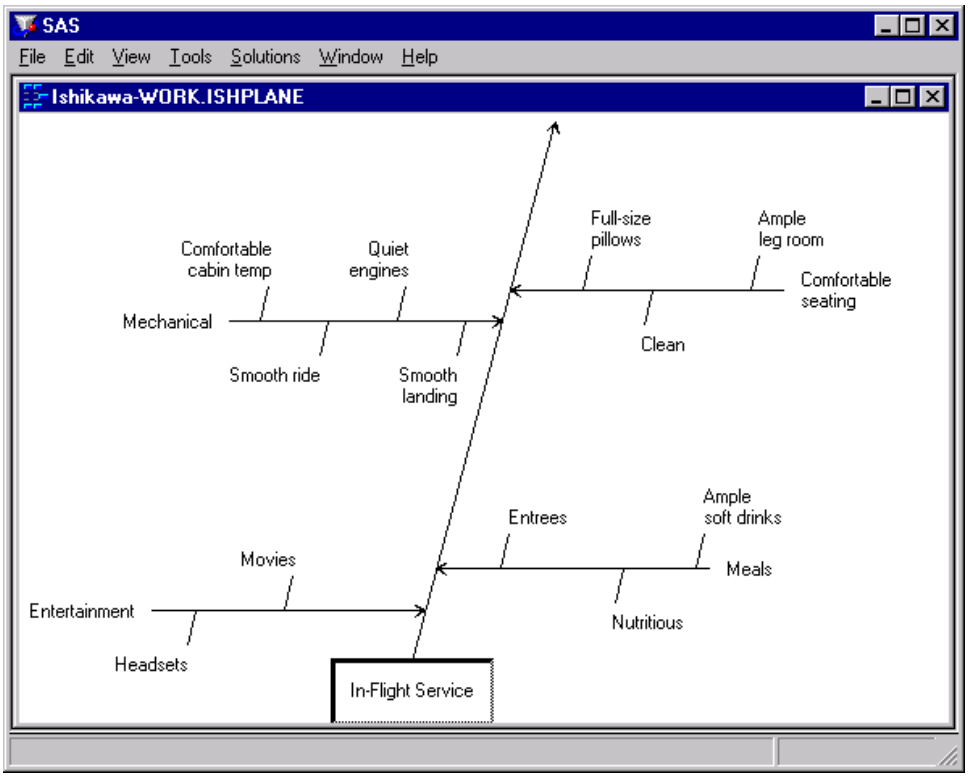
You can have up to four windows open at one time.

To reduce the amount of window management, you can specify that zoomed diagrams are to be displayed in the current window rather than in new windows by setting **Zoom Window** to *Current* in the **View ► Ishikawa Settings ► Other…** dialog.

Example

The following figure shows a branch labeled *In-Flight Service* after it has been zoomed into a new window:

Figure 9.49 Zooming a Branch



Isolating Arrows

A third method for managing a highly complex Ishikawa diagram is to view the entire diagram as a collection of smaller diagrams. Any arrow (along with its sub-arrows) can be isolated into a separate diagram in a new window. This diagram can then be easily saved in a separate file.

To isolate a branch as a separate diagram, do the following:

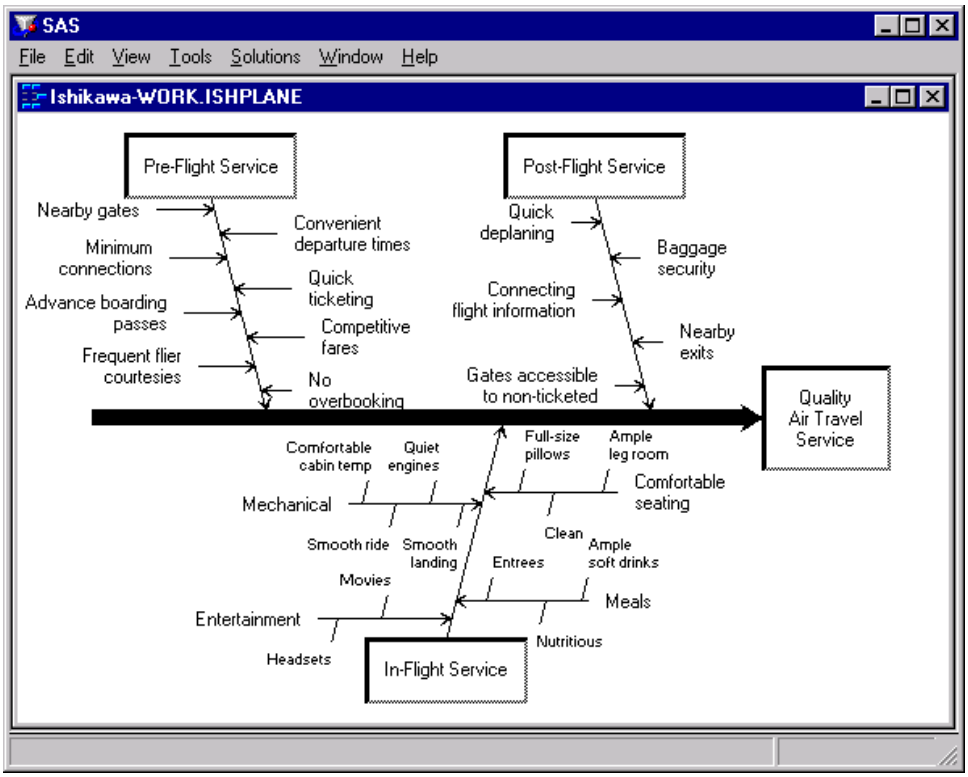
- Move the cursor over the head of the arrow.
- Activate the popup menu using the right mouse button.
- Select **Isolate**.

You can have up to four ISHIKAWA windows open at one time.

Example

Consider the following diagram:

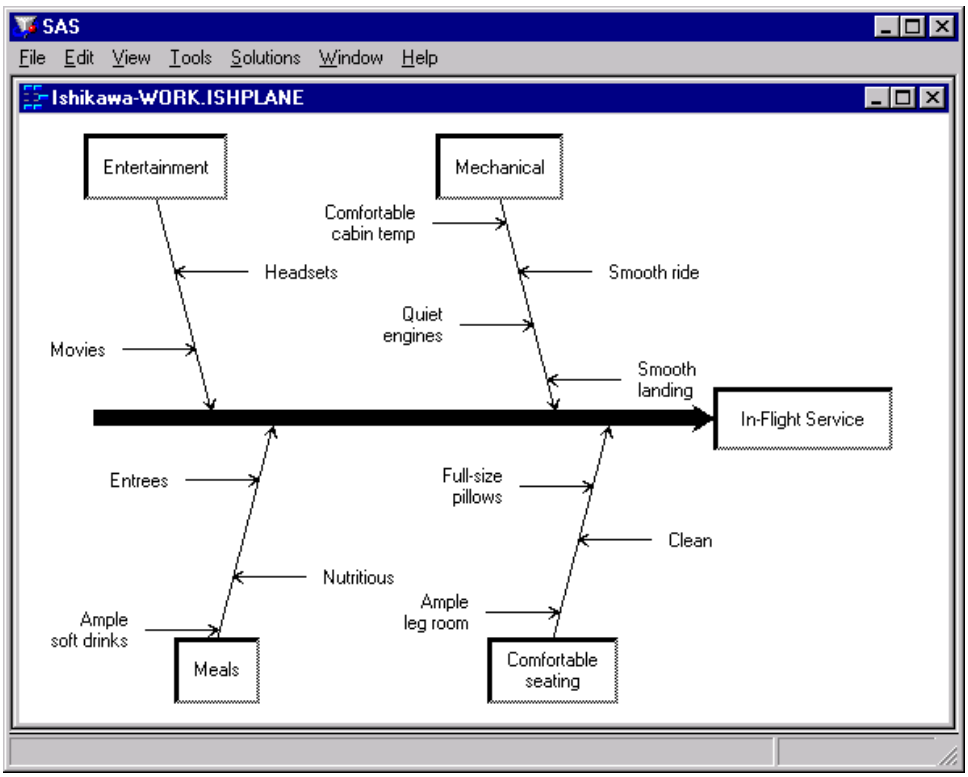
Figure 9.50 A Highly Detailed Diagram



To isolate the branch labeled *In-Flight Service* as a separate Ishikawa diagram, move your cursor over the head of the arrow. Use the right mouse button to activate the popup menu and select **Isolate**.

The following figure shows the main diagram in one window and the branch labeled *In-Flight Service* after it has been isolated to another window:

Figure 9.51 Promoting a Branch into a New Diagram



To return to the original diagram, select **File ► Close**.

Merging Diagrams

You can combine multiple Ishikawa diagrams into a *master* diagram by using the merge operation. To merge a stored diagram into the current diagram, proceed as follows:

- Select **File ► Merge**.
- Specify the name of a SAS data set that contains a saved Ishikawa diagram.

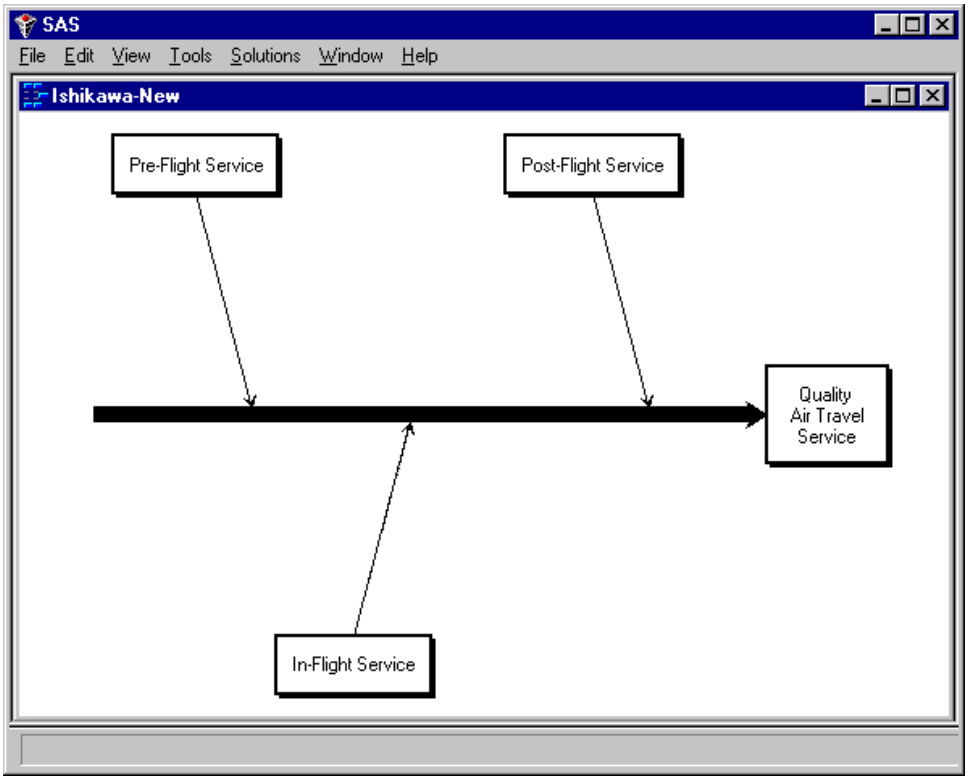
Another way to combine diagrams is to open separate ISHIKAWA windows for each sub-diagram then copy them into the master diagram. To copy all or part of an Ishikawa diagram from one window to another, do the following:

- Move the cursor over the head of the arrow.
- Activate the popup menu using the right mouse button.
- Select **Copy**.
- Position the cursor slightly to one side of the new attachment point and click (just as though you are adding a new arrow).

Example

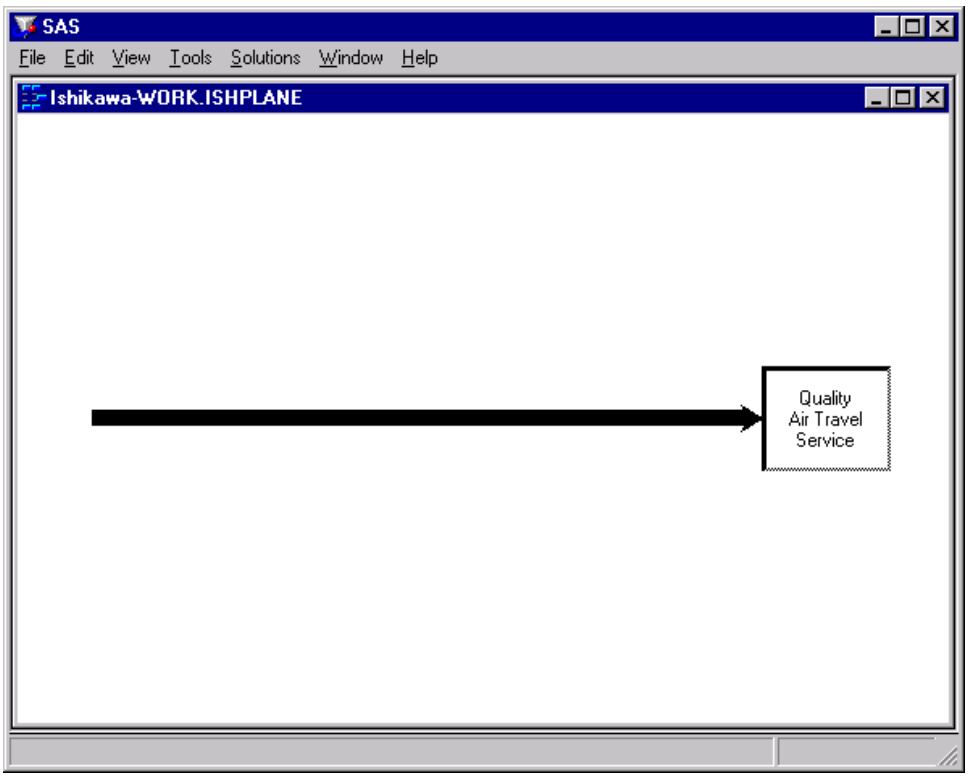
Suppose you want to create the following diagram by combining information from diagrams already created by each of the major service areas (Pre-Flight, In-Flight, and Post-Flight) and stored in different SAS data sets:

Figure 9.52 A Completed Master Diagram



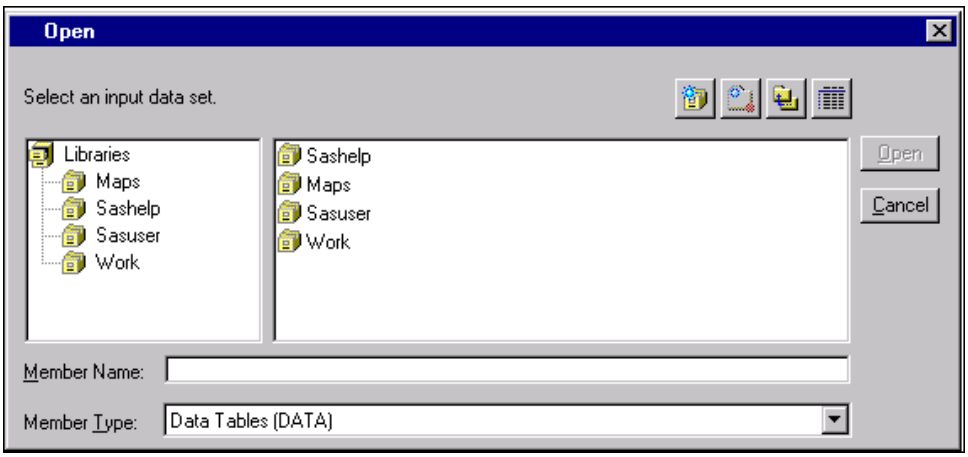
First, use the ISHIKAWA environment to create the trunk for the new master diagram.

Figure 9.53 Starting a Master Diagram



Select **File ► Merge** from the command bar to open the File Requestor dialog.

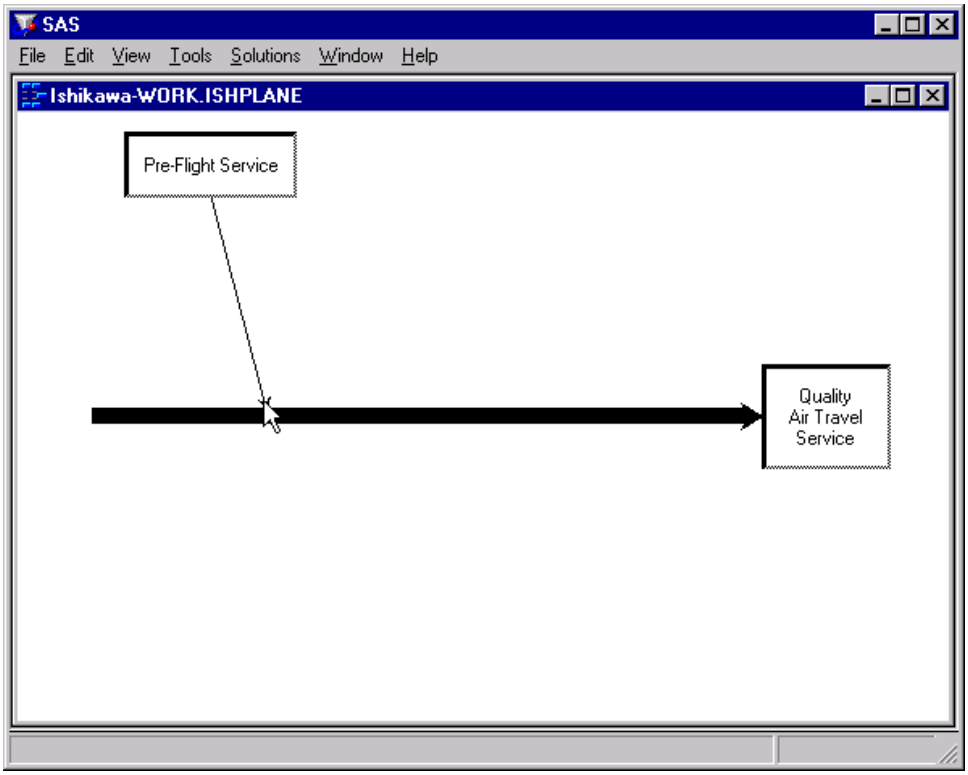
Figure 9.54 Member Selector



Specify the name of the data set for *Pre-flight services* and press **Open**.

Now click on a point along the trunk where this sub-diagram is to attach.

Figure 9.55 Constructing a Master Diagram



To complete the diagram, repeat the process for the remaining branches.

Creating Graphics Output Using SAS/GRAPH Software

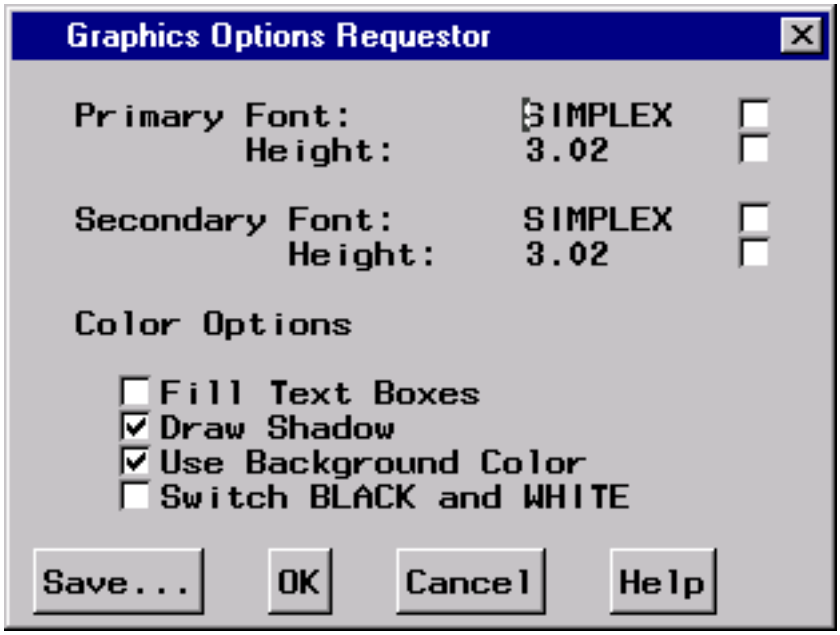
One way to create a hard copy of your Ishikawa diagram is to send it to a graphics device using SAS/GRAPH software. To do this, you should submit a GOPTIONS statement to direct the graphics output to the appropriate location and control the output format *before you invoke the ISHIKAWA environment*. For example, the following GOPTIONS statement directs the output to a PostScript device:

```
goptions target=psl noprompt;
```

If you do not specify a target device before invoking the ISHIKAWA environment, you will be prompted for one before the graph is generated.

In the ISHIKAWA environment, when you are ready to route your output to a hard copy device, select **File ► Save as ► Graph**. This opens a dialog that enables you to customize various aspects of your graph.

Figure 9.56 Hard Copy Requestor



To save the diagram to the default graphics catalog in the WORK library (WORK.GSEG), simply press **OK** and close the dialog. The default member name is ISHIKAWA.

To save the diagram to a different graphics catalog, select **Save...** and then use the Member selector window to specify a library, a SAS catalog, and a member name.

When sending a diagram directly to an output device, you can ignore the member name entirely.

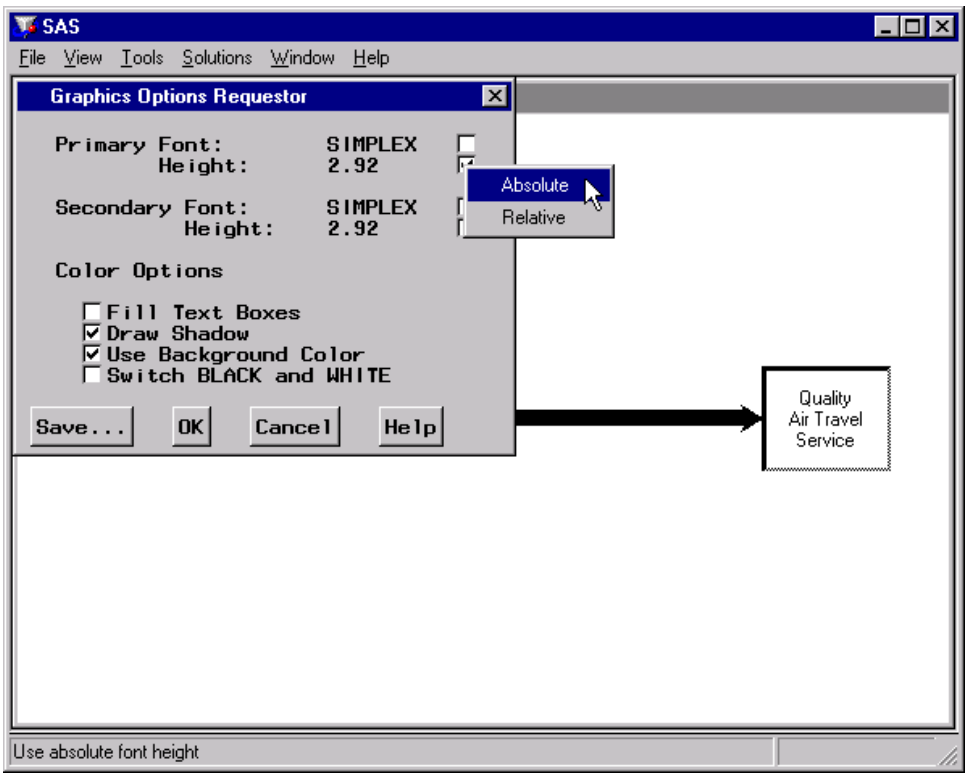
To save to your own graphics catalog, select **Save...** and then use the Save a member selection window to specify a catalog and data set name. Simply select **OK** when you want to save your diagram to the default graphics catalog (WORK.GSEG). When sending a diagram directly to an output device, you can use **OK**.

You must specify two SAS/GRAPH fonts for drawing the labels in the hard copy of the diagram. The hardware fonts used in the ISHIKAWA environment cannot be used for your hard copy. The *primary* font and size are used for the first three levels of text. The *secondary* font and size are used for the remaining levels of text.

To change fonts, enter a valid SAS/GRAPH font name in the font field or click on the button to the right of the font field to display a font requestor dialog. The default font is SIMPLEX.

You can specify the height of the text directly in the height field (in screen percent units), or you can click on the button to the right of the field to request an *absolute* height or a *relative* height.

Figure 9.57 Font Height Selector



Select **Absolute** when you want the font height in the output to be the same height as the font height used in the ISHIKAWA environment even if the output window and the ISHIKAWA window differ in size. Select **Relative** to maintain the same font height to window size proportion in both the ISHIKAWA window and the output window. The numeric value entered in the height field after either choice is a screen percent unit. The default text height is *absolute*.

Use the **Fill Text Boxes** and **Draw Shadow** check boxes to suppress the box fills and box shadows from the output. They cannot be used to *add* these features to the hard copy if they were not present in the ISHIKAWA window.

Use the **Use background color** check box to indicate whether the background color from the ISHIKAWA environment is used in the output. This option is useful when you are sending your diagram to a *color* device and you want the background in your hard copy to match that of your ISHIKAWA environment.

Use the **Switch Black and White** check box to interchange black and white when the diagram is sent to the output device. This option is useful when you send your diagram from a white-on-black display to a black-on-white hard copy device.

Click on **OK**to generate the hard copy output or click on **Cancel**to quit.

Creating Bitmap Graphics Output

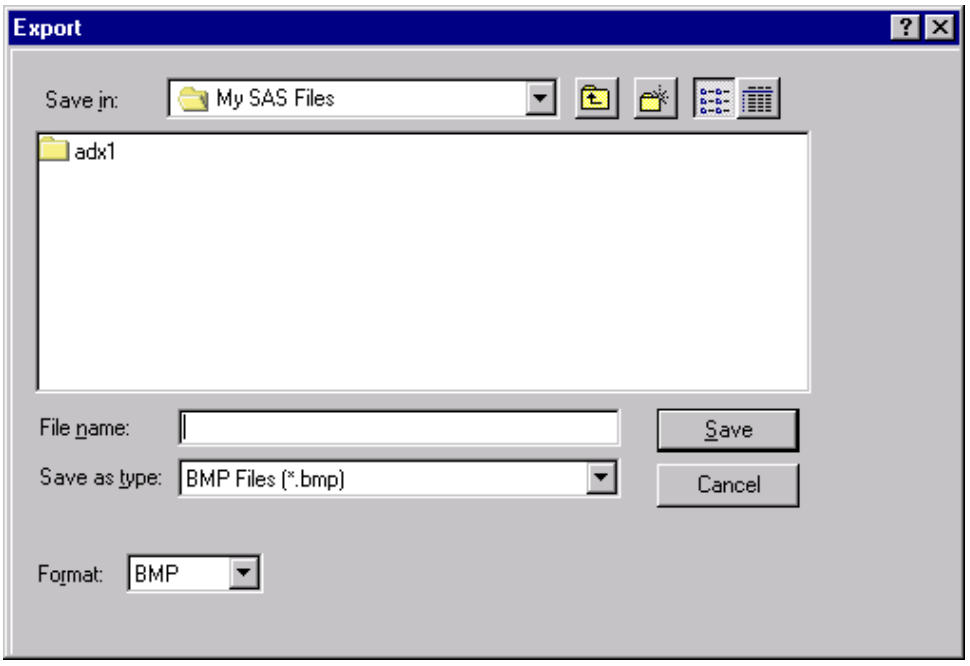
A second way to create a hard copy of your Ishikawa diagram is to export it as a bitmap to one of the following:

- the host graphical clipboard
- an external bitmap file
- a SAS/GRAPH Image catalog entry

To copy the Ishikawa diagram as a bitmap to the host clipboard, select **Edit ► Copy**. The results are host specific. For more details about copying to the host clipboard on your system, consult the SAS companion for your host.

To export the Ishikawa diagram to a bitmap file using SAS/GRAPH software, select **File ► Export as Bitmap ► File...**.

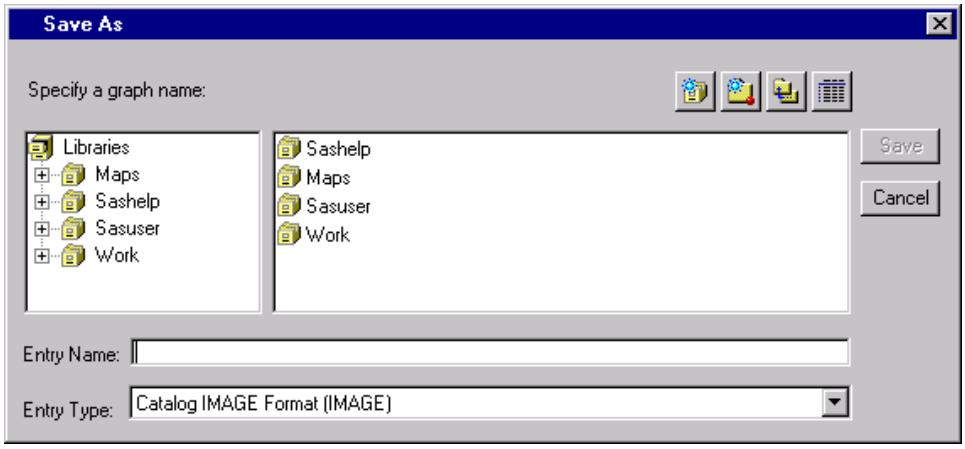
Figure 9.58 Export File Requestor



The appearance of this dialog will be host specific. For more details about the format of this dialog on your system, consult the SAS companion for your host.

To save the Ishikawa diagram as an IMAGE entry in a SAS catalog, select **File ► Save as ► Image**.

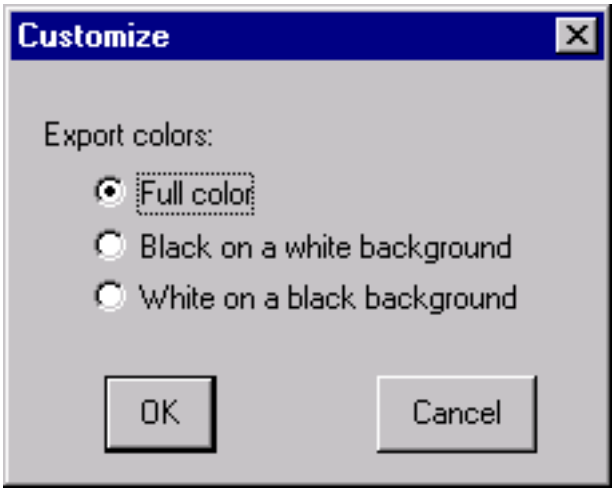
Figure 9.59 Entry Selector



You must specify a SAS catalog in which to save the IMAGE entry as well as a name for the object.

When exporting your diagram to a bitmap or saving to a SAS/GRAPH IMAGE entry, you can have the colors mapped so that color diagrams are saved in black on white or white on black. You do not have to make those changes to the diagram yourself. Use **File ► Export as Bitmap ► Customize...** to display the following dialog:

Figure 9.60 Customize Export Dialog



Select **Black on white** to convert the output to a black diagram on a white background. This is useful when the diagram is being exported to a document.

Select **White on black** to convert the output to a white diagram on a black background. This is useful when the diagram is being exported for display on a black and white terminal.

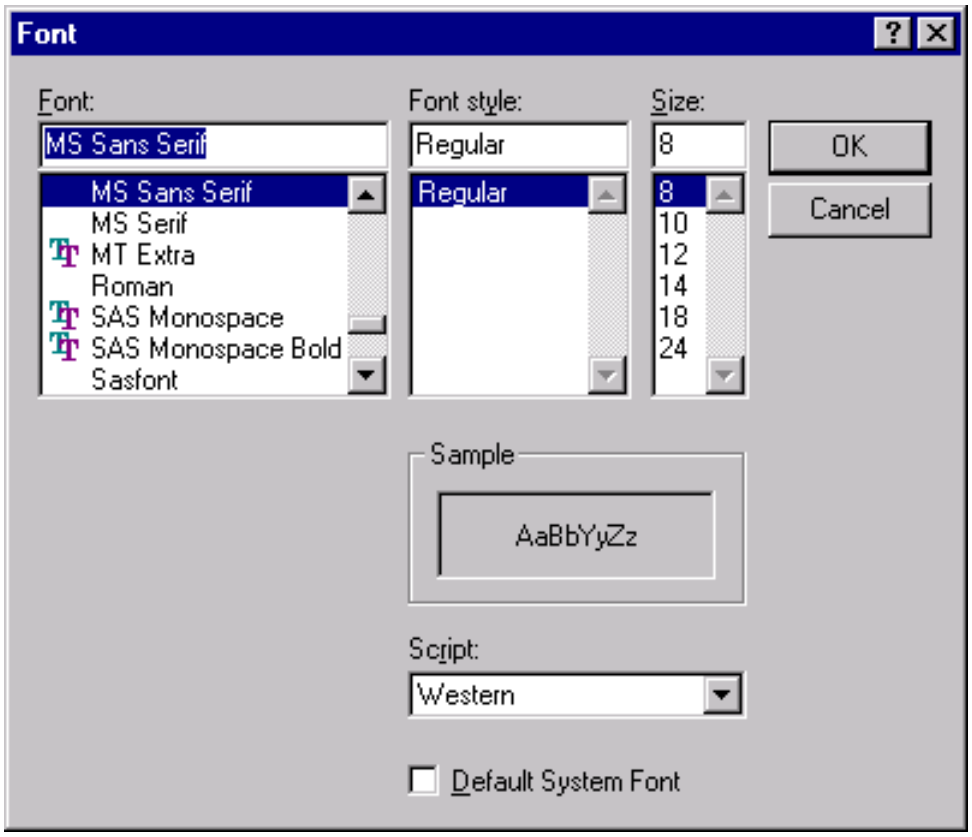
Modifying Fonts

By default, the ISHIKAWA environment uses the same hardware font as the SAS windowing environment. However, you have the option of specifying two different font styles/sizes.

The *primary* font is used for labeling arrows in the first three levels of the diagram. The *secondary* font is used for labeling arrows in the remaining levels. You will typically use a smaller font in the detailed (secondary) areas of the diagram.

To change a font, select **View ► Ishikawa Settings ► Primary Fonts...** or **View ► Ishikawa Settings ► Secondary Fonts...** to display the Font Requestor window, as follows:

Figure 9.61 Font Requestor



The layout of the Font requestor window is host specific. Typically, it will contain a list of available fonts and sizes displayed in a scrollable region. Refer to your host documentation for specific information regarding the format of this dialog.

To change fonts, select a font from the list.

You must close the Font Requestor window before you can proceed. Select **OK**to apply the font or **Cancel**to cancel the dialog.

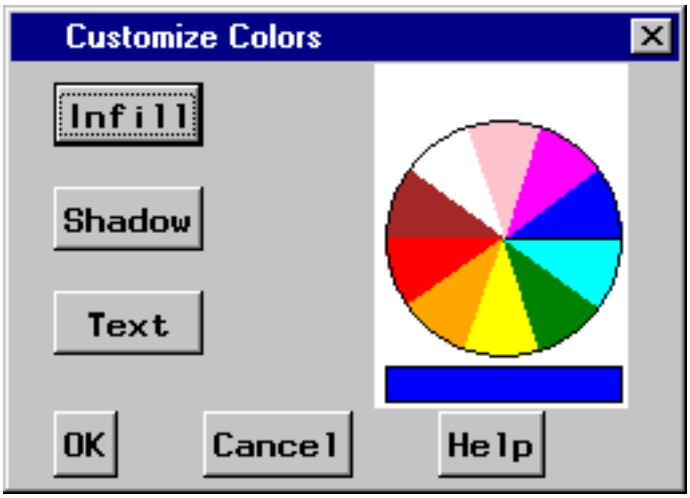
To customize your session so that these fonts are permanently associated with the ISHIKAWA environment, select **View ► Save Attributes**from the command bar.

Modifying Box Colors

By default, the box fill (background) color is empty and the shadow (outline) color is the same as the arrow color.

To modify the colors associated with trunk and branch boxes, select **View ► Ishikawa Settings ► Colors...**. A dialog, similar to the following, is displayed:

Figure 9.62 Colors Dialog



To change the fill color of all the boxes[6] in the Ishikawa diagram, do the following:

- Select a color from the color palette.
- Select **Infill**.

Once modified, the fill color is unaffected by changes in the arrow color. To return the box to an empty fill, proceed as follows:

- Select the current infill color from the color palette (if it is not already the current color).
- Select **Infill**.

To change the shadow color of the boxes, select **Shadow** and follow the same procedure.

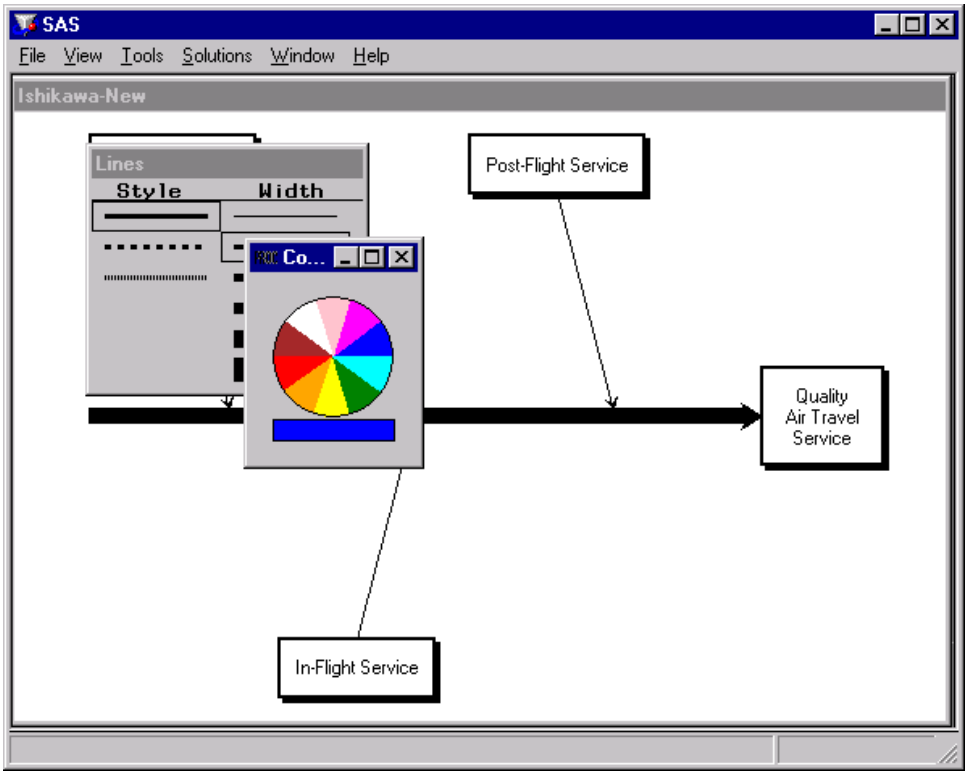
Select **OK**to close the dialog or **Cancel** to cancel the changes.

Modifying Arrow Colors and Line Styles

The ISHIKAWA environment provides a line style palette and a color palette that you can use to customize the arrows in your Ishikawa diagram. Select **View ► Palettes** to activate both palettes.[7]

[6] You cannot directly modify the colors of individual boxes from this dialog.
[7] If you are working on a black-and-white terminal, you should not use the color palette.

Figure 9.63 Line Style and Color Palettes



To specify the arrows to which color and line selections apply, subset them with the subset function. To toggle an arrow in or out of the list of subsetted arrows, do the following:

- Use the right mouse button to display the arrow head or the arrow tail popup menu. To subset an arrow and all its descendants, use the arrow head popup menu. Use the arrow tail popup menu to subset an arrow without any descendants.
- Select **Subset**.

The labels of all subsetted arrows are underlined.

On some hosts, shift-clicking on the arrow head or tail will also subset the arrow. You can subset any combination of arrows in the diagram.

You can change the color of all the subsetted arrows by selecting the desired color in the color palette with the mouse. Likewise, use the line palette to control the style and width of the arrows.

To unsubset all the arrows in the diagram, do the following:

- Move the cursor to a *background* area of the ISHIKAWA window.
- Use the right mouse button to activate the background popup menu.
- Select **Unsubset** from the popup menu.

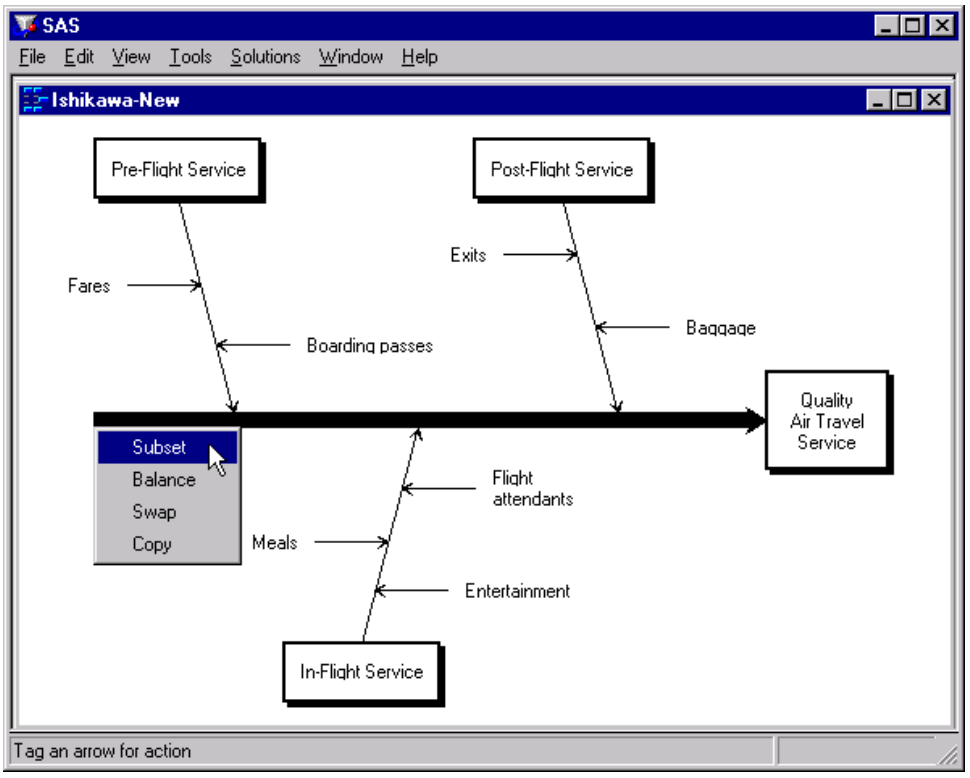
To unsubset a specific arrow in the diagram, select **Subset** from the context-sensitive popup menu for the arrow head or tail.

Be sure to remove all subsets once you have finished modifying the diagram, since subsets affect the focus of many other operations.

Example

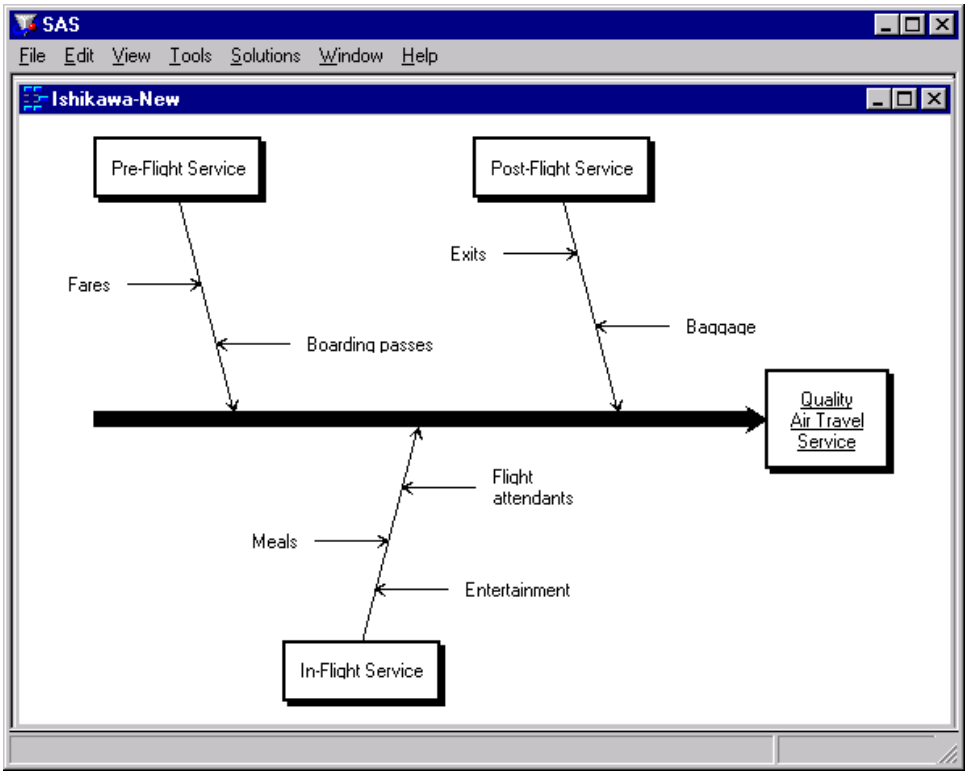
Continuing with the diagram from the previous section, subset the trunk using the arrow tail popup menu.

Figure 9.64 Subsetting Only the Trunk



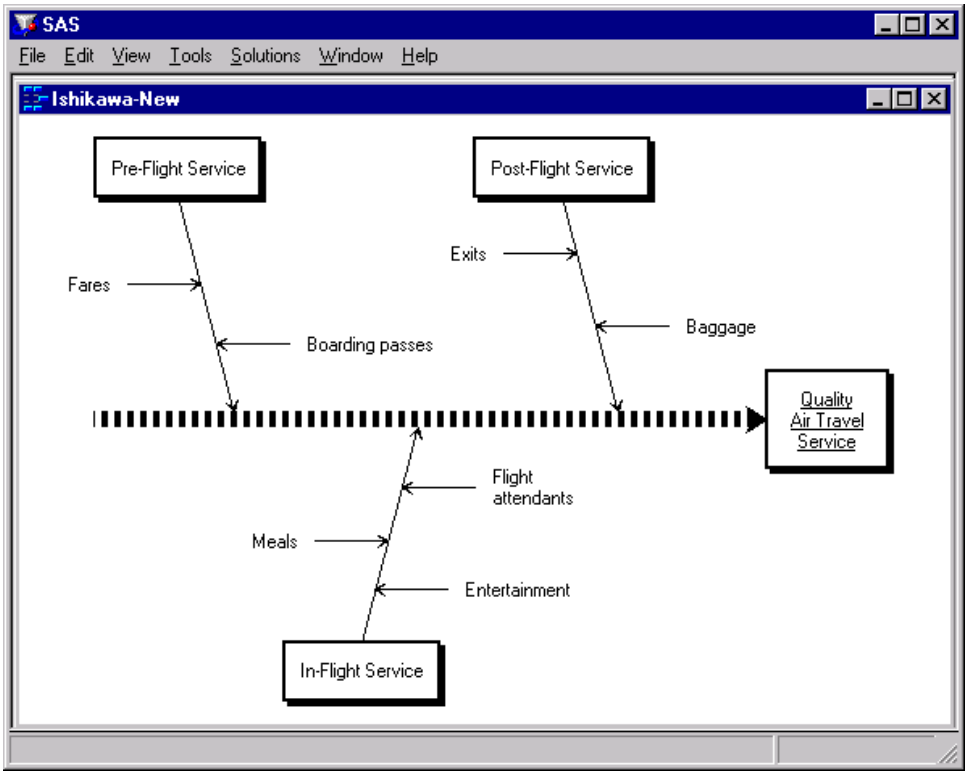
Note that only the trunk is subsetted (as indicated by the underlined label).

Figure 9.65 Subsetting Only the Trunk *(continued)*



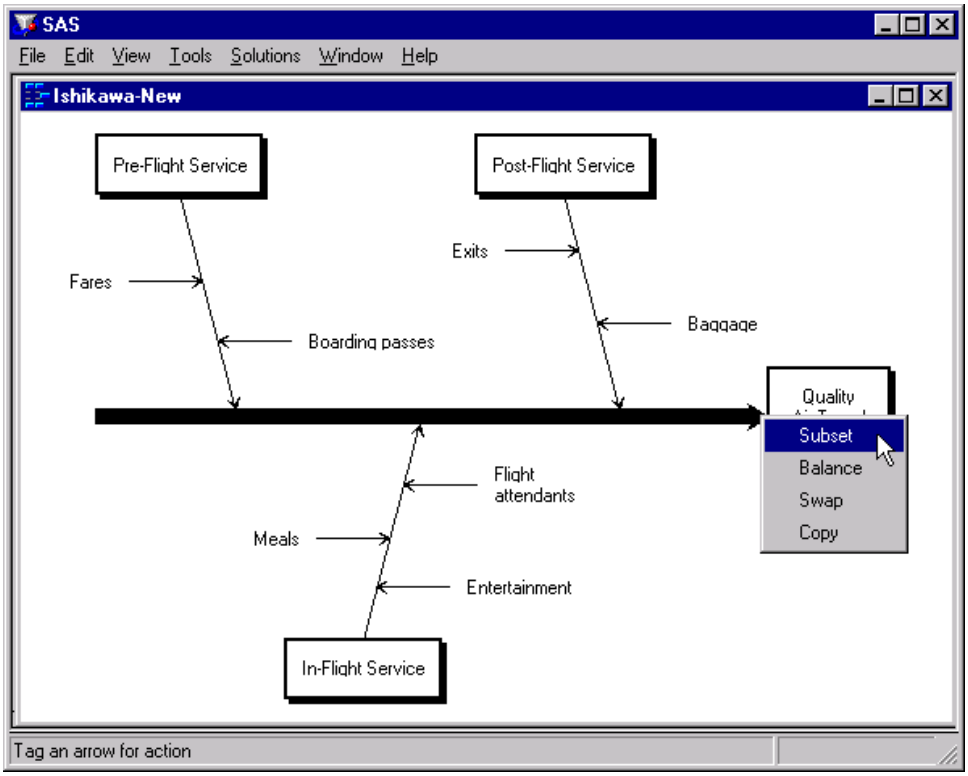
When you select a line style from the line palette, only the line style of the subsetted arrow is changed.

Figure 9.66 Modified Diagram



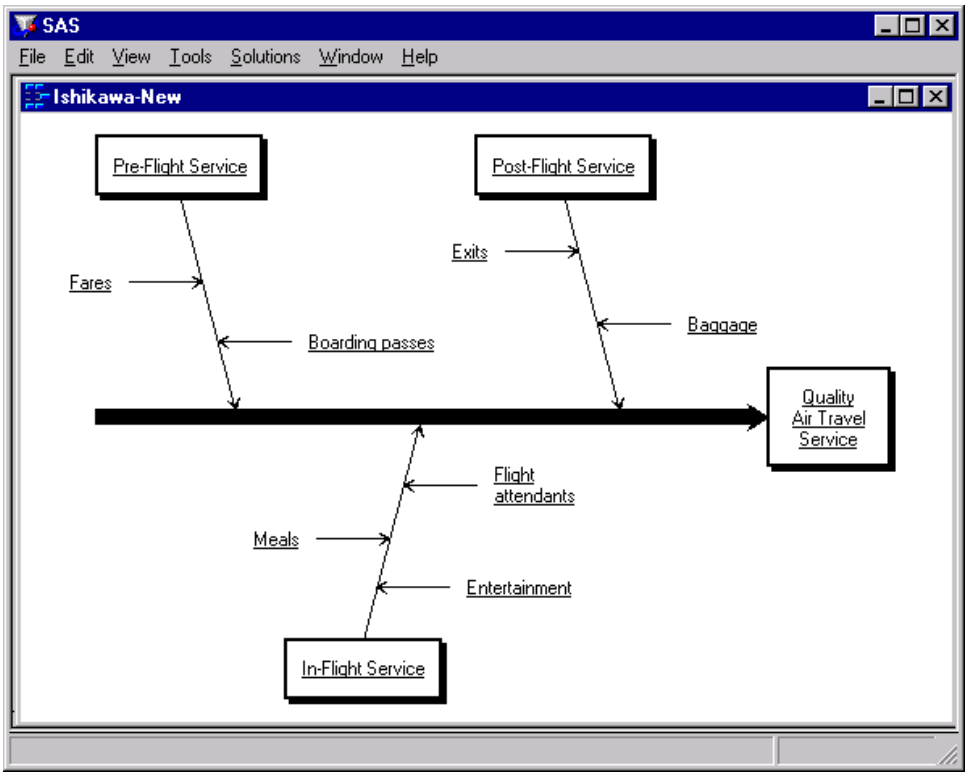
Alternately, if you subset the trunk using the arrow head popup menu, all of the arrows in the diagram are subsetted.

Figure 9.67 Subsetting the Entire Diagram



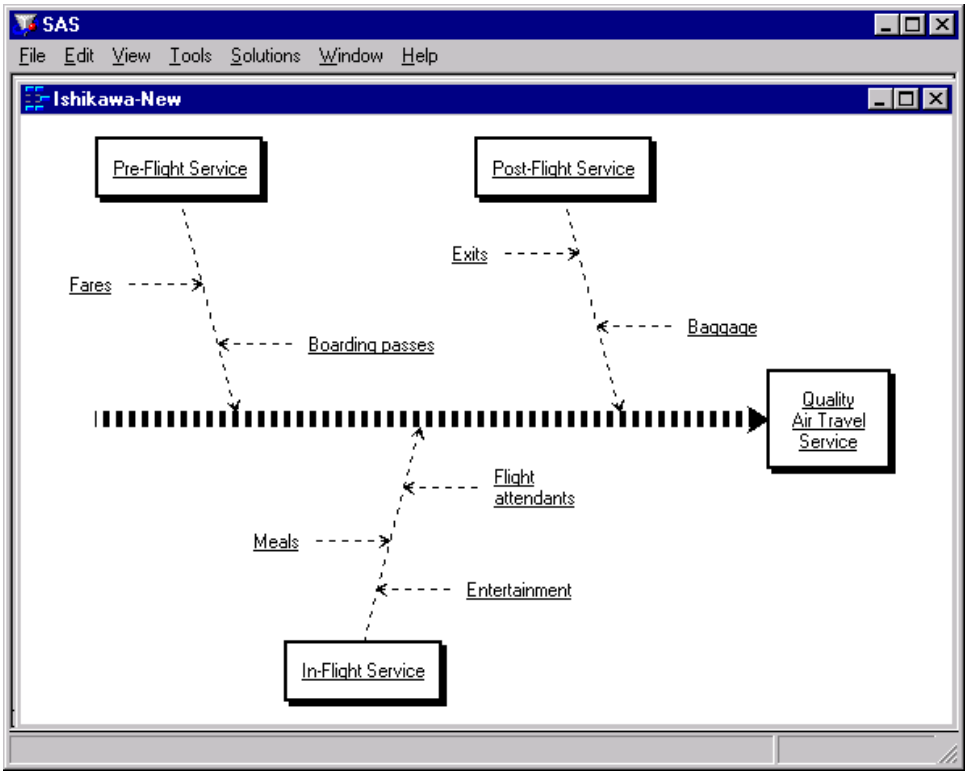
Note that all of the labels in the diagram are now underlined.

Figure 9.68 Subsetting the Entire Diagram *(continued)*



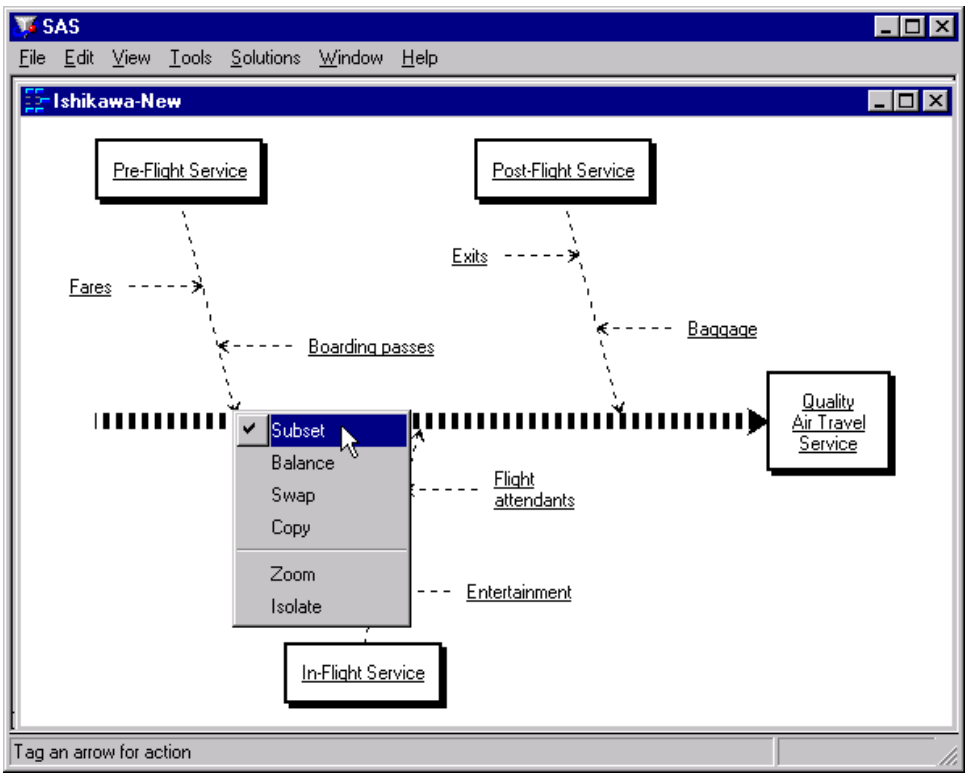
Now, when you select a new line style from the line palette, all the arrows are drawn with this line style.

Figure 9.69 Modified Diagram



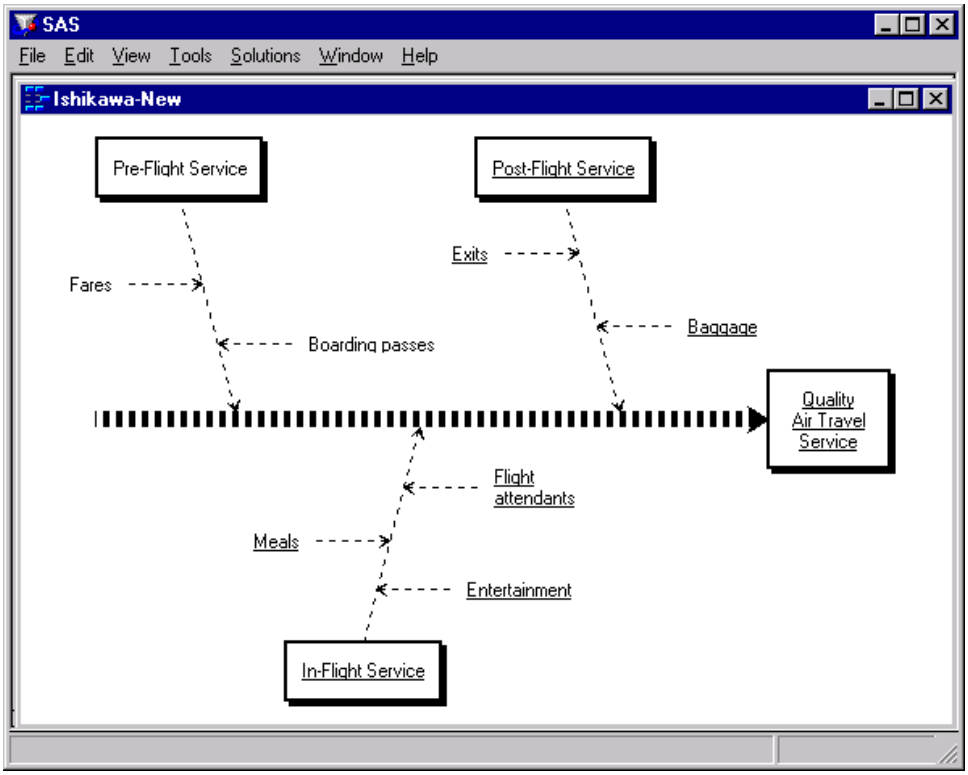
To remove the subset from the *Pre-Flight Service* branch and all its descendants, select **Subset** from the arrow head popup menu.

Figure 9.70 Selectively Removing Tags



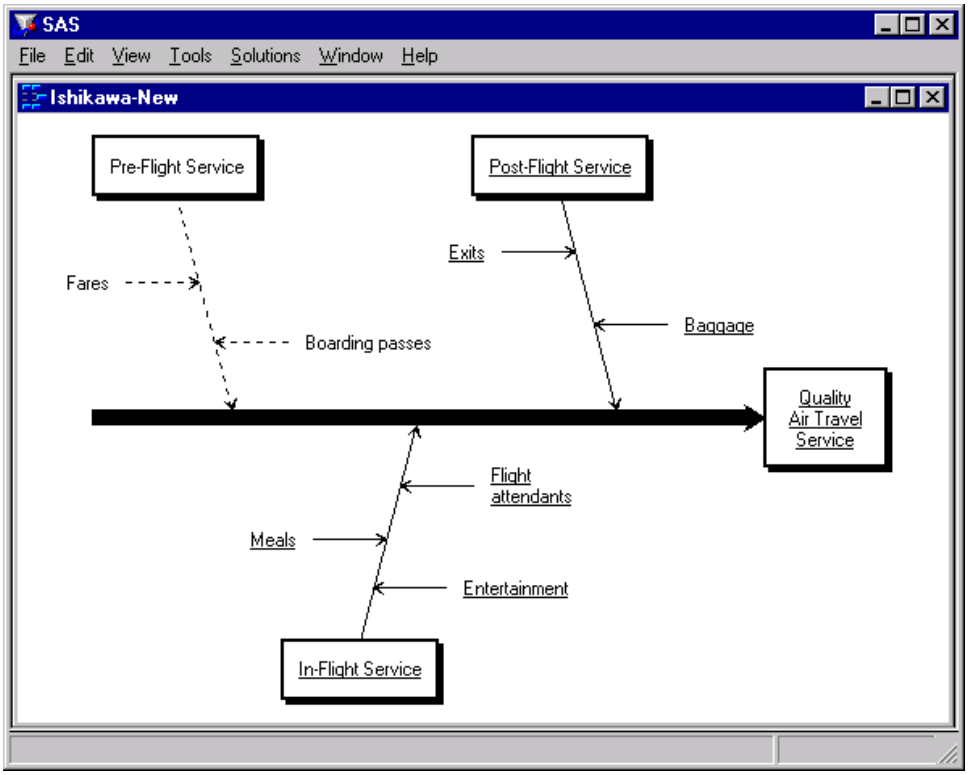
This removes the underlines from the labels in these arrows.

Figure 9.71 Selectively Removing Subsets *(continued)*



You can now use the line palette to change the line style for all the arrows in the diagram with the exception of the *Pre-Flight Service* branch and its descendants:

Figure 9.72 Modified Diagram

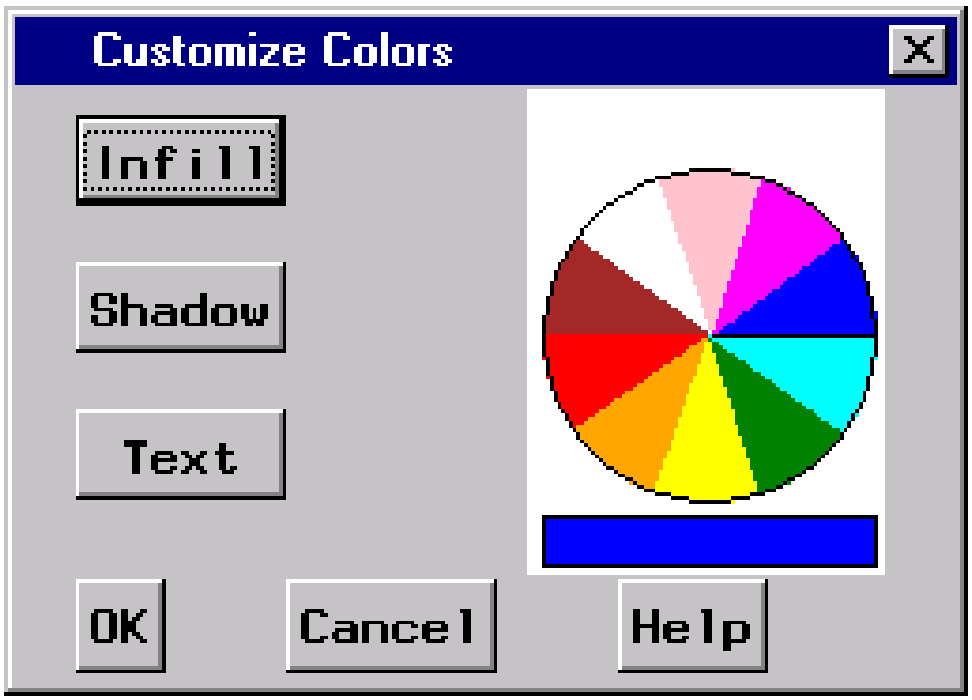


The same principles apply when making color changes—simply use the color palette instead of the line style palette.

Modifying Text Colors

By default, labels have the same color as the arrow. To modify the text color independently of the arrow color, select **View ► Ishhikawa Settings ► Colors...**. The Customize Color window, similar to the following, will open:

Figure 9.73 Colors Dialog



To change the text color of all the arrows[8] in the Ishikawa diagram, do the following:

- Select a color from the color palette.
- Select **Text**.

Once modified, the text color is unaffected by changes to the arrow color. To relink the text color to the arrow color, do the following:

- Select the current text color from the color palette (if it is not already the current color).
- Select **Text**.

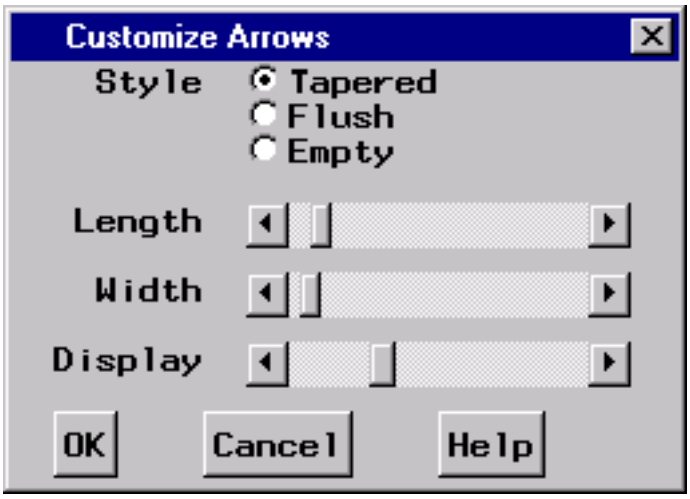
Select **OK**to close the dialog window. To cancel the changes, select **Cancel**.

[8] You cannot directly modify the text color for individual arrows from this dialog.

Modifying Arrow Heads

To modify the characteristics of the arrow heads in your diagram, select **View ► Ishikawa Settings ► Arrows...**. This opens the following dialog:

Figure 9.74 Arrows Dialog



The dialog controls the characteristics of all arrow heads. Arrow heads cannot be modified individually.

Arrow heads can be tapered, flush, or empty. Use the sliders labeled **Length** and **Width** to control the length and width of the arrow heads. Move the sliders to the right to increase the length/width of the arrow head and to the left to decrease the length/width.

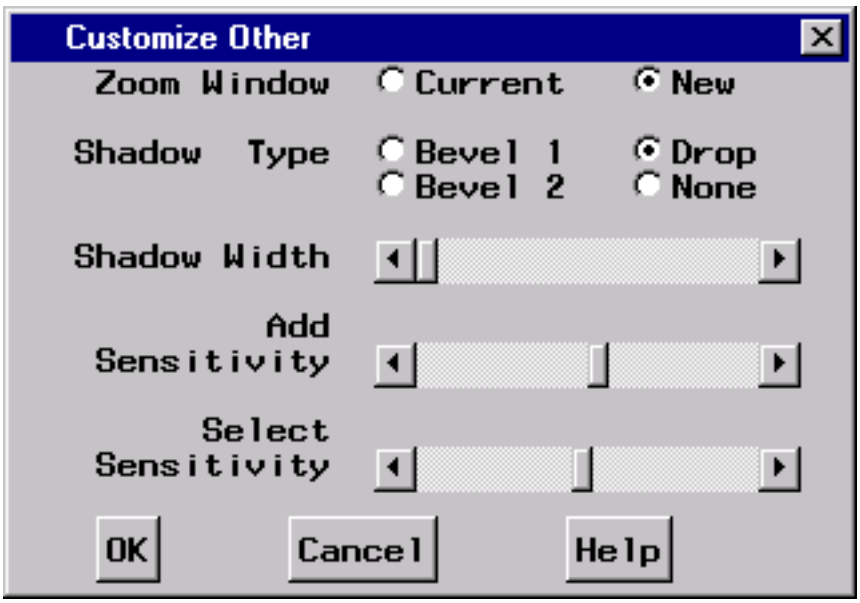
Removing arrow heads increases the readability of a highly detailed diagram. Use the **Display** slider to control the level at which arrow heads are displayed. Move the slider to the extreme left to remove all the arrow heads and to the extreme right to display all the arrow heads. Use the intermediate settings to select a threshold level of detail, above which arrow heads are not displayed. By default, arrow heads are displayed for all levels.

Select **OK**to close the window. To cancel the changes, select **Cancel**.

Modifying Environmental Attributes

You can modify other features of the ISHIKAWA environment such as zooming, mouse sensitivity, and shadow attributes by selecting **View ► Ishikawa Settings ► Other...** to open the following dialog:

Figure 9.75 Others Dialog



Zoom Window controls whether the zoom operation opens a new window or draws in the current window. Select **Current**to reduce the amount of window management required.

The **Shadow Type** button controls the type of shadow that is drawn around the trunk and branch boxes.

- **Bevel 1** draws a beveled edge box with a lower-right light source.
- **Bevel 2** draws a beveled edge box with an upper-left light source.
- **Drop** draws a box with a drop shadow. This is the default.
- **None** suppresses the shadow.

The **Shadow Width** slider controls the shadow width if the boxes have shadows or the outline width when boxes are displayed without shadows. Move the slider to the right to increase the shadow width and to the left to decrease the width.

The **Add Sensitivity** slider controls how closely you must position the cursor to an existing arrow before a mouse click results in an add arrow operation. Move the slider to the right to increase the size of the context-sensitive area and to the left to reduce the size of the context-sensitive area.

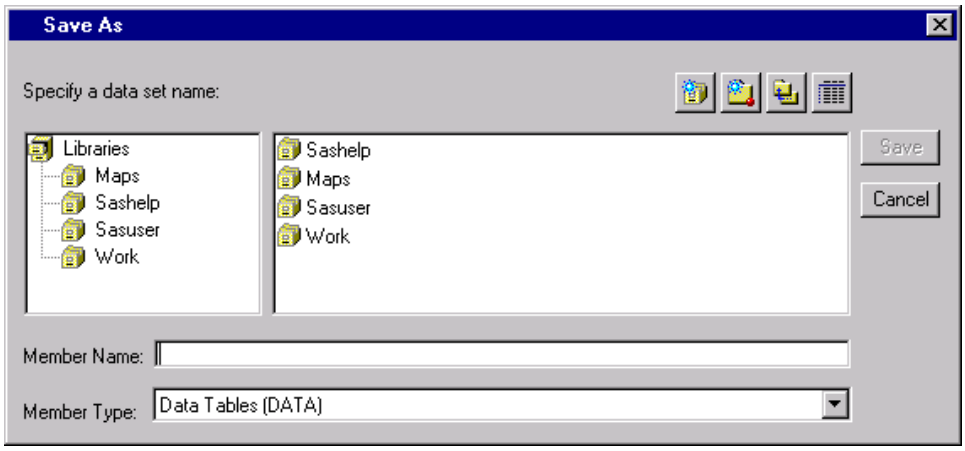
The **Select Sensitivity** slider controls how closely you must position the cursor to an existing arrow before a mouse click results in an edit, delete, move, or popup arrow operation. Operate this slider in the same manner as the **Add Sensitivity** slider.

Saving an Ishikawa Diagram for Future Editing

You must save your Ishikawa diagram as a SAS data set if you intend to edit it in the future with the ISHIKAWA environment. The ISHIKAWA environment does not reconstruct Ishikawa diagrams by reading graphics entries (GRSEG) from SAS catalogs.

Select **File ► Save As ► Data Set**to activate the Data Set Requestor window.

Figure 9.76 Output File Requestor



A list of SAS *librefs* is displayed in the Libraries tree in the left region of the dialog. Begin by selecting a libref from the list. A libref refers to a permanent SAS data library located on your host system. For example, the default SASUSER libref (on most hosts) points to a directory called SASUSER, located under the working directory of your current SAS session. Any data sets saved with the libref **SASUSER** will be saved in that directory.

To direct your SAS data sets to a different directory, select the *Create new library* tool icon to open the New Library dialog. Use this dialog to specify the directory and assign a libref to that directory.

To select the libref **SASUSER**, move your cursor over that entry in the list and click. The region to the right of the Libraries tree is used to display any existing SAS data sets in that library.

To save your diagram in an existing SAS data set, use the mouse to click on an entry in the list. The *member name* field will be updated to reflect your choice. If you want to save your diagram in a new SAS data set, move your cursor to the *member name* field and type the new name (in this example, SERVICE).

Select **Save**to save the diagram and return to the ISHIKAWA environment or select **Cancel**to cancel the save.

Reading an Existing Ishikawa Diagram

To enter the ISHIKAWA environment and resume editing an existing diagram, you must have previously saved the diagram as a SAS data set. The ISHIKAWA environment *does not* allow you to modify graphs stored in SAS/GRAPH catalogs.

You can specify the name of this data set when you establish the ISHIKAWA environment with the following statements:

```
proc ishikawa data=libref.dataset;
run;
```

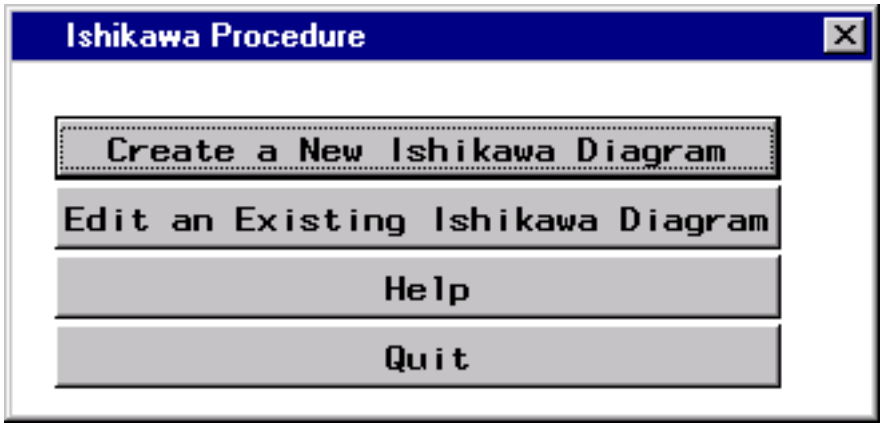
Alternatively, the ISHIKAWA environment will prompt you for a data set after you invoke the environment with the following statements:

```
proc ishikawa;
run;
```

When you specify a data set in the PROC statement, the ISHIKAWA environment is initialized and your diagram is displayed up to the branch level. The message area will indicate if any additional detail is hidden. You can edit your diagram even if some of the diagram is hidden. To add or remove detail one level at a time, select **> Detail** or **< Detail** from the background popup menu.

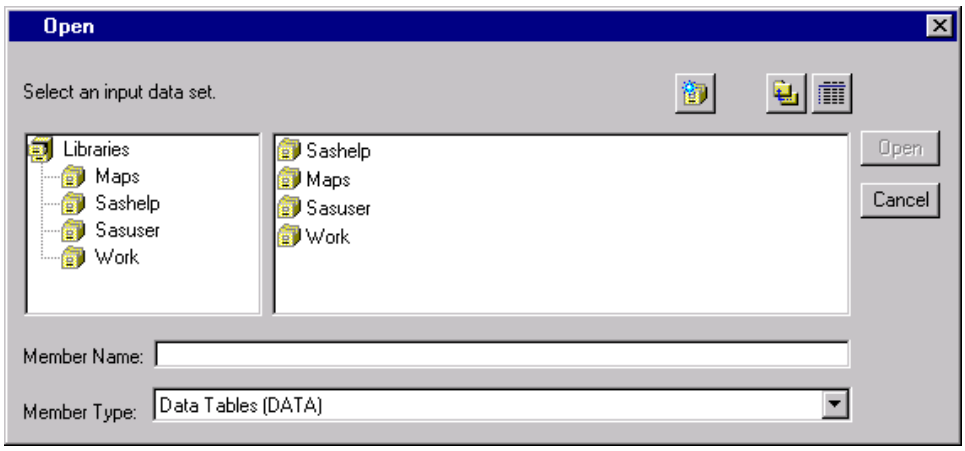
When you do not specify a data set in the PROC statement, you will see the following menu:

Figure 9.77 Initial Menu



Since you are editing an existing diagram rather than starting a new diagram, select **Edit an Existing Diagram** to activate the Member Selector window.

Figure 9.78 Input Member Selector



Use the Member Selector window to specify an input SAS data set. For information on how to specify the SAS data set name, follow the steps outlined in "Saving an Ishikawa Diagram for Future Editing" on page 776.

To establish the ISHIKAWA environment and display the diagram you have selected, select **Open**. The diagram is displayed up to the branch level.

To quit or start a new diagram, return to the main menu by selecting **Cancel**.

Displaying Multiple Ishikawa Diagrams

The ISHIKAWA environment enables you to view multiple Ishikawa diagrams simultaneously for side-by-side comparisons of different diagrams. You can also use this feature to transfer information between diagrams, since the move and copy operations function across windows. You can have up to four ISHIKAWA windows open at one time.

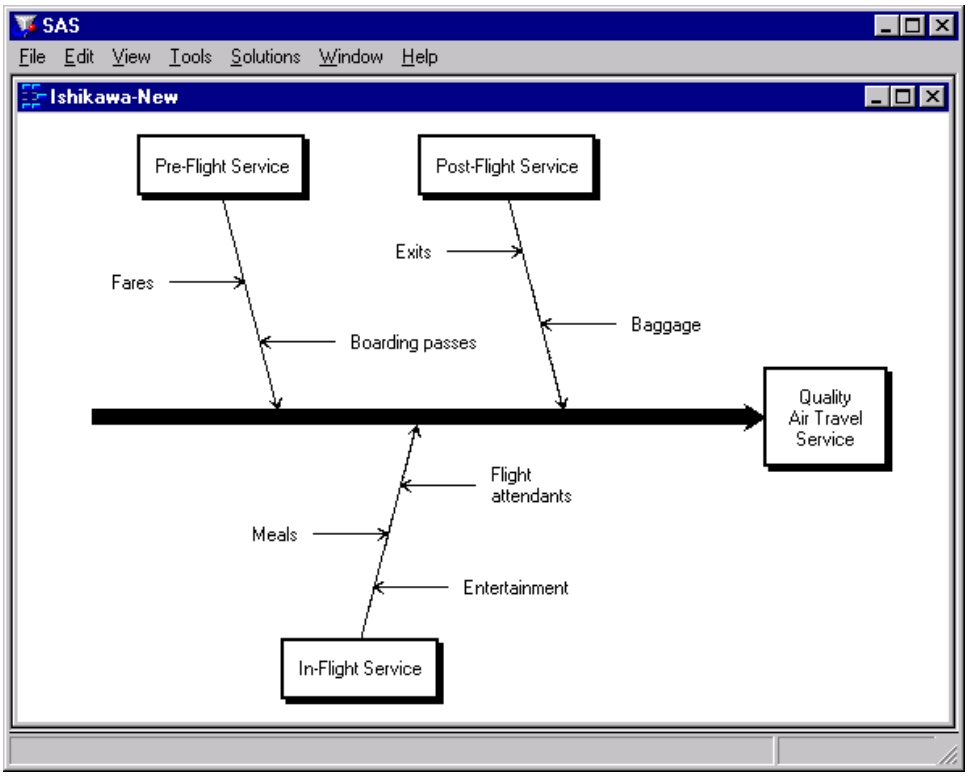
To open a window for another Ishikawa diagram, select **File ► Open**. This will display the Member Selector window, which you can use to specify the name of the input SAS data set for the other Ishikawa diagram.

You can also start new diagrams while displaying other Ishikawa diagrams. To open a window for a new Ishikawa diagram, select **File ► New**. This opens an ISHIKAWA window with a template for a new diagram.

Example

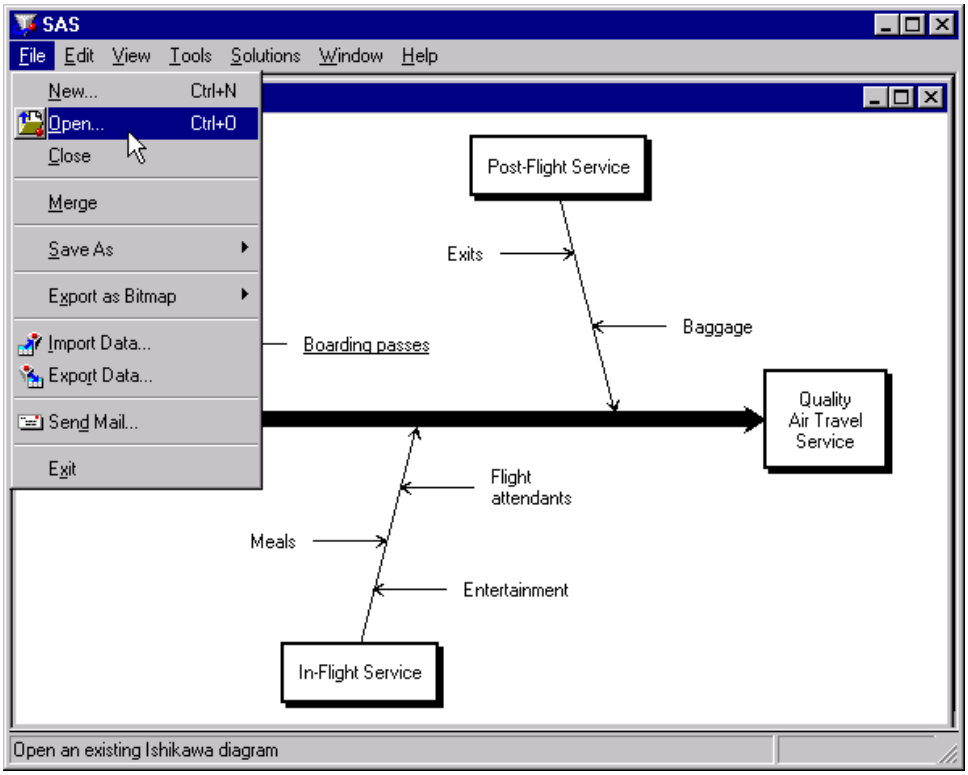
The following figure shows an Ishikawa diagram for *Quality Air Travel Service* after an initial brainstorming session:

Figure 9.79 Single Ishikawa Diagram



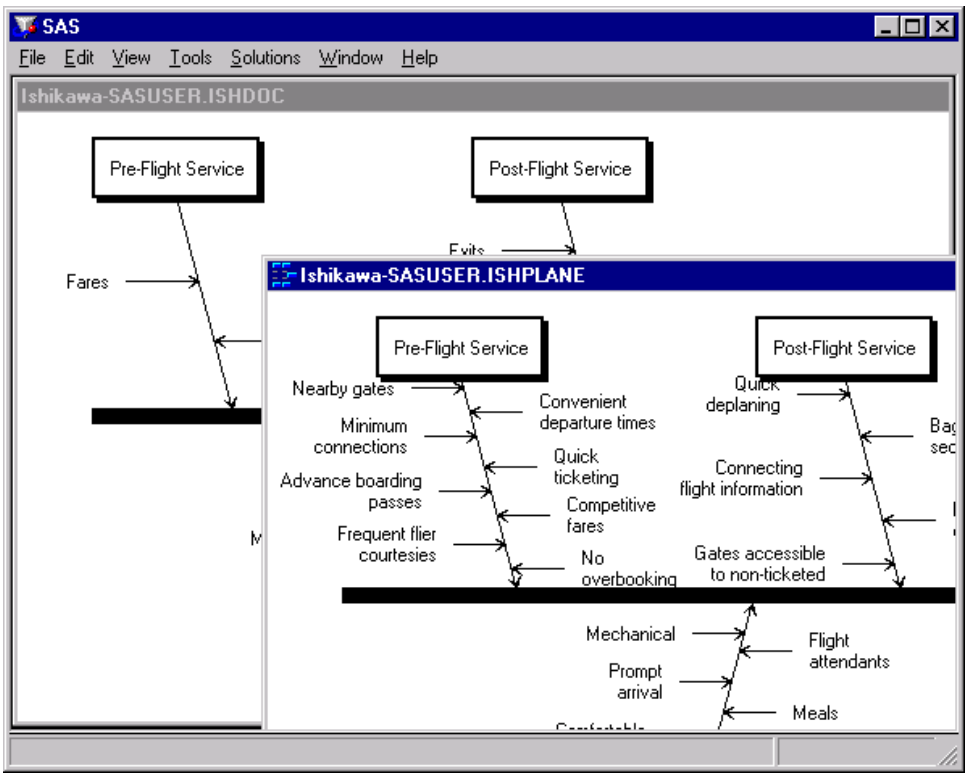
The current diagram and another Ishikawa diagram can be viewed simultaneously by selecting **File ► Open** from the command bar.

Figure 9.80 Opening a Second Diagram



In this situation, displaying both diagrams concurrently emphasizes the improved understanding of the process. It also enables you to transfer information from one diagram to another.

Figure 9.81 Viewing Multiple Ishikawa Diagrams



Input and Output Data Sets

The following is a complete list of the variables in output SAS data sets created by the ISHIKAWA environment:

Variable	Type	Len	Description
LEVEL	Num	8	Level of detail
TEXT1	Char	40	First line of label
TEXT2	Char	40	Second line of label
TEXT3	Char	40	Third line of label
TEXT4	Char	40	Fourth line of label
TEXT5	Char	40	Fifth line of label
NOTE1	Char	40	First line of note
NOTE2	Char	40	Second line of note
NOTE3	Char	40	Third line of note
NOTE4	Char	40	Fourth line of note
RELPOS	Num	8	Relative arrow position
SIDE	Char	1	Side arrow attaches to parent
ANGLE	Num	8	Angle (non-horizontal arrows)
LWIDTH	Num	8	Line width
LSTYLE	Num	8	Line style
LCOLOR	Char	8	Line color
TCOLOR	Char	8	Text color
ICOLOR	Char	8	Box infill color
SCOLOR	Char	8	Shadow color
STYPE	Char	1	Shadow type
SWIDTH	Num	8	Shadow width
RELLNG	Num	8	Relative length of an arrow
HLEVEL	Num	8	Arrow head threshold
HSTYLE	Num	8	Arrow head style
HLNGTH	Num	8	Arrow head length
HWIDTH	Num	8	Arrow head width
HTEXT	Num	8	Font height
FTEXT	Char	8	Font

Only the variables _LEVEL_ and _TEXT1_ are required in the input data set for the ISHIKAWA procedure. Each observation in the input data set corresponds to a particular arrow in the diagram. The order of the observations is critical because it defines the relationships of the arrows.

- The trunk is always the first observation.
- The remaining observations are ordered so that leaves are nested within stems, stems are nested within branches, and branches are nested within the trunk.
- The variable _LEVEL_ is numeric and indicates the level within the diagram. The trunk has a level of 0, branches have a level of 1, stems have a level of 2, and so on.
- The first line of text in a label is stored as _TEXT1_, the second as _TEXT2_, and so on.

Example

The following is a partial listing of the SAS data set used to create the Ishikawa diagram shown in Figure 9.15:

```
data ishplane;
   length _text1_ _text2_ _text3_ $ 40 _side_ $ 1;
   input  _level_ _text1_ & _text2_ & _text3_ & _relpos_ _side_;
   datalines;
0  Quality              Air Travel          Service  -1.00  .
1  Pre-Flight Service   .                   .         0.26  T
2  Competitive          fares               .         0.68  R
2  Convenient           departure times     .         0.18  R
2  Quick                ticketing           .         0.43  R
2  Frequent flier       courtesies          .         0.81  L
1  In-Flight Service    .                   .         0.61  B
2  Prompt               departures          .         0.21  R
2  Comfortable          seating             .         0.35  L
;
```

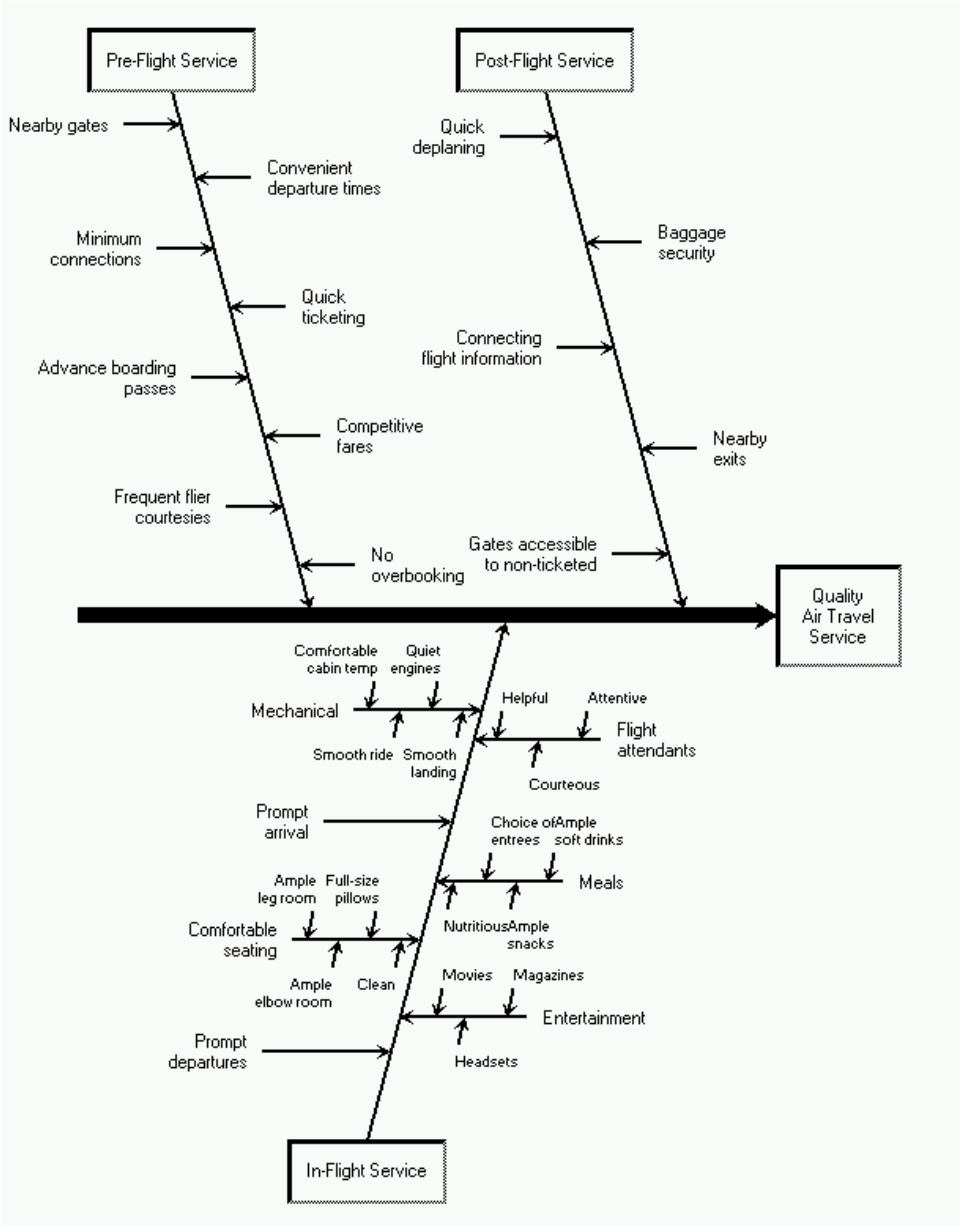
Note the structure of this data set:

- The trunk (always the first observation) has a _LEVEL_ value of zero.
- All subsequent observations for which _LEVEL_ is equal to one are branches that emerge from the trunk.
- Observations 4 and 5 are both leaves that emerge from the preceding stem (observation 3).
- Likewise, leaves 7 and 8 emerge from the preceding stem (observation 6).

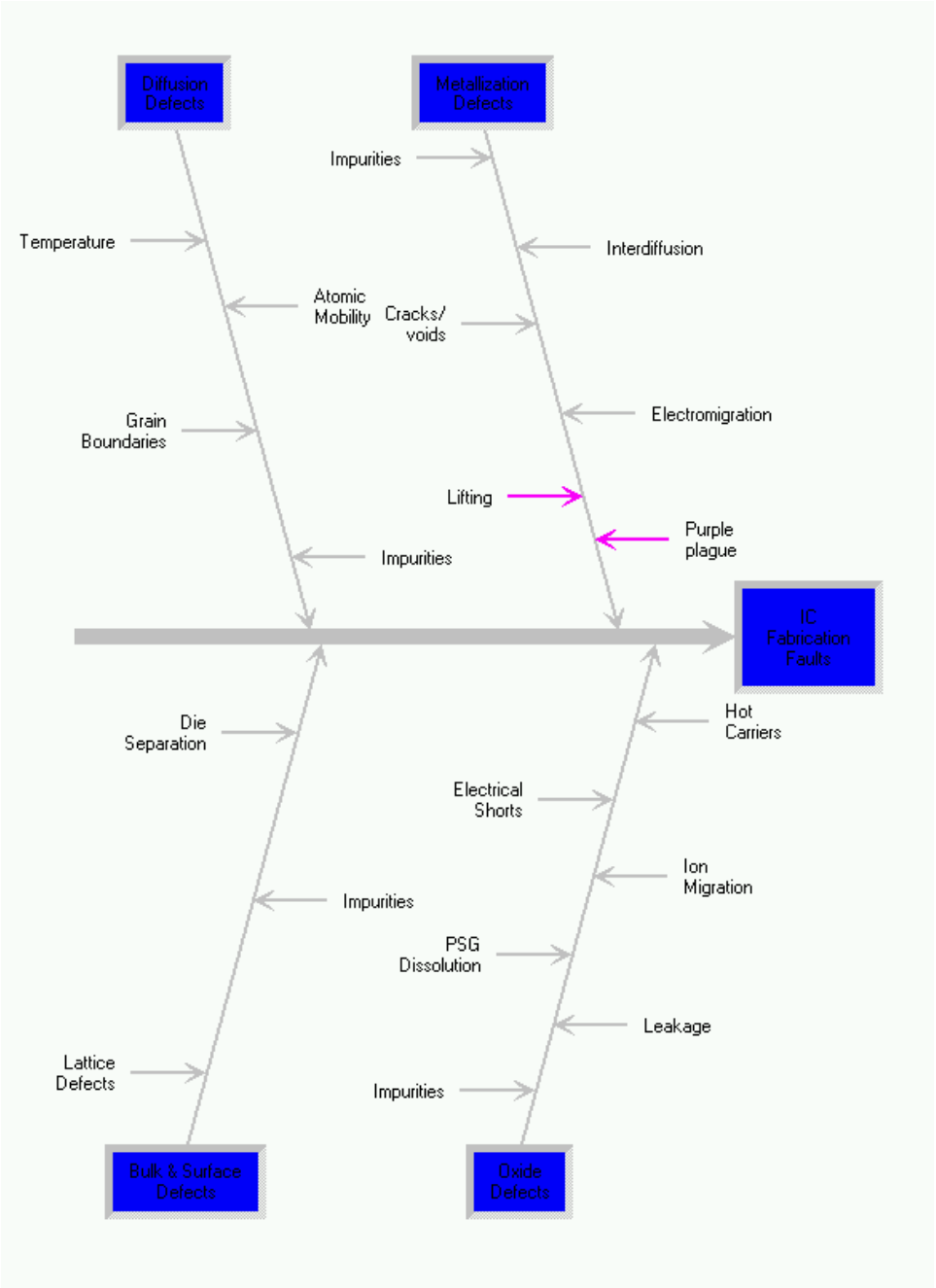
You can use this data set as a way of extracting text and notepad information from the diagram.

Examples: ISHIKAWA Procedure

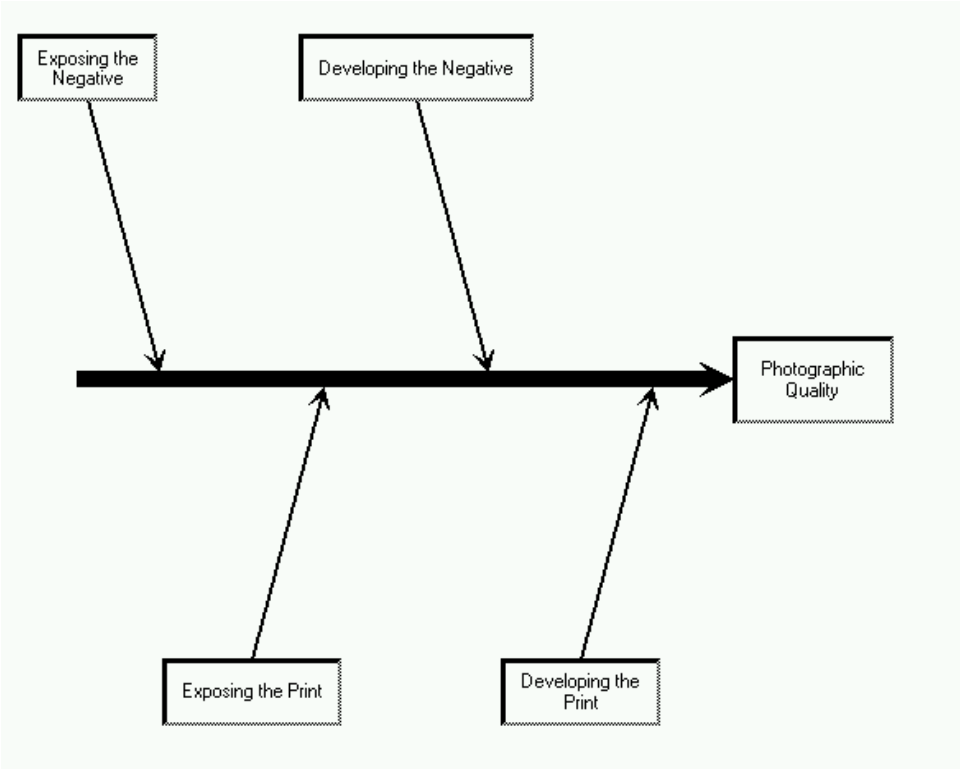
Example 9.1: Quality of Air Travel Service



Example 9.2: Integrated Circuit Failures



Example 9.3: Photographic Development Process



References

Ishikawa, K. (1982). *Guide to Quality Control*. 2nd rev. English ed. Tokyo: Asian Productivity Organization.

Karabatsos, N. A. (1989). "In Memoriam: Dr. Kaoru Ishikawa, Quality Organizer." *Quality Progress* 22:20.

Kume, H. (1985). *Statistical Methods for Quality Improvement*. Tokyo: AOTS Chosakai.

Rodriguez, R. N. (1991). "Applications of Computer Graphics to Two Basic Statistical Quality Improvement Methods." In *Proceedings of the National Computer Graphics Association Conference*, 17–26. Fairfax, VA: NCGA.

Sarazen, J. S. (1990). "The Tools of Quality, Part 2: Cause-and-Effect Diagrams." *Quality Progress* 23:59–62.

Chapter 10
The MACONTROL Procedure

Contents

Introduction: MACONTROL Procedure **788**
Learning about the MACONTROL Procedure 790
PROC MACONTROL Statement **790**
Overview: PROC MACONTROL Statement 790
Syntax: PROC MACONTROL Statement 791
BY Statement 793
Input and Output Data Sets: MACONTROL Procedure 794
EWMACHART Statement: MACONTROL Procedure **795**
Overview: EWMACHART Statement 795
Getting Started: EWMACHART Statement 796
Creating EWMA Charts from Raw Data 796
Creating EWMA Charts from Subgroup Summary Data 799
Saving Summary Statistics 802
Saving Control Limit Parameters 803
Reading Preestablished Control Limit Parameters 805
Syntax: EWMACHART Statement 807
Summary of Options 808
Dictionary of Special Options 817
Details: EWMACHART Statement 820
Constructing EWMA Charts 820
Output Data Sets 825
ODS Tables 828
ODS Graphics 828
Input Data Sets 829
Methods for Estimating the Standard Deviation 832
Axis Labels 834
Missing Values 834
Examples: EWMACHART Statement 835
Example 10.1: Specifying Standard Values for the Process Mean and Process Standard Deviation 835
Example 10.2: Displaying Limits Based on Asymptotic Values 837
Example 10.3: Working with Unequal Subgroup Sample Sizes 838
Example 10.4: Displaying Individual Measurements on an EWMA Chart 844
Example 10.5: Computing Average Run Lengths 846
MACHART Statement: MACONTROL Procedure **848**
Overview: MACHART Statement 848

Getting Started: MACHART Statement . . . 849
Creating Moving Average Charts from Raw Data . . . 849
Creating Moving Average Charts from Subgroup Summary Data . . . 853
Saving Summary Statistics . . . 855
Saving Control Limit Parameters . . . 856
Reading Preestablished Control Limit Parameters . . . 859
Syntax: MACHART Statement . . . 861
Summary of Options . . . 862
Dictionary of Special Options . . . 871
Details: MACHART Statement . . . 874
Constructing Uniformly Weighted Moving Average Charts . . . 874
Output Data Sets . . . 879
ODS Tables . . . 882
ODS Graphics . . . 883
Input Data Sets . . . 883
Methods for Estimating the Standard Deviation . . . 886
Axis Labels . . . 888
Missing Values . . . 888
Examples: MACHART Statement . . . 889
Example 10.6: Specifying Standard Values for the Process Mean and Process Standard Deviation . . . 889
Example 10.7: Annotating Average Run Lengths on the Chart . . . 891
INSET Statement: MACONTROL Procedure . . . **892**
Overview: INSET Statement . . . 892
Getting Started: INSET Statement . . . 893
Syntax: INSET Statement . . . 894
References . . . **896**

Introduction: MACONTROL Procedure

The MACONTROL procedure creates moving average control charts, which are tools for deciding whether a process is in a state of statistical control and for detecting shifts in a process average. The procedure creates the following two types of charts:

- *uniformly weighted moving average charts* (commonly referred to as *moving average charts*). Each point on a moving average chart represents the average of the w most recent subgroup means, including the present subgroup mean. The next moving average is computed by dropping the oldest of the previous w subgroup means and including the newest subgroup mean.

  The constant w, often referred to as the *span* of the moving average, is a parameter of the moving average chart. There is an inverse relationship between w and the magnitude of the shift to be detected; larger values of w are used to guard against smaller shifts.

- *exponentially weighted moving average (EWMA) charts*, also referred to as *geometric moving average (GMA) charts*. Each point on an EWMA chart represents the weighted average of all the previous subgroup means, including the mean of the present subgroup sample. The weights decrease exponentially going backward in time.

  The weight r ($0 < r \leq 1$) assigned to the present subgroup sample mean is a parameter of the EWMA chart. Small values of r are used to guard against small shifts. If $r = 1$, the EWMA chart reduces to a Shewhart $\overline{X}$ chart.

In the MACONTROL procedure, the EWMACHART statement produces EWMA charts, and the MACHART statement produces uniformly weighted moving average charts.

In contrast to the Shewhart chart where each point is based on information from a single subgroup sample, each point on a moving average chart combines information from the current sample and past samples. Consequently, the moving average chart is more sensitive to small shifts in the process average. On the other hand, it is more difficult to interpret patterns of points on a moving average chart, since consecutive moving averages can be highly correlated, as pointed out by Nelson (1983).

You can use the MACONTROL procedure to

- read raw data (actual measurements) or summarized data (subgroup means and standard deviations) to create charts
- specify control limits as probability limits or in terms of a multiple of the standard error of the moving average
- adjust the control limits to compensate for unequal subgroup sample sizes
- accept numeric- or character-valued subgroup variables
- display subgroups with date and time formats
- estimate the process standard deviation σ using a variety of methods or specify a standard (known) value for σ
- analyze multiple process variables in the same chart statement
- provide multiple chart statements. If used with a BY statement, the procedure generates charts separately for BY groups of observations.
- tabulate the information displayed in the control chart
- save moving averages, control limits, and control limit parameters in output data sets
- superimpose plotted points with stars (polygons) whose vertices indicate the values of multivariate data related to the process
- display a trend chart below the moving average chart that plots a systematic or fitted trend in the data
- produce charts as traditional graphics, ODS Graphics output, or legacy line printer charts. Line printer charts can use special formatting characters that improve the appearance of the chart. Traditional graphics can be annotated, saved, and replayed.

Learning about the MACONTROL Procedure

If you are using the MACONTROL procedure for the first time, begin by reading "PROC MACONTROL Statement" on page 790 to learn about input data sets. Then read the section "Getting Started: EWMACHART Statement" on page 796 in "EWMACHART Statement: MACONTROL Procedure" on page 795 or the section "Getting Started: MACHART Statement" on page 849 in "MACHART Statement: MACONTROL Procedure" on page 848. These chapters also provide syntax information, computational details, and advanced examples.

PROC MACONTROL Statement

Overview: PROC MACONTROL Statement

The PROC MACONTROL statement starts the MACONTROL procedure and it identifies input data sets.

After the PROC MACONTROL statement, you provide either an EWMACHART or an MACHART statement that specifies the type of moving average chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request a uniformly weighted moving average chart:

```
proc macontrol data=values;
   machart weight*lot / mu0    = 8.10
                        sigma0 = 0.05
                        span   = 5;
run;
```

In this example, the DATA= option specifies an input data set named values that contains the *process* measurement variable weight and the *subgroup-variable* lot.

You can use options in the PROC MACONTROL statement to do the following:

- specify input data sets containing variables to be analyzed, parameters for calculating moving averages, or annotation information
- specify a graphics catalog for saving traditional graphics
- specify that charts be produced as traditional graphics or line printer charts
- define characters used for features on line printer charts

In addition to the chart statement, you can provide BY statements, ID statements, TITLE statements, and FOOTNOTE statements. If you are producing traditional graphics, you can also provide graphics enhancement statements, such as SYMBOL*n* statements, which are described in *SAS/GRAPH: Reference*.

See Chapter 4, "SAS/QC Graphics," for a detailed discussion of the alternatives available for producing charts with SAS/QC procedures.

NOTE: If you are using the MACONTROL procedure for the first time, you should also read the sections "Getting Started: EWMACHART Statement" on page 796 and "Getting Started: MACHART Statement" on page 849.

Syntax: PROC MACONTROL Statement

The syntax for the PROC MACONTROL statement is as follows:

PROC MACONTROL < *options* > **;**

The PROC MACONTROL statement starts the MACONTROL procedure, and it optionally identifies various data sets and requests line printer charts. You can specify the following options in the PROC MACONTROL statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains appropriate annotate variables, as described in *SAS/GRAPH: Reference*. The ANNOTATE= option enables you to add features to the moving average chart (for example, labels that explain out-of-control points). The ANNOTATE= data set is used only when the chart is created as traditional graphics; it is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option.

The data set specified with the ANNOTATE= option in the PROC MACONTROL statement is a "global" annotate data set in the sense that the information in this data set is displayed on every chart produced in the current run of the MACONTROL procedure.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an input data set that contains appropriate annotate variables that add features to the trend chart (secondary chart) produced with the TRENDVAR= option in the EWMACHART or MACHART statement. This option is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option.

DATA=*SAS-data-set*

names an input data set that contains raw data (measurements) as observations. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). The DATA= data set can contain more than one observation for each value of the *subgroup-variable*.

You cannot specify a DATA= data set with a HISTORY= or TABLE= data set. If you do not specify an input data set, PROC MACONTROL uses the most recently created data set as a DATA= data set. For more information, see "DATA= Data Set" in the appropriate chart statement chapter.

FORMCHAR(*index***)='***string***'**

defines characters used for features on line printer charts, where *index* is a list of numbers ranging from 1 to 17 and *string* is a character or hexadecimal string. This option applies only if you also specify the LINEPRINTER option.

The *index* identifies which features are controlled with the *string* characters, as described in Table 10.1. If you specify the FORMCHAR= option and omit the *index*, the *string* controls all 17 features.

Table 10.1 FORMCHAR= Features

Value of *index*	Description of Character	Chart Feature
1	Vertical bar	Frame
2	Horizontal bar	Frame, central line
3	Box character (upper left)	Frame
4	Box character (upper middle)	Serifs, tick (horizontal axis)
5	Box character (upper right)	Frame
6	Box character (middle left)	Not used
7	Box character (middle middle)	Serifs
8	Box character (middle right)	Tick (vertical axis)
9	Box character (lower left)	Frame
10	Box character (lower middle)	Serifs
11	Box character (lower right)	Frame
12	Vertical bar	Control limits
13	Horizontal bar	Control limits
14	Box character (upper right)	Control limits
15	Box character (lower left)	Control limits
16	Box character (lower right)	Control limits
17	Box character (upper left)	Control limits

Not all printers can produce the characters in the preceding list. By default, the form character list specified by the SAS system FORMCHAR= option is used; otherwise, the default is FORMCHAR='|—-|+|—|=====’. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that you can use the FORMCHAR= option to temporarily override the values of the SAS system FORMCHAR= option. The values of the SAS system FORMCHAR= option are not altered by the FORMCHAR= option in the PROC MACONTROL statement.

GOUT=*graphics-catalog*

specifies the graphics catalog for traditional graphics output from PROC MACONTROL. This is useful if you want to save the output. The GOUT= option is used only when the chart is created using traditional graphics; it is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option.

HISTORY=*SAS-data-set*

HIST=*SAS-data-set*

names an input data set that contains subgroup summary statistics (means, standard deviations, and sample sizes). Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC MACONTROL or PROC SHEWHART, but it can also be created with a SAS summarization procedure such as PROC MEANS.

If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). A HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set with a DATA= or TABLE= data set. If you do not specify an input data set, PROC MACONTROL uses the most recently created data set as a DATA= data set. For more information on HISTORY= data sets, see "HISTORY= Data Set" in the appropriate chart statement chapter.

LIMITS=*SAS-data-set*
names an input data set that contains the control limit parameters for the moving average chart. Each observation in a LIMITS= data set contains the parameters for a *process*.

For details about the variables needed in a LIMITS= data set, see "LIMITS= Data Set" in the appropriate chart statement chapter.

If you do not provide a LIMITS= data set, you must specify the parameters with options in the chart statement.

LINEPRINTER
requests that legacy line printer charts be produced.

TABLE=*SAS-data-set*
names an input data set that contains subgroup summary statistics and control limits. Each observation in a TABLE= data set provides information for a particular subgroup and *process*. Typically, this data set is created as an OUTTABLE= data set in a previous run of PROC MACONTROL.

You cannot use a TABLE= data set with a DATA= or HISTORY= data set. If you do not specify an input data set, PROC MACONTROL uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" section in the appropriate chart statement chapter.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MACONTROL to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

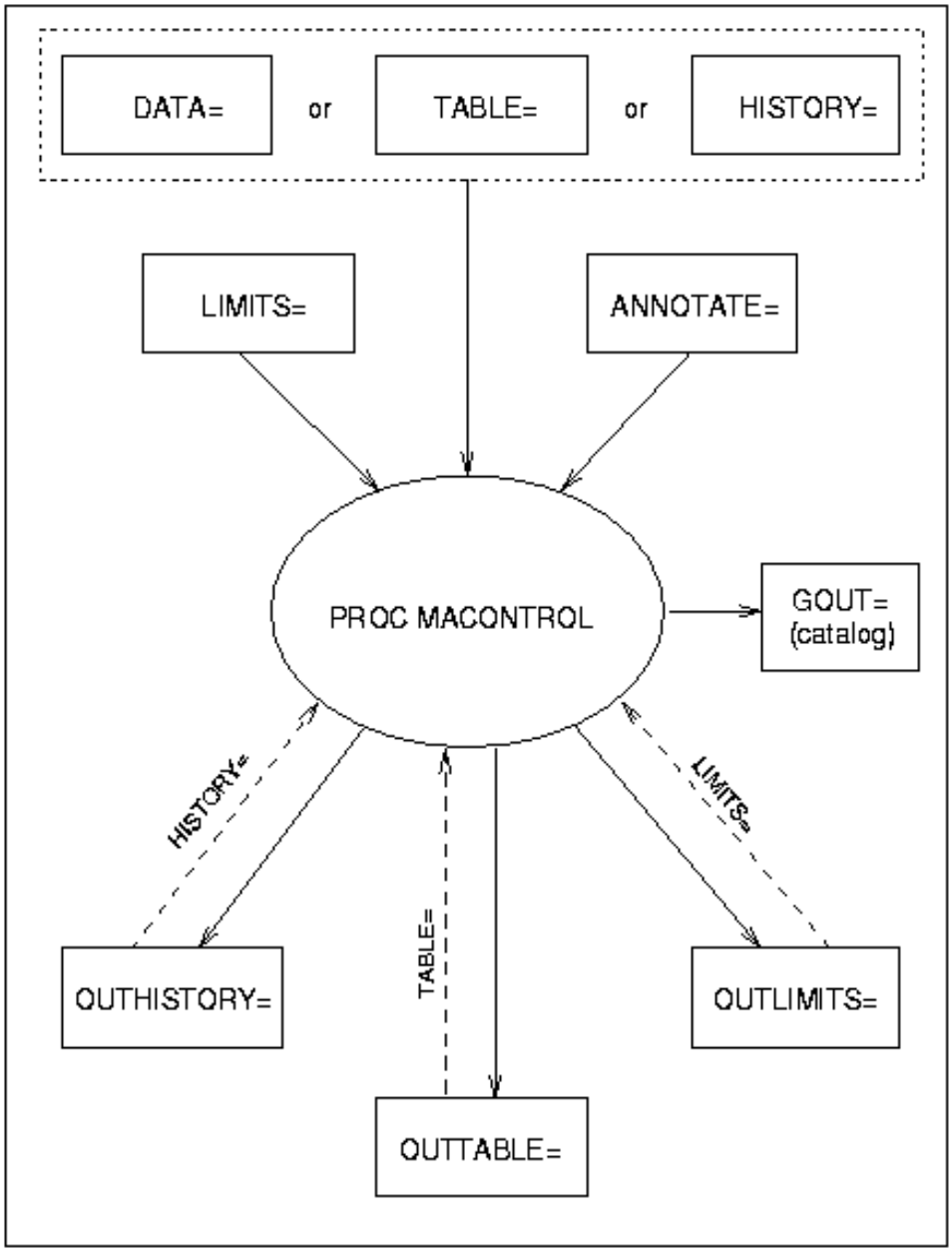
- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the MACONTROL procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

Input and Output Data Sets: MACONTROL Procedure

Figure 10.1 summarizes the data sets used with the MACONTROL procedure.

Figure 10.1 Input and Output Data Sets in the MACONTROL Procedure



EWMACHART Statement: MACONTROL Procedure

Overview: EWMACHART Statement

The EWMACHART statement creates an exponentially weighted moving average (EWMA) control chart, which is used to determine whether a process is in a state of statistical control and to detect shifts in the process average.

You can use options in the EWMACHART statement to

- specify the weight assigned to the most recent subgroup mean in the computation of the EWMAs
- compute control limits from the data based on a multiple of the standard error of the plotted EWMAs or as probability limits
- tabulate the EWMAs, subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limit parameters in an output data set
- save the EWMAs, subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read control limit parameters from an input data set
- specify one of several methods for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display a secondary chart that plots a time trend removed from the data
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing EWMA charts with the EWMACHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: EWMACHART Statement

This section introduces the EWMACHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the EWMACHART statement is presented in the section "Syntax: EWMACHART Statement" on page 807, and advanced examples are given in the section "Examples: EWMACHART Statement" on page 835.

Creating EWMA Charts from Raw Data

NOTE: See *Exponentially Weighted Moving Average Chart* in the SAS/QC Sample Library.

In the manufacture of a metal clip, the gap between the ends of the clip is a critical dimension. To monitor the process for a change in the average gap, subgroup samples of five clips are selected daily. The data are analyzed with an EWMA chart. The gaps recorded during the first twenty days are saved in a SAS data set named Clips1.

```
data Clips1;
   input Day @ ;
   do i=1 to 5;
      input Gap @ ;
      output;
   end;
   drop i;
   datalines;
 1  14.76  14.82  14.88  14.83  15.23
 2  14.95  14.91  15.09  14.99  15.13
 3  14.50  15.05  15.09  14.72  14.97
 4  14.91  14.87  15.46  15.01  14.99
 5  14.73  15.36  14.87  14.91  15.25
 6  15.09  15.19  15.07  15.30  14.98
 7  15.34  15.39  14.82  15.32  15.23
 8  14.80  14.94  15.15  14.69  14.93
 9  14.67  15.08  14.88  15.14  14.78
10  15.27  14.61  15.00  14.84  14.94
11  15.34  14.84  15.32  14.81  15.17
12  14.84  15.00  15.13  14.68  14.91
13  15.40  15.03  15.05  15.03  15.18
14  14.50  14.77  15.22  14.70  14.80
15  14.81  15.01  14.65  15.13  15.12
16  14.82  15.01  14.82  14.83  15.00
17  14.89  14.90  14.60  14.40  14.88
18  14.90  15.29  15.14  15.20  14.70
19  14.77  14.60  14.45  14.78  14.91
20  14.80  14.58  14.69  15.02  14.85
;
```

A partial listing of Clips1 is shown in Figure 10.2.

Figure 10.2 Partial Listing of the Data Set Clips1

The Data Set Clips1

Day	Gap
1	14.76
1	14.82
1	14.88
1	14.83
1	15.23
2	14.95
2	14.91
2	15.09
2	14.99
2	15.13
3	14.50
3	15.05
3	15.09
3	14.72
3	14.97

The data set Clips1 is said to be in "strung-out" form, since each observation contains the day and gap measurement of a single clip. The first five observations contain the gap measurements for the first day, the second five observations contain the gap measurements for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Gap contains the gap measurements and is referred to as the *process variable* (or *process* for short).

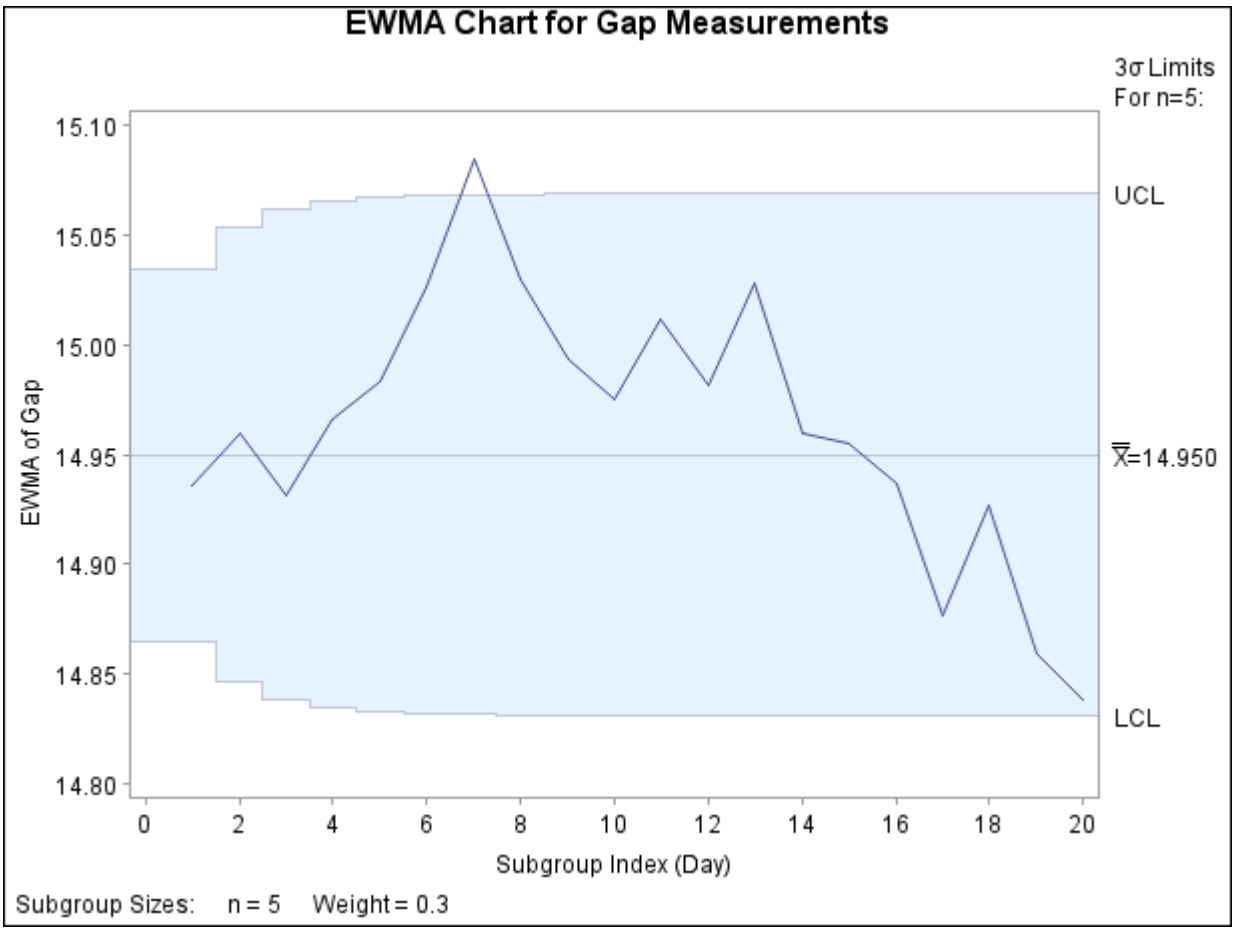
The within-subgroup variability of the gap measurements is known to be stable. You can use an EWMA chart to determine whether the mean level is in control. The following statements create the EWMA chart shown in Figure 10.3:

```
ods graphics off;
symbol h = 0.8;
title 'EWMA Chart for Gap Measurements';
proc macontrol data=Clips1;
   ewmachart Gap*Day / weight=0.3;
run;
```

This example illustrates the basic form of the EWMACHART statement. After the keyword EWMACHART, you specify the *process* to analyze (in this case, Gap) followed by an asterisk and the *subgroup-variable* (Day). The WEIGHT= option specifies the weight parameter used to compute the EWMAs. Options such as WEIGHT= are specified after the slash (/) in the EWMACHART statement. A complete list of options is presented in the section "Syntax: EWMACHART Statement" on page 807. You must provide the weight parameter to create an EWMA chart. As an alternative to specifying the WEIGHT= option, you can read the weight parameter from an input data set; see "Reading Preestablished Control Limit Parameters" on page 805.

The input data set is specified with the DATA= option in the PROC MACONTROL statement.

Figure 10.3 Exponentially Weighted Moving Average Chart



Each point on the chart represents the EWMA for a particular day. The EWMA E_1 plotted at Day=1 is the weighted average of the overall mean and the subgroup mean for Day=1. The EWMA E_2 plotted at Day=2 is the weighted average of the EWMA E_1 and the subgroup mean for Day=2.

$$
\begin{aligned}
E_1 &= 0.3(14.904) + 0.7(14.952) = 14.9376\text{mm} \\
E_2 &= 0.3(15.014) + 0.7(14.9376) = 14.9605\text{mm}
\end{aligned}
$$

For succeeding days, the EWMA is the weighted average of the previous EWMA and the present subgroup mean. In the example, a weight parameter of 0.3 is used (since WEIGHT=0.3 is specified in the EWMACHART statement).

Note that the EWMA for the 7th day lies above the upper control limit, signaling an out-of-control process.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 10.5.

For computational details, see "Constructing EWMA Charts" on page 820. For more details on reading from a DATA= data set, see "DATA= Data Set" on page 829.

Creating EWMA Charts from Subgroup Summary Data

NOTE: See *Exponentially Weighted Moving Average Chart* in the SAS/QC Sample Library.

The previous example illustrates how you can create EWMA charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the EWMACHART statement with data of this type.

The following data set (Clipsum) provides the data from the preceding example in summarized form:

```
data Clipsum;
   input Day GapX GapS;
   GapN=5;
   datalines;
 1  14.904  0.18716
 2  15.014  0.09317
 3  14.866  0.25006
 4  15.048  0.23732
 5  15.024  0.26792
 6  15.126  0.12260
 7  15.220  0.23098
 8  14.902  0.17254
 9  14.910  0.19824
10  14.932  0.24035
11  15.096  0.25618
12  14.912  0.16903
13  15.138  0.15928
14  14.798  0.26329
15  14.944  0.20876
16  14.896  0.09965
17  14.734  0.22512
18  15.046  0.24141
19  14.702  0.17880
20  14.788  0.16634
;
```

A partial listing of Clipsum is shown in Figure 10.4. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day). The variable GapX contains the subgroup means, the variable GapS contains the subgroup standard deviations, and the variable GapN contains the subgroup sample sizes (these are all five).

Figure 10.4 The Summary Data Set Clipsum

The Data Set Clipsum

Day	GapX	GapS	GapN
1	14.904	0.18716	5
2	15.014	0.09317	5
3	14.866	0.25006	5
4	15.048	0.23732	5
5	15.024	0.26792	5

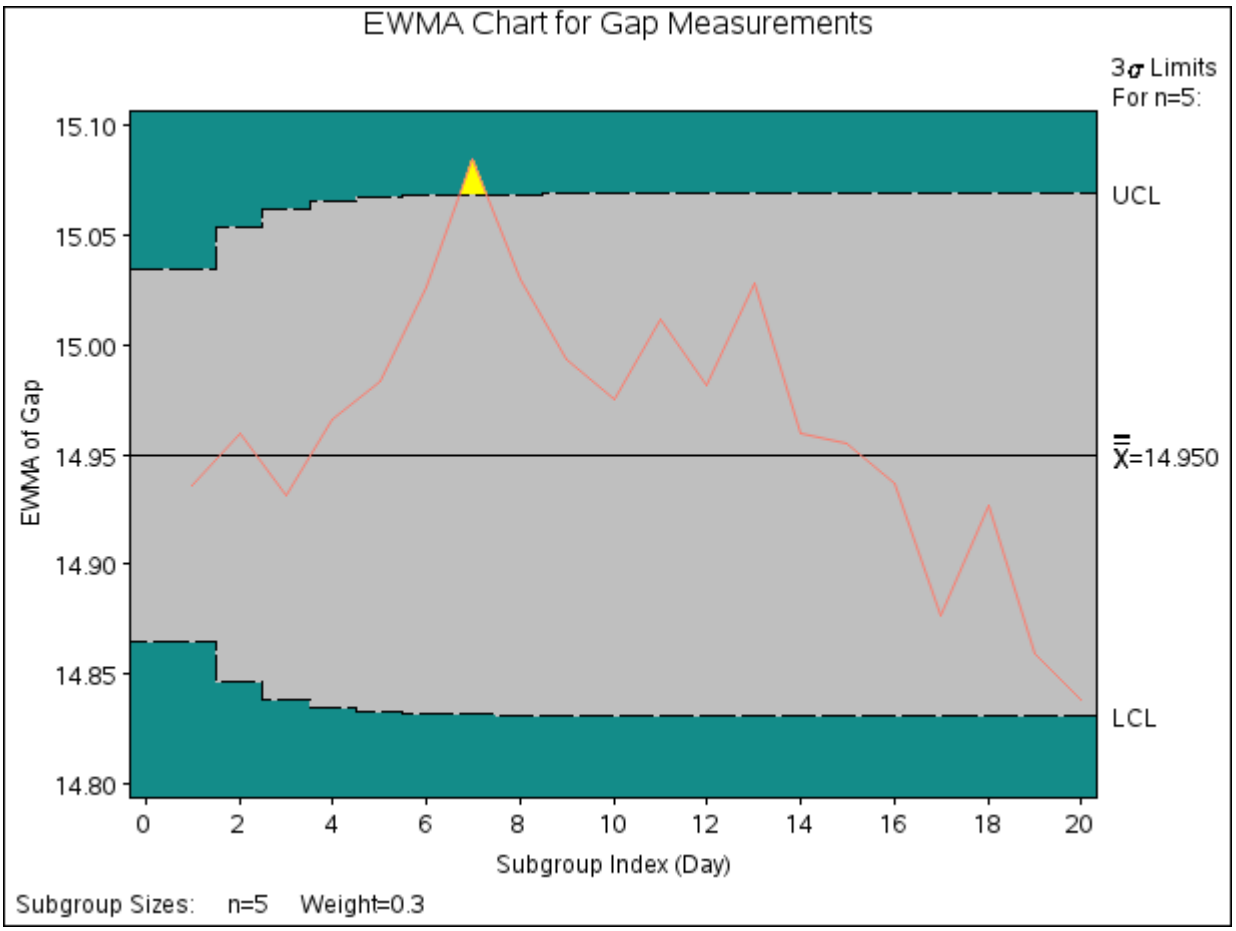
You can read this data set by specifying it as a HISTORY= data set in the PROC MACONTROL statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
symbol color=salmon h=0.8;
title 'EWMA Chart for Gap Measurements';
proc macontrol history=Clipsum;
   ewmachart Gap*Day / weight=0.3
                       cframe = vibg
                       cinfill = ligr
                       coutfill = yellow
                       cconnect = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the GOPTIONS and SYMBOL statements and EWMACHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting EWMA chart is shown in Figure 10.5.

Note that Gap is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables GapX, GapS, and GapN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (Gap), which is referred to as the *process*. The variables GapX, GapS, and GapN are all required. The name Day specified after the asterisk is the name of the *subgroup-variable*.

Figure 10.5 EWMA Chart from Summary Data



In general, a HISTORY= input data set used with the EWMACHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the EWMACHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC MACONTROL statement to rename the variables for the duration of the MACONTROL procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1885 for an example of the RENAME option).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 830.

Saving Summary Statistics

NOTE: See *Exponentially Weighted Moving Average Chart* in the SAS/QC Sample Library.

In this example, the EWMACHART statement is used to create a summary data set that can be read later by the MACONTROL procedure (as in the preceding example). The following statements read measurements from the data set Clips1 and create a summary data set named Cliphist:

```
title 'Summary Data Set for Gap Measurements';
proc macontrol data=Clips1;
   ewmachart Gap*Day / weight     = 0.3
                       outhistory = Cliphist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 10.3.

Figure 10.6 contains a partial listing of Cliphist.

Figure 10.6 The Summary Data Set Cliphist

Summary Data Set for Gap Measurements

Day	GapX	GapS	GapE	GapN
1	14.904	0.18716	14.9362	5
2	15.014	0.09317	14.9595	5
3	14.866	0.25006	14.9315	5
4	15.048	0.23732	14.9664	5
5	15.024	0.26792	14.9837	5

There are five variables in the data set Cliphist.

- Day contains the subgroup index.
- GapX contains the subgroup means.
- GapS contains the subgroup standard deviations.
- GapE contains the subgroup exponentially weighted moving averages.
- GapN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, *E*, and *N* to the *process* Gap specified in the EWMACHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 826.

Saving Control Limit Parameters

NOTE: See *Exponentially Weighted Moving Average Chart* in the SAS/QC Sample Library.

You can save the control limit parameters for an EWMA chart in a SAS data set; this enables you to use these parameters with future data (see "Reading Preestablished Control Limit Parameters" on page 805) or modify the parameters with a DATA step program.

The following statements read measurements from the data set Clips1 (see "Creating EWMA Charts from Raw Data" on page 796) and save the control limit parameters in a data set named Cliplim:

```
title 'Control Limit Parameters';
proc macontrol data=Clips1;
   ewmachart Gap*Day / weight    = 0.3
                       outlimits = Cliplim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limit parameters, and the NOCHART option suppresses the display of the chart. The data set Cliplim is listed in Figure 10.7.

Figure 10.7 The Data Set Cliplim Containing Control Limit Information

Control Limit Parameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_MEAN_	_STDDEV_	_WEIGHT_
Gap	Day	ESTIMATE	5	.002699796	3	14.95	0.21108	0.3

Note that the data set Cliplim does not contain the actual control limits but rather the parameters required to compute the limits.

The data set contains one observation with the parameters for *process* Gap. The variable _WEIGHT_ contains the weight parameter used to compute the EWMAs. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates that the values of _MEAN_ and _STDDEV_ are estimates rather than standard values. For more information, see "OUTLIMITS= Data Set" on page 825.

You can create an output data set containing the control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limits';
proc macontrol data=Clips1;
   ewmachart Gap*Day / weight    = 0.3
                       outtable = Cliptab
                       nochart;
run;
```

The data set Cliptab is listed in Figure 10.8.

Figure 10.8 The OUTTABLE= Data Set Cliptab

Summary Statistics and Control Limits

VAR	Day	_SIGMAS_	_LIMITN_	_WEIGHT_	_SUBN_	_SUBX_	_SUBS_	_LCLE_	_EWMA_
Gap	1	3	5	0.3	5	14.904	0.18716	14.8650	14.9362
Gap	2	3	5	0.3	5	15.014	0.09317	14.8463	14.9595
Gap	3	3	5	0.3	5	14.866	0.25006	14.8383	14.9315
Gap	4	3	5	0.3	5	15.048	0.23732	14.8345	14.9664
Gap	5	3	5	0.3	5	15.024	0.26792	14.8327	14.9837
Gap	6	3	5	0.3	5	15.126	0.12260	14.8319	15.0264
Gap	7	3	5	0.3	5	15.220	0.23098	14.8314	15.0845
Gap	8	3	5	0.3	5	14.902	0.17254	14.8312	15.0297
Gap	9	3	5	0.3	5	14.910	0.19824	14.8311	14.9938
Gap	10	3	5	0.3	5	14.932	0.24035	14.8311	14.9753
Gap	11	3	5	0.3	5	15.096	0.25618	14.8311	15.0115
Gap	12	3	5	0.3	5	14.912	0.16903	14.8310	14.9816
Gap	13	3	5	0.3	5	15.138	0.15928	14.8310	15.0285
Gap	14	3	5	0.3	5	14.798	0.26329	14.8310	14.9594
Gap	15	3	5	0.3	5	14.944	0.20876	14.8310	14.9548
Gap	16	3	5	0.3	5	14.896	0.09965	14.8310	14.9371
Gap	17	3	5	0.3	5	14.734	0.22512	14.8310	14.8762
Gap	18	3	5	0.3	5	15.046	0.24141	14.8310	14.9271
Gap	19	3	5	0.3	5	14.702	0.17880	14.8310	14.8596
Gap	20	3	5	0.3	5	14.788	0.16634	14.8310	14.8381

MEAN	_UCLE_	_STDDEV_	_EXLIM_
14.95	15.0350	0.21108	
14.95	15.0537	0.21108	
14.95	15.0617	0.21108	
14.95	15.0655	0.21108	
14.95	15.0673	0.21108	
14.95	15.0681	0.21108	
14.95	15.0686	0.21108	UPPER
14.95	15.0688	0.21108	
14.95	15.0689	0.21108	
14.95	15.0689	0.21108	
14.95	15.0689	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	
14.95	15.0690	0.21108	

This data set contains one observation for each subgroup sample. The variable _EWMA_ contains the EWMAs. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes, respectively. The variables _LCLE_ and _UCLE_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Day contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 827.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Cliptab and display a EWMA chart (not shown here) identical to Figure 10.3:

```
title 'EWMA Chart for Gap Measurements';
proc macontrol table=Cliptab;
   ewmachart Gap*Day ;
run;
```

For more information, see "TABLE= Data Set" on page 831.

Reading Preestablished Control Limit Parameters

NOTE: See *Exponentially Weighted Moving Average Chart* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set saved the control limit parameters in the data set Cliplim. This example shows how to apply these parameters to new data provided in the following data set:

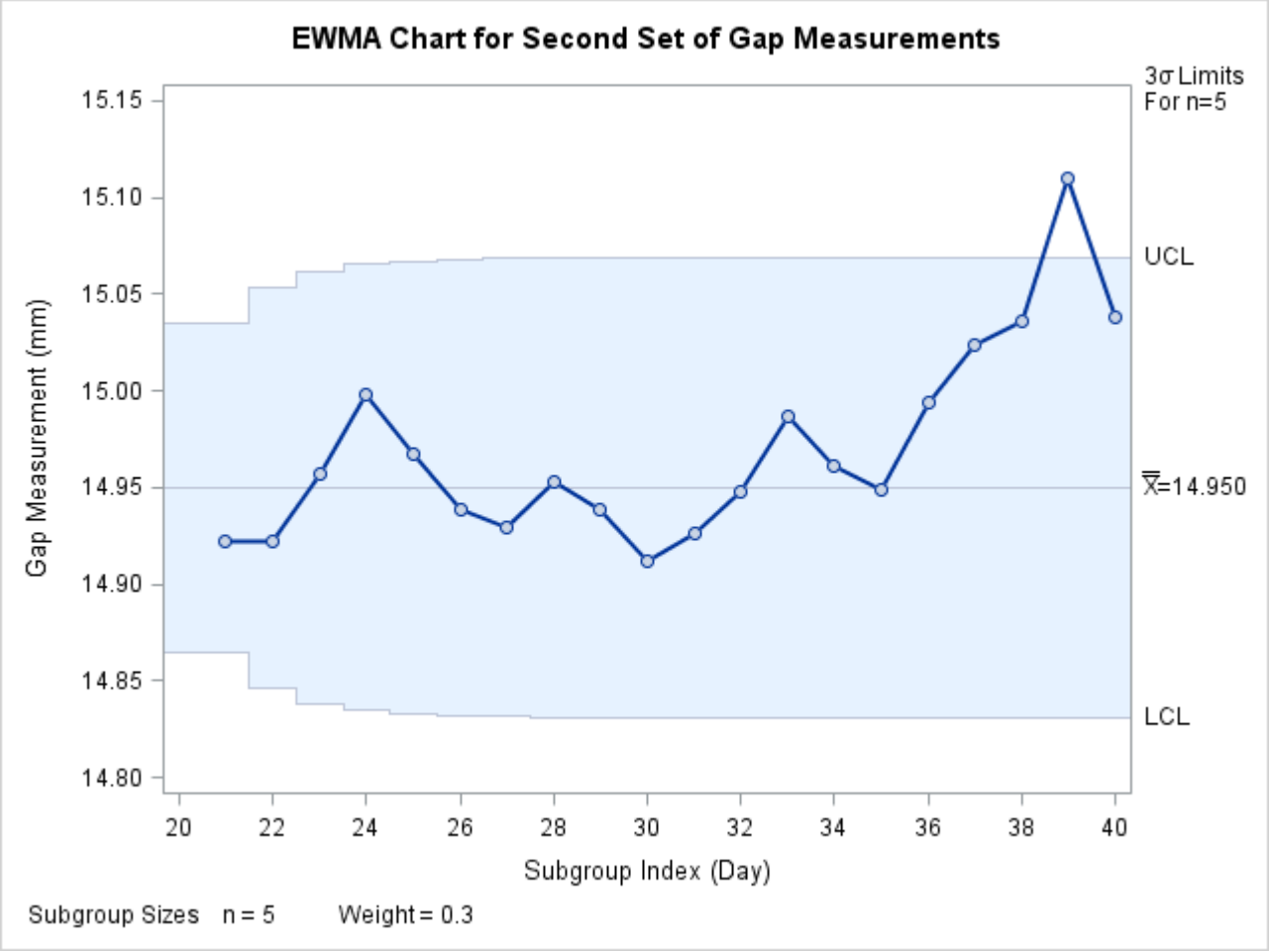
```
data Clips1a;
   label Gap='Gap Measurement (mm)';
   input Day @;
   do i=1 to 5;
      input Gap @;
      output;
   end;
   drop i;
   datalines;
21  14.86 15.01 14.67 14.67 15.07
22  14.93 14.53 15.07 15.10 14.98
23  15.27 14.90 15.12 15.10 14.80
24  15.02 15.21 14.93 15.11 15.20
25  14.90 14.81 15.26 14.57 14.94
26  14.78 15.29 15.13 14.62 14.54
27  14.78 15.15 14.61 14.92 15.07
28  14.92 15.31 14.82 14.74 15.26
29  15.11 15.04 14.61 15.09 14.68
30  15.00 15.04 14.36 15.20 14.65
31  14.99 14.76 15.18 15.04 14.82
32  14.90 14.78 15.19 15.06 15.06
33  14.95 15.10 14.86 15.27 15.22
34  15.03 14.71 14.75 14.99 15.02
35  15.38 14.94 14.68 14.77 14.83
36  14.95 15.43 14.87 14.90 15.34
37  15.18 14.94 15.32 14.74 15.29
38  14.91 15.15 15.06 14.78 15.42
39  15.34 15.34 15.41 15.36 14.96
40  15.12 14.75 15.05 14.70 14.74
;
```

The following statements create an EWMA chart for the data in Clips1a using the control limit parameters in Cliplim:

```
ods graphics on;
title 'EWMA Chart for Second Set of Gap Measurements';
proc macontrol data=Clips1a limits=Cliplim;
   ewmachart Gap*Day / odstitle=title markers;
run;
```

The ODS GRAPHICS ON statement specified before the PROC MACONTROL statement enables ODS Graphics, so the EWMA chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 10.9.

Figure 10.9 EWMA Chart Using Preestablished Control Limit Parameters



The LIMITS= option in the PROC MACONTROL statement specifies the data set containing the control limit parameters. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Gap
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

Note that the EWMA plotted for the 39th day lies above the upper control limit, signalling an out-of-control process.

In this example, the LIMITS= data set was created in a previous run of the MACONTROL procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 829 for details concerning the variables that you must provide, and see Example 10.1 for an illustration.

Syntax: EWMACHART Statement

The basic syntax for the EWMACHART statement is as follows:

EWMACHART *process* ∗ *subgroup-variable* / **WEIGHT=***value* < *options* > **;**

The general form of this syntax is as follows:

EWMACHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / **WEIGHT=***value* < *options* > **;**

Note that the WEIGHT= option is required unless its *value* is read from a LIMITS= data set. You can use any number of EWMACHART statements in the MACONTROL procedure. The components of the EWMACHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC MACONTROL statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating EWMA Charts from Raw Data" on page 796.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating EWMA Charts from Subgroup Summary Data" on page 799.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limit Parameters" on page 803.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct EWMA charts (each using a weight parameter of 0.3) for Weight, Length, and Width:

```
proc macontrol data=Measures;
   ewmachart (Weight Length Width)*Day / weight=0.3;
run;
```

subgroup-variable

is the variable that classifies the data into subgroups. The *subgroup-variable* is required. In the preceding EWMACHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1970.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or plotting character used to plot the EWMAs.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create an EWMA chart using an asterisk (*) to plot the points:

```
proc macontrol data=Values lineprinter;
   ewmachart Length*Hour='*' / weight=0.3;
run;
```

options
specify chart parameters, enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" on page 808, which follows, lists all options by function.

Summary of Options

The following tables list the EWMACHART statement options by function. Options unique to the MACONTROL procedure are listed in Table 10.2, and are described in detail in the section "Dictionary of Special Options" on page 817. Options that are common to both the MACONTROL and SHEWHART procedures are listed in Table 10.3. They are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 10.2 EWMACHART Statement Special Options

Option	Description
Options for Specifying Exponentially Weighted Moving Average Charts	
ALPHA=	Requests probability limits for control charts
ASYMPTOTIC	Requests constant control limits based on asymptotic expressions
LIMITN=	Specifies either a fixed nominal sample size (n) for control limits or allows the control limits to vary with subgroup sample size

Table 10.2 *continued*

Option	Description
MU0=	Specifies a standard (known) value μ_0 for the process mean
NOREADLIMITS	Specifies that control limit parameters are not to be read from a LIMITS= data set
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from the LIMITS= data set when both variables are available
READINDEX=	Reads control limit parameters from the first observation in the LIMITS= data set where the variable _INDEX_ equals *value*
READLIMITS	Reads control limit parameters from a LIMITS= data set (SAS 6.09 and earlier releases)
RESET	Requests that the value of the EWMA be reset after each out-of-control point
SIGMA0=	Specifies standard (known) value σ_0 for process standard deviation
SIGMAS=	Specifies width of control limits in terms of multiple of standard error of plotted EWMAs
WEIGHT=	Specifies weight assigned to the most recent subgroup mean in the computation of the EWMA
Options for Plotting Subgroup Means	
CMEANSYMBOL=	Specifies color for MEANSYMBOL= symbol
MEANCHAR=	Specifies *character* to plot subgroup means on line printer charts
MEANSYMBOL=	Specifies symbol to plot subgroup means in traditional graphics

Table 10.3 EWMACHART Statement General Options

Option	Description
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits

Table 10.3 *continued*

Option	Description
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line
Process Mean and Standard Deviation Options	
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on EWMA chart
ALLLABEL2=	Labels every point on trend chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
NOTRENDCONNECT	Suppresses line segments that connect points on trend chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles

Table 10.3 *continued*

Option	Description
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of EWMA chart
VAXIS2=	Specifies major tick mark values for vertical axis of trend chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines

Table 10.3 *continued*

Option	Description
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	Specifies the natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	Specifies the maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
TRENDVAR=	Specifies list of trend variables
YPCT1=	Specifies length of vertical axis on EWMA chart as a percentage of sum of lengths of vertical axes for EWMA and trend charts
ZEROSTD	Displays EWMA chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on EWMA chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on EWMA chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines

Table 10.3 *continued*

Option	Description
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on EWMA chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	Specifies the position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to EWMA chart
ANNOTATE2=	Specifies annotate data set that adds features to trend chart
DESCRIPTION=	Specifies description of EWMA chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of EWMA chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination

Table 10.3 *continued*

Option	Description
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
WTREND=	Specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	Draws portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Draws areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title

Table 10.3 *continued*

Option	Description
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	Creates all the tables that are produced by the TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS options
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend

Table 10.3 *continued*

Option	Description
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	Superimposes star at each point on EWMA chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars

Table 10.3 *continued*

Option	Description
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
VREFCHAR=	Specifies line character for VREF= and VREF2= lines

Dictionary of Special Options

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a single EWMA exceeds its control limits. The value of α can range between 0 and 1. This assumes that the process is in statistical control and that the data follow a normal distribution. For the equations used to compute probability limits, see "Control Limits" on page 821.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option.
- As an alternative to specifying ALPHA=α (or reading _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

ASYMPTOTIC

requests constant upper and lower control limits based on the following asymptotic expressions:

$$\mathrm{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r/n(2-r)}$$

$$\mathrm{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r/n(2-r)}$$

Here r is the weight parameter ($0 < r \leq 1$), and n is the nominal sample size associated with the control limits. Substitute $\Phi^{-1}(1 - \alpha/2)$ for k if you specify probability limits with the ALPHA= option. When you do not specify the ASYMPTOTIC option, the control limits are computed using the exact formulas in Table 10.5. Use the ASYMPTOTIC option only if all the subgroup sample sizes are the same or if you specify LIMITN=n. See Example 10.2.

CMEANSYMBOL=*color*
specifies the *color* used for the symbol requested with the MEANSYMBOL= option in traditional graphics. This option is ignored unless you are producing traditional graphics.

LIMITN=*n*
LIMITN=VARYING
specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=n, EWMAs are calculated and displayed only for those subgroups with a sample size equal to n, unless you also specify the ALLN option, which causes all the EWMAs to be calculated and displayed. By default (or if you specify LIMITN=VARYING), EWMAs are calculated and displayed for all subgroups, regardless of sample size.

MEANCHAR='*character*'
specifies a *character* used in legacy line printer charts to plot the subgroup mean for each subgroup. By default, subgroup means are not plotted. This option is ignored unless you specify the LINEPRINTER option in the PROC MACONTROL statement.

MEANSYMBOL=*keyword*
specifies a symbol used to plot the subgroup mean for each subgroup in traditional graphics. By default, subgroup means are not plotted. This option is ignored unless you are producing traditional graphics.

MU0=*value*
specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data. See Example 10.1.

NOTE: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set.

NOREADLIMITS
specifies that control limit parameters for each *process* listed in the EWMACHART statement are *not* to be read from the LIMITS= data set specified in the PROC MACONTROL statement.

The following example illustrates the NOREADLIMITS option:

```
proc macontrol data=Pistons limits=Diamlim;
   ewmachart Diameter*Hour;
   ewmachart Diameter*Hour / noreadlimits weight=0.3;
run;
```

The first EWMACHART statement reads the control limits from the first observation in the data set Diamlim for which the variable _VAR_ is equal to 'Diameter' and the variable _SUBGRP_ is equal to 'Hour'. The second EWMACHART statement computes estimates of the process mean and standard deviation for the control limits from the measurements in the data set Pistons. Note that the second EWMACHART statement is equivalent to the following statements, which would be more commonly used:

```
proc macontrol data=Pistons;
   ewmachart Diameter*Hour / weight=0.3;
run;
```

For more information about reading control limit parameters from a LIMITS= data set, see the READLIMITS option later in this list.

READALPHA

specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from a LIMITS= data set when both variables are available in the data set. Thus the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default.

READINDEX=*'value'*

reads control limit parameters from a LIMITS= data set (specified in the PROC MACONTROL statement) for each *process* listed in the EWMACHART statement.

The control limit parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable*
- the value of _INDEX_ matches *value*

The *value* can be up to 48 characters and must be enclosed in quotes.

READLIMITS

specifies that control limit parameters are to be read from a LIMITS= data set specified in the PROC MACONTROL statement. The parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

NOTE: In SAS 6.10 and later releases, the READLIMITS option is not necessary.

RESET

requests that the value of the EWMA be reset after each out-of-control point. Specifically, when a point exceeds the control limits, the EWMA for the next subgroup is computed as the weighted average of the subgroup mean and the overall mean. By default, the EWMAs are not reset.

SIGMA0=*value*

specifies a known (standard) value σ_0 for the process standard deviation σ. The *value* must be positive. By default, the MACONTROL procedure estimates σ from the data using the formulas given in "Methods for Estimating the Standard Deviation" on page 832.

NOTE: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set.

SIGMAS=*value*
specifies the width of the control limits in terms of the multiple k of the standard error of the plotted EWMAs on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are 3σ limits.

WEIGHT=*value*
specifies the weight r assigned to the most recent subgroup mean in the computation of the EWMA ($0 < r \leq 1$). The WEIGHT= option is required unless you read control limit parameters from a LIMITS= data set or a TABLE= data set. See the section "Choosing the Value of the Weight Parameter" on page 822 for details.

Details: EWMACHART Statement

Constructing EWMA Charts

The following notation is used in this section:

E_i	Exponentially weighted moving average for the ith subgroup
r	EWMA weight parameter ($0 < r \leq 1$)
μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
x_{ij}	jth measurement in ith subgroup, with $j = 1, 2, 3, \ldots, n_i$
n_i	Sample size of ith subgroup
$\overline{X}_i$	Mean of measurements in ith subgroup. If $n_i = 1$, then the subgroup mean reduces to the single observation in the subgroup
$\overline{\overline{X}}$	Weighted average of subgroup means
$\Phi^{-1}(\cdot)$	Inverse standard normal function

Plotted Points

Each point on the chart indicates the value of the exponentially weighted moving average (EWMA) for that subgroup. The EWMA for the ith subgroup (E_i) is defined recursively as

$$E_i = r\overline{X}_i + (1-r)E_{i-1}, \quad i > 0$$

where r is a weight parameter ($0 < r \leq 1$). Some authors (for example, Hunter 1986 and Crowder 1987a,b) use the symbol λ instead of r for the weight. You can specify the weight with the WEIGHT= option in the EWMACHART statement or with the variable _WEIGHT_ in a LIMITS= data set. If you specify a known value (μ_0) for μ, $E_0 = \mu_0$; otherwise, $E_0 = \overline{\overline{X}}$.

The preceding equation can be rewritten as

$$E_i = E_{i-1} + r(\overline{X}_i - E_{i-1})$$

which expresses the current EWMA as the previous EWMA plus the weighted error in the prediction of the current mean based on the previous EWMA.

The EWMA for the ith subgroup can also be written as

$$E_i = r\sum_{j=0}^{i-1}(1-r)^j\overline{X}_{i-j} + (1-r)^i E_0$$

which expresses the EWMA as a weighted average of past subgroup means, where the weights decline exponentially, and the heaviest weight is assigned to the most recent subgroup mean.

Central Line

By default, the central line on an EWMA chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1 \bar{X}_1 + \cdots + n_N \bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of E_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that E_i exceeds the limits

Table 10.5 presents the formulas for the limits.

Table 10.5 Limits for an EWMA Chart

Control Limits
LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma} r \sqrt{\sum_{j=0}^{i-1} (1-r)^{2j}/n_{i-j}}$
UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma} r \sqrt{\sum_{j=0}^{i-1} (1-r)^{2j}/n_{i-j}}$
Probability Limits
LCL = lower limit = $\overline{\overline{X}} - \Phi^{-1}(1-\alpha/2)\hat{\sigma} r \sqrt{\sum_{j=0}^{i-1} (1-r)^{2j}/n_{i-j}}$
UCL = upper limit = $\overline{\overline{X}} + \Phi^{-1}(1-\alpha/2)\hat{\sigma} r \sqrt{\sum_{j=0}^{i-1} (1-r)^{2j}/n_{i-j}}$

These formulas assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 10.5. Note that the limits vary with both n_i and i.

If the subgroup sample sizes are constant ($n_i = n$), the formulas for the control limits simplify to

$$\text{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r(1-(1-r)^{2i})/n(2-r)}$$

$$\text{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r(1-(1-r)^{2i})/n(2-r)}$$

Consequently, when the subgroup sample sizes are constant, the width of the control limits increases monotonically with i. For probability limits, replace k with $\Phi^{-1}(1-\alpha/2)$ in the previous equations. Refer to Roberts (1959) and Montgomery (1996).

As i becomes large, the upper and lower control limits approach constant values:

$$\mathrm{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r/n(2-r)}$$

$$\mathrm{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r/n(2-r)}$$

Some authors base the control limits for EWMA charts on the asymptotic expressions in the two previous equations. For asymptotic probability limits, replace k with $\Phi^{-1}(1-\alpha/2)$ in these equations. You can display asymptotic limits by specifying the ASYMPTOTIC option.

Uniformly weighted moving average charts and exponentially weighted moving average charts have similar properties, and their asymptotic control limits are identical provided that

$$r = 2/(w+1)$$

where w is the weight factor for uniformly weighted moving average charts. Refer to Wadsworth, Stephens, and Godfrey (1986) and the American Society for Quality Control (1983).

You can specify parameters for the EWMA limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify r with the WEIGHT= option or with the variable _WEIGHT_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Choosing the Value of the Weight Parameter

Various approaches have been proposed for choosing the value of r.

- Hunter (1986) states that the choice "can be left to the judgment of the quality control analyst" and points out that the smaller the value of r, "the greater the influence of the historical data."
- Hunter (1986) also discusses a least squares procedure for estimating r from the data, **assuming an exponentially weighted moving average model for the data**. In this context, the fitted EWMA model provides a forecast of the process that is the basis for dynamic process control. You can use the ARIMA procedure in SAS/ETS® software to compute the least squares estimate of r. (Refer to *SAS/ETS User's Guide* for information about PROC ARIMA.) Also see "Autocorrelation in Process Data" on page 2144.
- A number of authors have studied the design of EWMA control schemes based on average run length (ARL) computations. The ARL is the expected number of points plotted before a shift is detected. Ideally, the ARL should be short when a shift occurs, and it should be long when there is no shift (the process is in control.) The effect of r on the ARL was described by Roberts (1959), who used simulation

methods. The ARL function was approximated and tabulated by Robinson and Ho (1978), and a more general method for studying run-length distributions of EWMA charts was given by Crowder (1987a, b). Unlike Hunter (1986), these authors assume the data are independent and identically distributed; typically the normal distribution is assumed for the data, although the methods extend to nonnormal distributions. A more detailed discussion of the ARL approach follows.

Average run lengths for two-sided EWMA charts are shown in Table 10.6, which is patterned after Table 1 of Crowder (1987a, b). The ARLs were computed using the EWMAARL DATA step function (see "EWMAARL Function" on page 2228 for details on the EWMAARL function). Note that Crowder (1987a, b). uses the notation L in place of k and the notation λ in place of r.

You can use Table 10.6 to find a combination of k and r that yields a desired ARL for an in-control process ($\delta = 0$) and for a specified shift of δ. Note that δ is assumed to be standardized; in other words, if a shift of Δ is to be detected in the process mean μ, and if σ is the process standard deviation, you should select the table entry with

$$\delta = \Delta/(\sigma/\sqrt{n})$$

where n is the subgroup sample size. Thus, δ can be regarded as the shift in the sampling distribution of the subgroup mean.

For example, suppose you want to construct an EWMA scheme with an in-control ARL of 90 and an ARL of 9 for detecting a shift of $\delta = 1$. Table 10.6 shows that the combination $r = 0.5$ and $k = 2.5$ yields an in-control ARL of 91.17 and an ARL of 8.27 for $\delta = 1$.

Crowder (1987a, b) cautions that setting the in-control ARL at a desired level does not guarantee that the probability of an early false signal is acceptable. For further details concerning the distribution of the ARL, refer to Crowder (1987a, b).

In addition to using Table 10.6 or the EWMAARL DATA step function to choose a EWMA scheme with desired average run length properties, you can use them to evaluate an existing EWMA scheme. For example, the "Getting Started" section of this chapter contains EWMA schemes with $r = 0.3$ and $k = 3$. The following statements use the EWMAARL function to compute the in-control ARL and the ARLs for shifts of $\delta = 0.25$ and $\delta = 0.5$:

```
data arlewma;
   arlin = ewmaarl(  0,0.3,3.0);
   arl1  = ewmaarl(.25,0.3,3.0);
   arl2  = ewmaarl(.50,0.3,3.0);
run;
```

The in-control ARL is 465.553, the ARL for $\delta = .25$ is 178.741, and the ARL for $\delta = .5$ is 53.1603. See Example 10.5 for an illustration of how to use the EWMAARL function to compute average run lengths for various EWMA schemes and shifts.

Table 10.6 Average Run Lengths for Two-Sided EWMA Charts

		***r* (weight parameter)**					
k	δ	**0.05**	**0.10**	**0.25**	**0.50**	**0.75**	**1.00**
2.0	0.00	127.53	73.28	38.56	26.45	22.88	21.98
2.0	0.25	43.94	34.49	24.83	20.12	18.86	19.13
2.0	0.50	18.97	15.53	12.74	11.89	12.34	13.70

Table 10.6 *continued*

k	δ	**0.05**	**0.10**	**0.25**	**0.50**	**0.75**	**1.00**
2.0	0.75	11.64	9.36	7.62	7.29	7.86	9.21
2.0	1.00	8.38	6.62	5.24	4.91	5.26	6.25
2.0	1.25	6.56	5.13	3.96	3.59	3.76	4.40
2.0	1.50	5.41	4.20	3.19	2.80	2.84	3.24
2.0	1.75	4.62	3.57	2.68	2.29	2.26	2.49
2.0	2.00	4.04	3.12	2.32	1.95	1.88	2.00
2.0	2.25	3.61	2.78	2.06	1.70	1.61	1.67
2.0	2.50	3.26	2.52	1.85	1.51	1.42	1.45
2.0	2.75	2.99	2.32	1.69	1.37	1.29	1.29
2.0	3.00	2.76	2.16	1.55	1.26	1.19	1.19
2.0	3.25	2.56	2.03	1.43	1.18	1.13	1.12
2.0	3.50	2.39	1.93	1.32	1.12	1.08	1.07
2.0	3.75	2.26	1.83	1.24	1.08	1.05	1.04
2.0	4.00	2.15	1.73	1.17	1.05	1.03	1.02
2.5	0.00	379.09	223.35	124.18	91.17	82.49	80.52
2.5	0.25	73.98	66.59	59.66	58.33	61.07	65.77
2.5	0.50	26.63	23.63	23.28	27.16	33.26	41.49
2.5	0.75	15.41	12.95	11.96	13.96	18.05	24.61
2.5	1.00	10.79	8.75	7.52	8.27	10.57	14.92
2.5	1.25	8.31	6.60	5.39	5.52	6.75	9.46
2.5	1.50	6.78	5.31	4.18	4.03	4.65	6.30
2.5	1.75	5.75	4.46	3.43	3.14	3.43	4.41
2.5	2.00	5.00	3.86	2.92	2.57	2.67	3.24
2.5	2.25	4.43	3.42	2.56	2.18	2.17	2.49
2.5	2.50	4.00	3.07	2.29	1.90	1.83	2.00
2.5	2.75	3.64	2.80	2.08	1.69	1.59	1.67
2.5	3.00	3.36	2.57	1.91	1.52	1.41	1.45
2.5	3.25	3.12	2.39	1.77	1.39	1.29	1.29
2.5	3.50	2.92	2.24	1.64	1.28	1.19	1.19
2.5	3.75	2.74	2.13	1.52	1.20	1.13	1.12
2.5	4.00	2.58	2.04	1.42	1.13	1.08	1.07
3.0	0.00	1383.62	842.15	502.90	397.46	374.50	370.40
3.0	0.25	133.61	144.74	171.09	208.54	245.76	281.15
3.0	0.50	37.33	37.41	48.45	75.35	110.95	155.22
3.0	0.75	19.95	17.90	20.16	31.46	50.92	81.22
3.0	1.00	13.52	11.38	11.15	15.74	25.64	43.89
3.0	1.25	10.24	8.32	7.39	9.21	14.26	24.96
3.0	1.50	8.26	6.57	5.47	6.11	8.72	14.97
3.0	1.75	6.94	5.45	4.34	4.45	5.80	9.47
3.0	2.00	6.00	4.67	3.62	3.47	4.15	6.30
3.0	2.25	5.30	4.10	3.11	2.84	3.16	4.41
3.0	2.50	4.76	3.67	2.75	2.41	2.52	3.24
3.0	2.75	4.32	3.32	2.47	2.10	2.09	2.49
3.0	3.00	3.97	3.05	2.26	1.87	1.79	2.00
3.0	3.25	3.67	2.82	2.09	1.69	1.57	1.67

Table 10.6 *continued*

k	δ	**0.05**	**0.10**	**0.25**	**0.50**	**0.75**	**1.00**
3.0	3.50	3.42	2.62	1.95	1.53	1.41	1.45
3.0	3.75	3.22	2.45	1.84	1.41	1.29	1.29
3.0	4.00	3.04	2.30	1.73	1.31	1.20	1.19
3.5	0.00	12851.0	4106.4	2640.16	2227.34	2157.99	2149.34
3.5	0.25	281.09	381.29	625.78	951.18	1245.90	1502.76
3.5	0.50	53.58	64.72	123.43	267.36	468.68	723.81
3.5	0.75	25.62	25.33	38.68	88.70	182.12	334.40
3.5	1.00	16.65	14.79	17.71	35.97	78.05	160.95
3.5	1.25	12.36	10.37	10.48	17.64	37.15	81.80
3.5	1.50	9.86	8.00	7.25	10.19	19.63	43.96
3.5	1.75	8.22	6.54	5.52	6.70	11.46	24.96
3.5	2.00	7.07	5.55	4.47	4.86	7.33	14.97
3.5	2.25	6.21	4.83	3.77	3.78	5.08	9.47
3.5	2.50	5.55	4.29	3.28	3.10	3.76	6.30
3.5	2.75	5.03	3.87	2.91	2.63	2.94	4.41
3.5	3.00	4.60	3.54	2.63	2.30	2.40	3.24
3.5	3.25	4.25	3.26	2.41	2.05	2.03	2.49
3.5	3.50	3.95	3.03	2.23	1.85	1.76	2.00
3.5	3.75	3.70	2.84	2.10	1.69	1.56	1.67
3.5	4.00	3.47	2.66	1.99	1.55	1.40	1.45

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves the control limit parameters. Table 10.7 lists the variables that can be saved.

Table 10.7 OUTLIMITS= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding limits
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LIMITN	Sample size associated with the control limits
MEAN	Process mean ($\overline{\overline{X}}$ or μ_0)
SIGMAS	Multiple (k) of standard error of E_i
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the EWMACHART statement
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
VAR	*Process* specified in the EWMACHART statement
WEIGHT	Weight (r) assigned to most recent subgroup mean in computation of EWMA

The OUTLIMITS= data set does not contain the control limits; instead, it contains control limit parameters that can be used to recompute the control limits.

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variable _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of E_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the EWMACHART statement.

You can use OUTLIMITS= data sets

- to keep a permanent record of the control limit parameters
- to write reports. You may prefer to use OUTTABLE= data sets for this purpose.
- as LIMITS= data sets in subsequent runs of PROC MACONTROL

For an example of an OUTLIMITS= data set, see the section "Saving Control Limit Parameters" on page 803.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup EWMA variable named by *process* suffixed with *E*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the EWMACHART statement. For example, consider the following statements:

```
proc macontrol data=Clips;
   ewmachart (Gap YieldStrength)*Day /
      weight     = 0.2
      outhistory = Cliphist;
run;
```

The data set Cliphist would contain nine variables named Day, GapX, GapS, GapE, GapN, YieldStrengthX, YieldStrengthS, YieldStrengthE, and YieldStrengthN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see the section "Saving Summary Statistics" on page 802.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 10.8 lists the variables that can be saved.

Table 10.8 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on EWMA chart
EWMA	Exponentially weighted moving average
LCLE	Lower control limit for EWMA
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
SUBX	Subgroup mean
UCLE	Upper control limit for EWMA
VAR	*Process* specified in the EWMACHART statement
WEIGHT	Weight (r) assigned to most recent subgroup mean in computation of EWMA

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- *symbol-variable*

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variables _VAR_ and _EXLIM_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limit Parameters" on page 803.

ODS Tables

The following table summarizes the ODS tables that you can request with the EWMACHART statement.

Table 10.9 ODS Tables Produced with the EWMACHART Statement

Table Name	Description	Options
EWMAChartSummary	Exponentially weighted moving average chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT
Parameters	Exponentially weighted moving average parameters	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. EWMACHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the EWMACHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 10.10.

Table 10.10 ODS Graphics Produced by the EWMACHART Statement

ODS Graph Name	Plot Description
EWMAChart	EWMA chart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC MACONTROL statement. Each *process* specified in the EWMACHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the EWMACHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the ith subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the ith subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating EWMA Charts from Raw Data" on page 796.

LIMITS= Data Set

You can read preestablished control limit parameters from a LIMITS= data set specified in the PROC MACONTROL statement. The LIMITS= data set used by the MACONTROL procedure does not contain the actual control limits, but rather it contains the parameters required to compute the limits. For example, the following statements read parameters from the data set Parms:

```
proc macontrol data=Parts limits=Parms;
   ewmachart Gap*Day;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the MACONTROL procedure. Such data sets always contain the variables required for a LIMITS= data set; see the section "OUTLIMITS= Data Set" on page 825. The LIMITS= data set can also be created directly using a DATA step.

When you create a LIMITS= data set, you must provide the variable _WEIGHT_, which specifies the weight parameter used to compute the EWMAs. In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables of length 8.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are 'ESTIMATE', 'STANDARD', 'STDMEAN', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

Some advantages of working with a LIMITS= data set are that

- it facilitates reusing a permanently saved set of parameters
- a distinct set of parameters can be read for each *process* specified in the EWMACHART statement
- it facilitates keeping track of multiple sets of parameters that accumulate for the same *process* as the process evolves over time

For an example, see the section "Reading Preestablished Control Limit Parameters" on page 805.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC MACONTROL statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the MACONTROL, SHEWHART, or CUSUM procedures or to read output data sets created with SAS summarization procedures such as PROC MEANS.

A HISTORY= data set used with the EWMACHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- a subgroup standard deviation variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters *X*, *S*, and *N*, respectively.

For example, consider the following statements:

```
proc macontrol history=Cliphist;
   ewmachart (Gap Diameter)*Day  / weight=0.2;
run;
```

The data set Cliphist must include the variables Day, GapX, GapS, GapN, DiameterX, DiameterS, and DiameterN.

Although a subgroup EWMA variable (named by the *process* name suffixed with *E*) is saved in an OUTHISTORY= data set, it is not required in a HISTORY= data set, because the subgroup mean variable is sufficient to compute the EWMAs.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all the observations in a HISTORY= data set. However, if the HISTORY= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating EWMA Charts from Subgroup Summary Data" on page 799.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC MACONTROL statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the MACONTROL procedure.

Table 10.11 lists the variables required in a TABLE= data set used with the EWMACHART statement:

Table 10.11 TABLE= Data Set Variables

Variable	Description
EWMA	Exponentially weighted moving average
LCLE	Lower control limit for EWMA
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
Subgroup-variable	Values of the *subgroup-variable*

Table 10.11 *continued*

Variable	Description
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
SUBX	Subgroup mean
UCLE	Upper control limit for EWMA
WEIGHT	Weight (r) assigned to most recent

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable of length 8.

For an example of a TABLE= data set, see "Saving Control Limit Parameters" on page 803.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, four methods are available for estimating the process standard deviation σ. Three methods (referred to as the default, MVLUE, and RMSDF) are available with subgrouped data. A fourth method is used if the data are individual measurements (see "Default Method for Individual Measurements" on page 833).

Default Method for Subgroup Samples

This method is the default for EWMA charts using subgrouped data. The default estimate of σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \ldots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method for Subgroup Samples

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \ldots + h_N s_N/c_4(n_N)}{h_1 + \ldots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method for Subgroup Samples

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ as follows:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method for Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as

$$\hat{\sigma} = \sqrt{\frac{1}{2(N-1)} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$$

where N is the number of observations, and $x_1, x_2, \ldots, x_N$ are the individual measurements. This formula is given by Wetherill (1977), who states that the estimate of the variance is biased if the measurements are autocorrelated.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	All	*Subgroup-variable*
Vertical	DATA=	*Process*
Vertical	HISTORY=	Subgroup mean variable
Vertical	TABLE=	_EWMA_

For example, the following sets of statements specify the label *EWMA of Clip Gaps* for the vertical axis and the label *Day* for the horizontal axis of the EWMA chart:

```
proc macontrol data=Clips1;
   ewmachart Gap*Day / weight=0.3;
   label Gap = 'EWMA of Clip Gaps';
   label Day = 'Day';
run;

proc macontrol history=Cliphist;
   ewmachart Gap*Day / weight=0.3;
   label Gapx = 'EWMA of Clip Gaps';
   label Day  = 'Day';
run;

proc macontrol table=Cliptab;
   ewmachart Gap*Day;
   label _EWMA_  = 'EWMA of Clip Gaps';
   label Day     = 'Day';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: EWMACHART Statement

This section provides advanced examples of the EWMACHART statement.

Example 10.1: Specifying Standard Values for the Process Mean and Process Standard Deviation

NOTE: See *Specifying Standard Values for EWMA Chart* in the SAS/QC Sample Library.

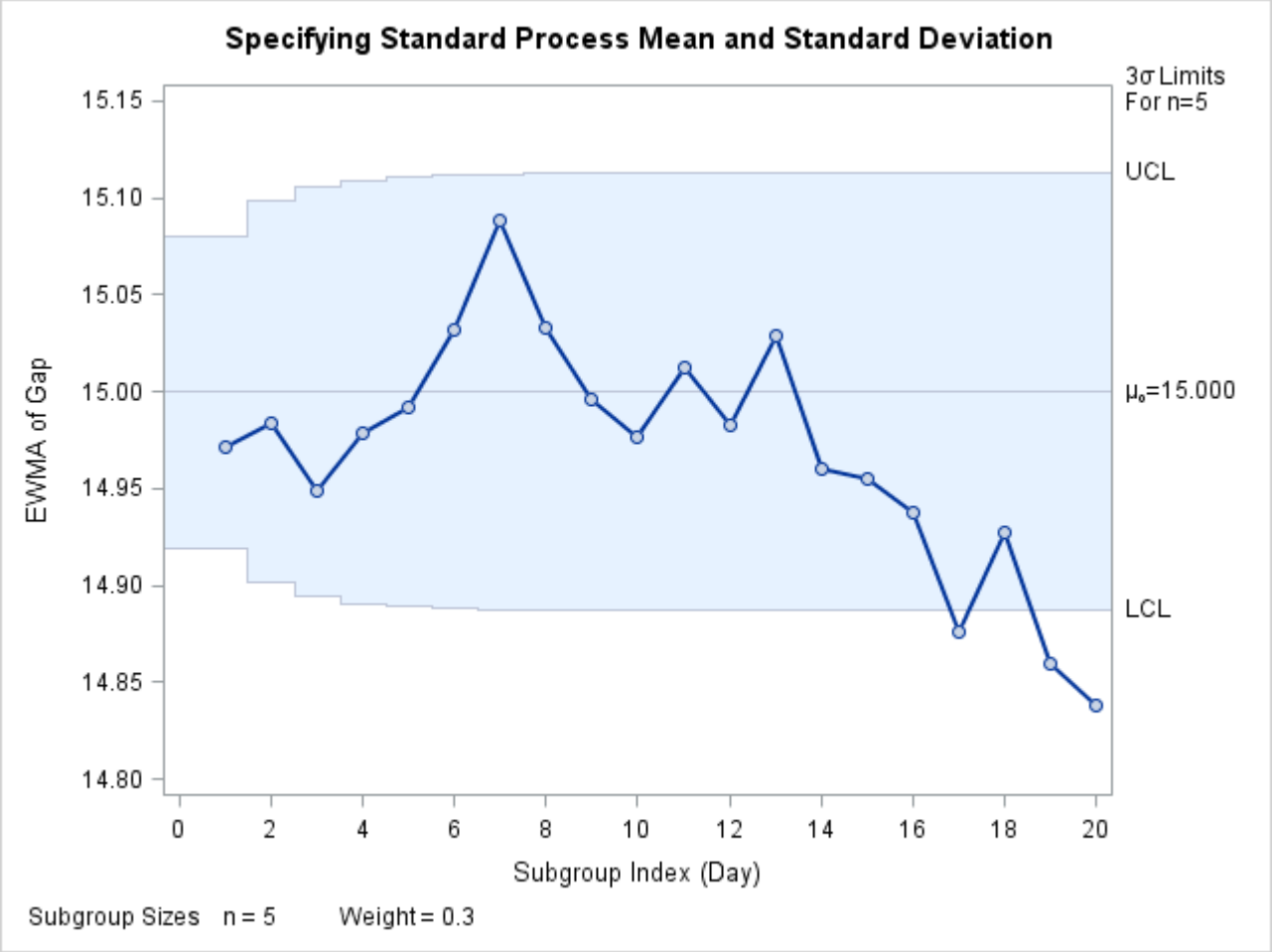
By default, the EWMACHART statement estimates the process mean (μ) and standard deviation (σ) from the data. This is illustrated in the "Getting Started" section of this chapter. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

For example, suppose it is known that the metal clip manufacturing process (introduced in "Creating EWMA Charts from Raw Data" on page 796) has a mean of 15 and standard deviation of 0.2. The following statements specify these standard values:

```
ods graphics on;
title 'Specifying Standard Process Mean and Standard Deviation';
proc macontrol data=Clips1;
   ewmachart Gap*Day /
      odstitle = title
      mu0      = 15
      sigma0   = 0.2
      weight   = 0.3
      xsymbol  = mu0
      markers;
run;
```

The XSYMBOL= option specifies the label for the central line. The resulting chart is shown in Output 10.1.1.

Output 10.1.1 Specifying Standard Values with MU0= and SIGMA0=



The central line and control limits are determined using μ_0 and σ_0 (see the equations in Table 10.5). Output 10.1.1 indicates that the process is out-of-control, since the moving averages for Day=17, Day=19, and Day=20 lie below the lower control limit.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data Cliplim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Gap';
   _subgrp_ = 'Day';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 15;
   _stddev_ = 0.2;
   _weight_ = 0.3;

proc macontrol data=Clips1 limits=Cliplim;
   ewmachart Gap*Day /
      odstitle = title
      xsymbol  = mu0
      markers;
run;
```

The variable _WEIGHT_ is required, and its value provides the weight parameter used to compute the EWMAs. The variables _VAR_ and _SUBGRP_ are also required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the EWMACHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting chart (not shown here) is identical to the one shown in Output 10.1.1.

Example 10.2: Displaying Limits Based on Asymptotic Values

NOTE: See *Displaying Limits Based on Asymptotic Values* in the SAS/QC Sample Library.

The upper (lower) control limits in Output 10.1.1 are monotonically increasing (decreasing). As the number of subgroups increases, the control limits approach the following asymptotic values:

$$\mathrm{LCL} = \overline{\overline{X}} - k\hat{\sigma}\sqrt{r/n(2-r)}$$

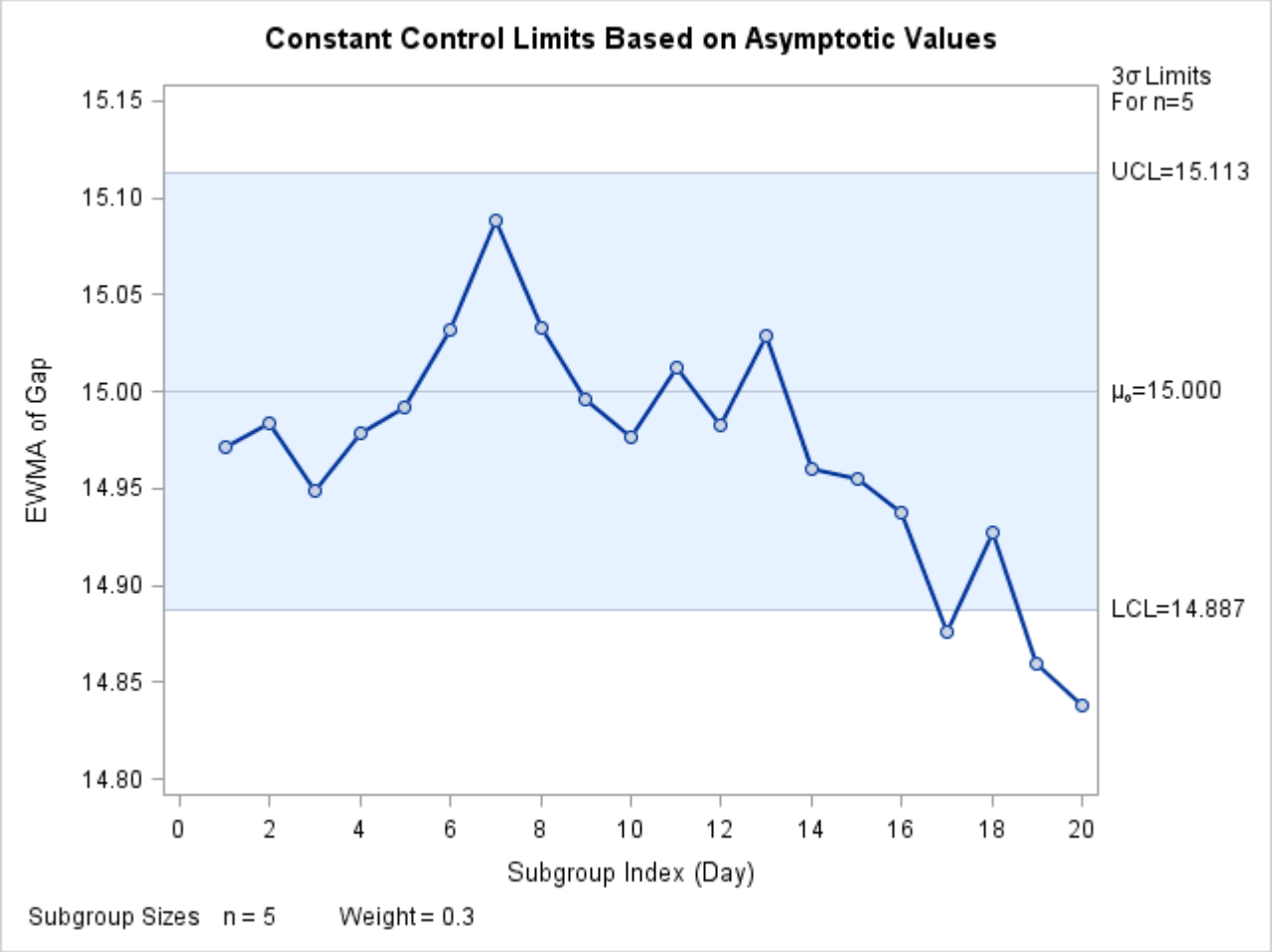
$$\mathrm{UCL} = \overline{\overline{X}} + k\hat{\sigma}\sqrt{r/n(2-r)}$$

These constant limits are displayed if you specify the ASYMPTOTIC option, as illustrated by the following statements:

```
ods graphics on;
title 'Constant Control Limits Based on Asymptotic Values';
proc macontrol data=Clips1;
   ewmachart Gap*Day /
      odstitle = title
      mu0      = 15
      sigma0   = 0.2
      weight   = 0.3
      xsymbol  = mu0
      asymptotic
      markers;
run;
```

The chart is shown in Output 10.2.1.

Output 10.2.1 Asymptotic Control Limits



Note that the same three points that were outside the exact limits (displayed in Output 10.1.1) fall outside the asymptotic limits. The exact limits quickly approach the asymptotic values, so only the first few subgroups have appreciably different limits.

Example 10.3: Working with Unequal Subgroup Sample Sizes

NOTE: See *EWMA Chart with Unequal Subgroup Sample Sizes* in the SAS/QC Sample Library.

This example contains measurements from the metal clip manufacturing process (introduced in "Creating EWMA Charts from Raw Data" on page 796). The following statements create a SAS data set named Clips4, which contains additional clip gap measurements taken on a daily basis:

```
data Clips4;
   input Day @;
   length Dayc $2.;
   informat Day ddmmyy8.;
   format   Day date5.;
   Dayc=put(Day,date5.);
   Dayc=substr(Dayc,1,2);
   do i=1 to 5;
      input Gap @;
      output;
   end;
   drop i;
   label Dayc='April';
   datalines;
 1/4/86  14.93  14.65  14.87  15.11  15.18
 2/4/86  15.06  14.95  14.91  15.14  15.41
 3/4/86  14.90  14.90  14.96  15.26  15.18
 4/4/86  15.25  14.57  15.33  15.38  14.89
 7/4/86  14.68  14.63  14.72  15.32  14.86
 8/4/86  14.48  14.88  14.98  14.74  15.48
 9/4/86  14.99  15.16  15.02  15.53  14.66
10/4/86  14.88  15.44  15.04  15.10  14.89
11/4/86  15.14  15.33  14.75  15.23  14.64
14/4/86  15.46  15.30  14.92  14.58  14.68
15/4/86  15.23  14.63    .      .      .
16/4/86  15.13  15.25    .      .      .
17/4/86  15.06  15.25  15.28  15.30  15.34
18/4/86  15.22  14.77  15.12  14.82  15.29
21/4/86  14.95  14.96  14.65  14.87  14.77
22/4/86  15.01  15.11  15.11  14.79  14.88
23/4/86  14.97  15.50  14.93  15.13  15.25
24/4/86  15.23  15.21  15.31  15.07  14.97
25/4/86  15.08  14.75  14.93  15.34  14.98
28/4/86  15.07  14.86  15.42  15.47  15.24
29/4/86  15.27  15.20  14.85  15.62  14.67
30/4/86  14.97  14.73  15.09  14.98  14.46
;
```

Note that only two gap measurements were recorded on April 15 and April 16.

A partial listing of Clips4 is shown in Output 10.3.1. This data set contains three variables: Day is a numeric variable that contains the date (month, day, and year) that the measurement is taken, Dayc is a character variable that contains the day the measurement is taken, and Gap is a numeric variable that contains the measurement.

Output 10.3.1 The Data Set Clips4

The Data Set Clips4

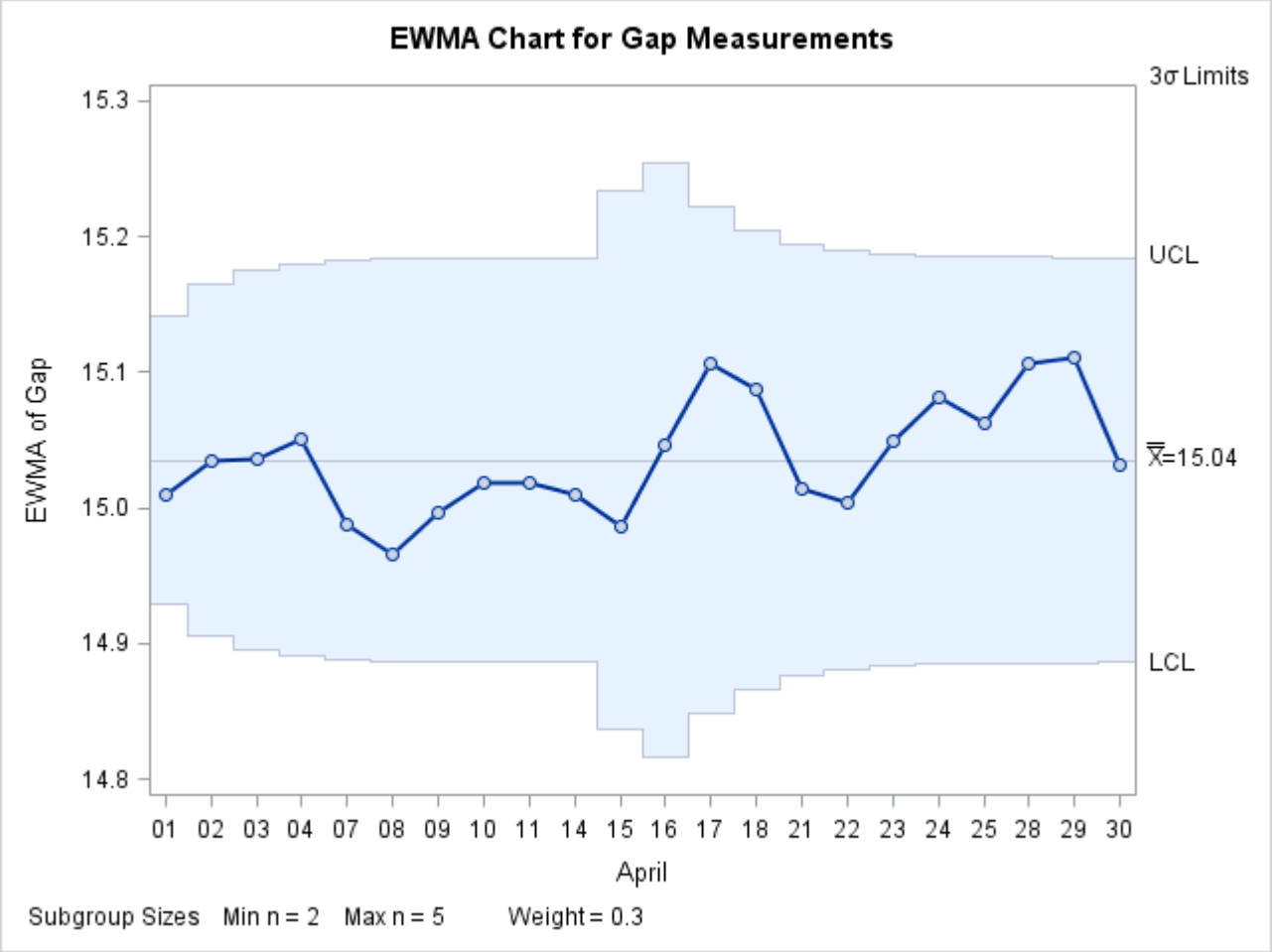
Day	Dayc	Gap
01APR	01	14.93
01APR	01	14.65
01APR	01	14.87
01APR	01	15.11
01APR	01	15.18
02APR	02	15.06
02APR	02	14.95
02APR	02	14.91
02APR	02	15.14
02APR	02	15.41
03APR	03	14.90
03APR	03	14.90
03APR	03	14.96
03APR	03	15.26
03APR	03	15.18

The following statements request an EWMA chart, shown in Output 10.3.2, for these gap measurements:

```
ods graphics on;
title 'EWMA Chart for Gap Measurements';
proc macontrol data=Clips4;
   ewmachart Gap*Dayc / odstitle = title
                        weight   = 0.3
                        markers;
run;
```

The character variable Dayc (rather than the numeric variable Day) is specified as the *subgroup-variable* in the preceding EWMACHART statement. If Day were the *subgroup-variable*, each day during April would appear on the horizontal axis, including the weekend days of April 5 and April 6 for which no measurements were taken. To avoid this problem, the *subgroup-variable* Dayc is created from Day using the PUT and SUBSTR function. Since Dayc is a character *subgroup-variable*, a discrete axis is used for the horizontal axis, and as a result, April 5 and April 6 do not appear on the horizontal axis in Output 10.3.2. A LABEL statement is used to specify the label *April* for the horizontal axis, indicating the month that these measurements were taken.

Output 10.3.2 EWMA Chart with Varying Sample Sizes



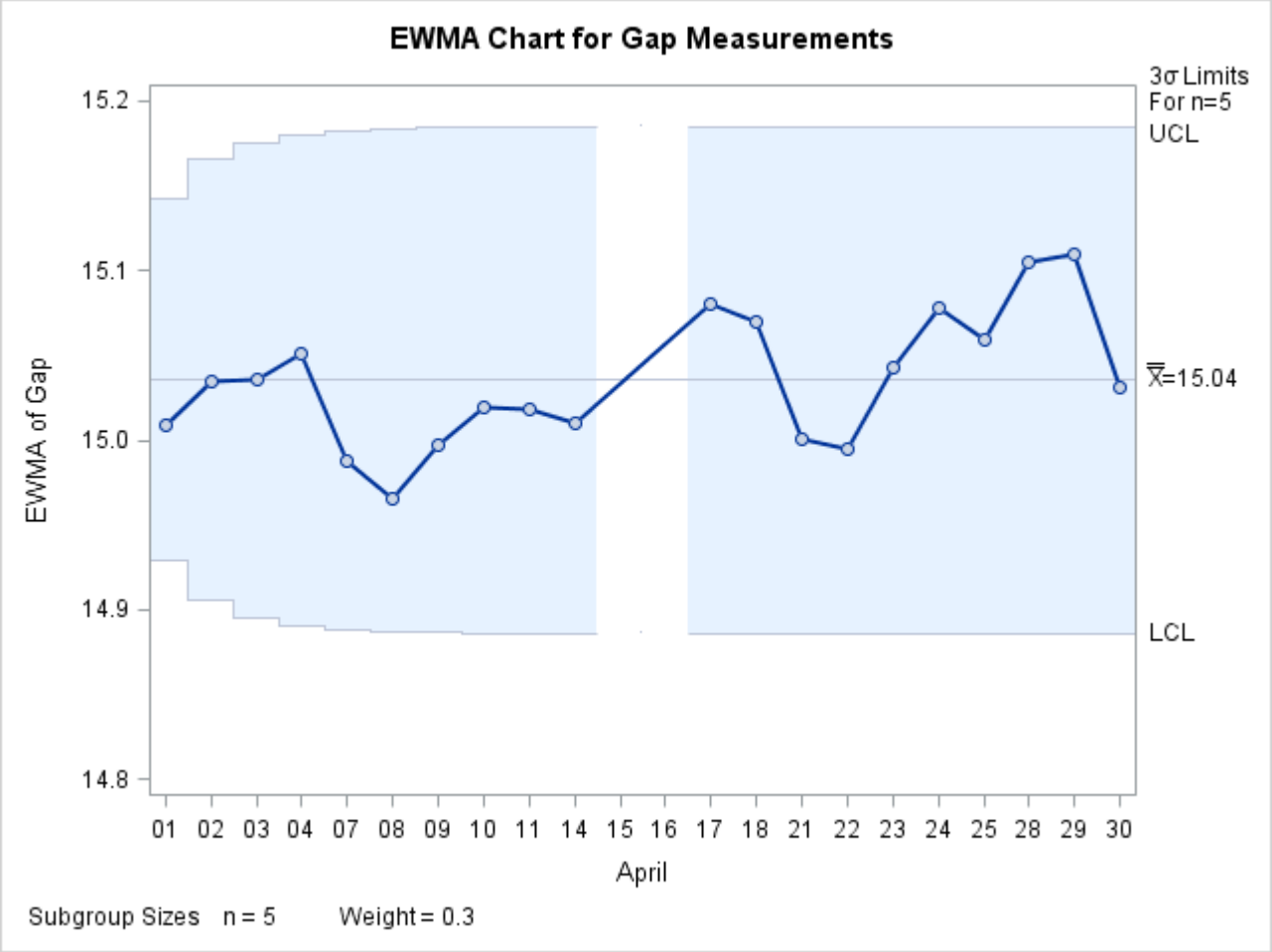
Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

The EWMACHART statement provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing control limits, as illustrated by the following statements:

```
title 'EWMA Chart for Gap Measurements';
proc macontrol data=Clips4;
   ewmachart Gap*Dayc / odstitle = title
                        weight   = 0.3
                        limitn   = 5
                        markers;
run;
```

The resulting chart is shown in Output 10.3.3.

Output 10.3.3 Control Limits Based on Fixed Sample Size

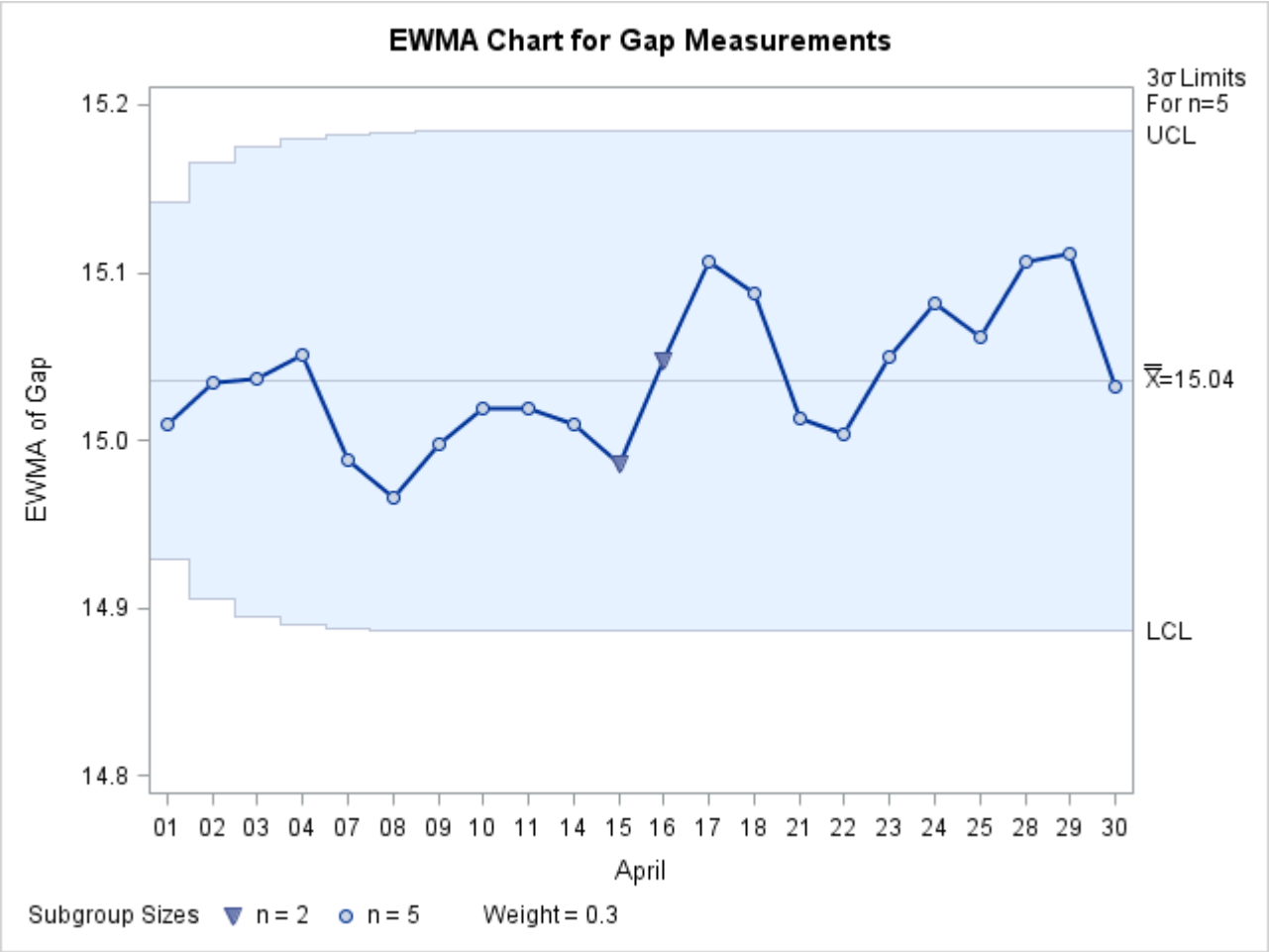


Note that the only points displayed are those corresponding to subgroups whose sample size matches the nominal sample size of five. Therefore, points are not displayed for April 15 and April 16. To plot points for all subgroups (regardless of subgroup sample size), you can specify the ALLN option, as follows:

```
title 'EWMA Chart for Gap Measurements';
proc macontrol data=Clips4;
   ewmachart Gap*Dayc/ odstitle = title
                       weight   = 0.3
                       limitn   = 5
                       alln
                       nmarkers;
run;
```

The chart is shown in Output 10.3.4. The NMARKERS option requests special symbols to identify points for which the subgroup sample size differs from the nominal sample size.

Output 10.3.4 Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes $\hat{\sigma}$ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests a minimum variance linear unbiased estimate (MVLUE), which assigns greater weight to estimates of σ from subgroups with larger sample sizes. Specifying SMETHOD=RMSDF requests a weighted root-mean-square estimate. If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the MVLUE. For more information, see "Methods for Estimating the Standard Deviation" on page 832.

The following statements apply all three methods:

```
proc macontrol data=Clips4;
   ewmachart Gap*Dayc / outlimits = Cliplim1
                        outindex  = 'Default'
                        weight    = 0.3
                        nochart;
   ewmachart Gap*Dayc / smethod   = mvlue
                        outlimits = Cliplim2
                        outindex  = 'MVLUE'
                        weight    = 0.3
                        nochart;
```

```
   ewmachart Gap*Dayc / smethod   = rmsdf
                        outlimits = Cliplim3
                        outindex  = 'RMSDF'
                        weight    = 0.3
                        nochart;
run;

data Climits;
   set Cliplim1 Cliplim2 Cliplim3;
run;
```

The data set Climits is listed in Output 10.3.5.

Output 10.3.5 Listing of the Data Set Climits

Estimating the Process Standard Deviation

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_MEAN_	_STDDEV_	_WEIGHT_
Gap	Dayc	Default	ESTIMATE	V	.002699796	3	15.0354	0.26503	0.3
Gap	Dayc	MVLUE	ESTIMATE	V	.002699796	3	15.0354	0.26096	0.3
Gap	Dayc	RMSDF	ESTIMATE	V	.002699796	3	15.0354	0.25959	0.3

Note that the estimate of the process standard deviation (stored in the variable _STDDEV_) is slightly different depending on the estimation method. The variable _LIMITN_ is assigned the special missing value *V* in the OUTLIMITS= data set, indicating that the subgroup sample sizes vary.

Example 10.4: Displaying Individual Measurements on an EWMA Chart

NOTE: See *EWMA Chart with Individual Measurements* in the SAS/QC Sample Library.

In the manufacture of automotive tires, the diameter of the steel belts inside the tire is measured. The following data set contains these measurements for 30 tires:

```
data Tires;
   input Sample Diameter @@;
   datalines;
 1  24.05    2  23.99    3  23.95
 4  23.93    5  23.97    6  24.02
 7  24.06    8  24.10    9  23.98
10  24.03   11  23.91   12  24.06
13  24.06   14  23.96   15  23.98
16  24.06   17  24.01   18  24.00
19  23.93   20  23.92   21  24.09
22  24.11   23  24.05   24  23.98
25  23.98   26  24.06   27  24.02
28  24.06   29  23.97   30  23.96
;
```

The following statements use the IRCHART statement in the SHEWHART procedure (see "IRCHART Statement: SHEWHART Procedure" on page 1518) to create a data set containing the control limits for individual measurements and moving range charts for Diameter:

```
proc shewhart data=Tires;
   irchart Diameter*Sample / nochart outlimits=Tlimits;
run;
```

A listing of the data set Tlimits is shown in Output 10.4.1.

Output 10.4.1 Listing of the Data Set Tlimits

Control Limits for Diameter Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_	_UCLI_
Diameter	Sample	ESTIMATE	2	.002699796	3	23.8571	24.0083	24.1596

LCLR	_R_	_UCLR_	_STDDEV_
0	0.056897	0.18585	0.050423

The upper and lower control limits for the diameter measurements are 24.1596 and 23.8571, respectively.

In this example, reference lines will be used to display the control limits for the individual measurements on the EWMA chart. The following DATA step reads these control limits from Tlimits and creates a data set named Vrefdata, which contains the reference line information:

```
data Vrefdata;
   set Tlimits;
   length _reflab_ $16.;
   keep _ref_ _reflab_;
   _ref_ = _lcli_; _reflab_= 'LCL for X'; output;
   _ref_ = _ucli_; _reflab_= 'UCL for X'; output;
run;
```

A listing of the data set Vrefdata is shown in Output 10.4.2.

Output 10.4.2 Listing of the Data Set Vrefdata

Reference Line Information

reflab	_ref_
LCL for X	23.8571
UCL for X	24.1596

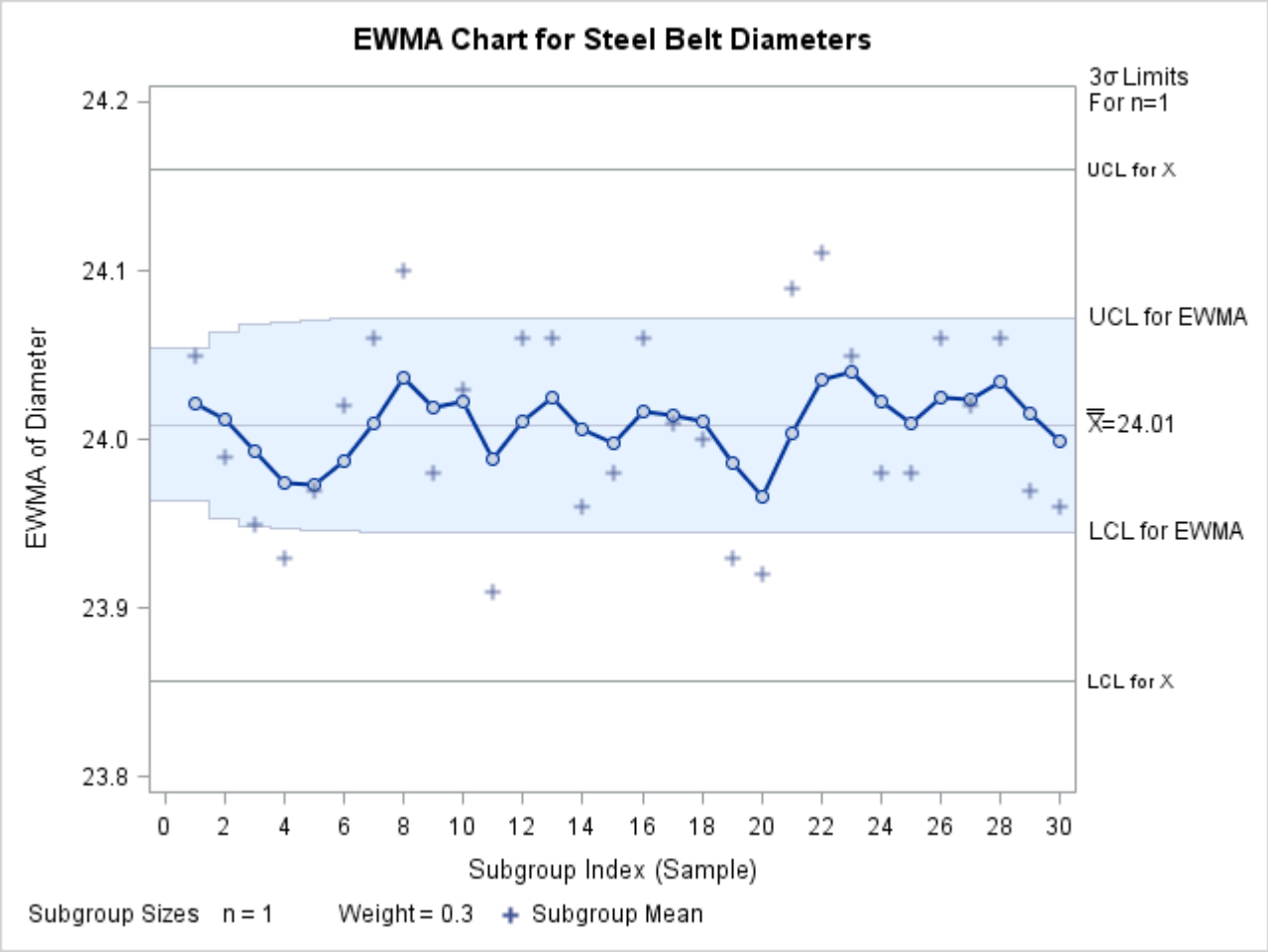
The following statements request an EWMA chart for these measurements:

```
ods graphics on;
title 'EWMA Chart for Steel Belt Diameters';
proc macontrol data=Tires;
   ewmachart Diameter*Sample / weight     = 0.3
                               meansymbol = square
                               lcllabel   = 'LCL for EWMA'
                               ucllabel   = 'UCL for EWMA'
                               vref       = Vrefdata
                               odstitle   = title
                               vreflabpos = 3
                               markers;
run;
```

The MEANSYMBOL= option displays the individual measurements on the EWMA chart. By default, these values are not displayed. For traditional graphics, the MEANSYMBOL= option specifies the symbol used to plot the individual measurements. For ODS Graphics, specifying a MEANSYMBOL= value causes the subgroup means to be plotted, but the symbol used is determined by the ODS style in effect. The VREF= option reads the reference line information from Vrefdata. The resulting chart is shown in Output 10.4.3.

Output 10.4.3 indicates that the process is in control. None of the diameter measurements (indicated by squares) exceed their control limits, and none of the EWMAs exceed their limits.

Output 10.4.3 Displaying Individual Measurements on EWMA Chart



Example 10.5: Computing Average Run Lengths

NOTE: See *Computing Average Run Lengths for EWMA Chart* in the SAS/QC Sample Library.

The EWMAARL DATA step function computes the average run length for an exponentially weighted moving average (EWMA) scheme (refer to Crowder 1987a,b for details). You can use this function to design a scheme by first calculating average run lengths for a range of values for the weight and then choosing the weight that yields a desired average run length.

The following statements compute the average run lengths for shifts between 0.5 and 2 and weights between 0.25 and 1. The data set ARLs is displayed in Output 10.5.1.

```
data ARLs;
   do shift=.5 to 2 by .5;
      do Weight=.25 to 1 by .25;
      arl=ewmaarl(shift,Weight,3.0);
      output;
      end;
   end;
run;
```

Output 10.5.1 Listing of the Data Set ARLs

Average Run Lengths for Various Shifts and Weights

shift=0.5

Weight	arl
0.25	48.453
0.50	75.354
0.75	110.950
1.00	155.224

shift=1

Weight	arl
0.25	11.1543
0.50	15.7378
0.75	25.6391
1.00	43.8947

shift=1.5

Weight	arl
0.25	5.4697
0.50	6.1111
0.75	8.7201
1.00	14.9677

shift=2

Weight	arl
0.25	3.61677
0.50	3.46850
0.75	4.15346
1.00	6.30296

Note that when the weight is 1.0, the EWMAARL function returns the average run length for a Shewhart chart for means. For more details, see "EWMAARL Function" on page 2228.

In addition to using the EWMAARL function to design a EWMA scheme with desired average run length properties, you can use it to evaluate an existing scheme. For example, suppose you have an EWMA chart with 3σ control limits using a weight parameter of 0.3. The following DATA step computes the average run lengths for various shifts using this scheme:

```
data ARLinfo;
   do shift=0 to 2 by .25;
      arl = ewmaarl(shift,0.3,3.0);
      output;
   end;
run;
```

The data set ARLinfo is displayed in Output 10.5.2.

Output 10.5.2 Listing of the Data Set ARLinfo

Average Run Lengths for EWMA Scheme (k=3 and r=0.3)

shift	arl
0.00	465.553
0.25	178.741
0.50	53.160
0.75	21.826
1.00	11.699
1.25	7.525
1.50	5.447
1.75	4.258
2.00	3.506

MACHART Statement: MACONTROL Procedure

Overview: MACHART Statement

The MACHART statement creates a uniformly weighted moving average control chart (commonly referred to as a moving average control chart), which is used to decide whether a process is in a state of statistical control and to detect shifts in the process average.

You can use options in the MACHART statement to

- specify the span of the moving averages (the number of terms in the moving average)
- compute control limits from the data based on a multiple of the standard error of the plotted moving averages or as probability limits
- tabulate the moving averages, subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limit parameters in an output data set
- save the moving averages, subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read control limit parameters from an input data set

- specify one of several methods for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display a secondary chart that plots a time trend that has been removed from the data
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing moving average control charts with the MACHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: MACHART Statement

This section introduces the MACHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the MACHART statement is presented in the section "Syntax: MACHART Statement" on page 861, and advanced examples are given in the section "Examples: MACHART Statement" on page 889.

Creating Moving Average Charts from Raw Data

NOTE: See *Uniformly Weighted Moving Average Chart* in the SAS/QC Sample Library.

In the manufacture of a metal clip, the gap between the ends of the clip is a critical dimension. To monitor the process for a change in the average gap, subgroup samples of five clips are selected daily. The data are analyzed with a uniformly weighted moving average chart. The gaps recorded during the first twenty days are saved in a SAS data set named Clips1.

```
data Clips1;
   input Day @ ;
   do i=1 to 5;
      input Gap @ ;
      output;
   end;
   drop i;
   datalines;
 1  14.76  14.82  14.88  14.83  15.23
 2  14.95  14.91  15.09  14.99  15.13
 3  14.50  15.05  15.09  14.72  14.97
 4  14.91  14.87  15.46  15.01  14.99
 5  14.73  15.36  14.87  14.91  15.25
 6  15.09  15.19  15.07  15.30  14.98
 7  15.34  15.39  14.82  15.32  15.23
 8  14.80  14.94  15.15  14.69  14.93
 9  14.67  15.08  14.88  15.14  14.78
10  15.27  14.61  15.00  14.84  14.94
11  15.34  14.84  15.32  14.81  15.17
12  14.84  15.00  15.13  14.68  14.91
13  15.40  15.03  15.05  15.03  15.18
14  14.50  14.77  15.22  14.70  14.80
15  14.81  15.01  14.65  15.13  15.12
16  14.82  15.01  14.82  14.83  15.00
17  14.89  14.90  14.60  14.40  14.88
18  14.90  15.29  15.14  15.20  14.70
19  14.77  14.60  14.45  14.78  14.91
20  14.80  14.58  14.69  15.02  14.85
;
```

The following statements produce the listing of the data set Clips1 shown in Figure 10.10:

```
title 'The Data Set Clips1';
proc print data=Clips1(obs=15) noobs;
run;
```

Figure 10.10 Partial Listing of the Data Set Clips1

The Data Set Clips1

Day	Gap
1	14.76
1	14.82
1	14.88
1	14.83
1	15.23
2	14.95
2	14.91
2	15.09
2	14.99
2	15.13
3	14.50
3	15.05
3	15.09
3	14.72
3	14.97

The data set Clips1 is said to be in "strung-out" form, since each observation contains the day and gap measurement of a single clip. The first five observations contain the gap measurements for the first day, the second five observations contain the gap measurements for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Gap contains the gap measurements and is referred to as the *process variable* (or *process* for short).

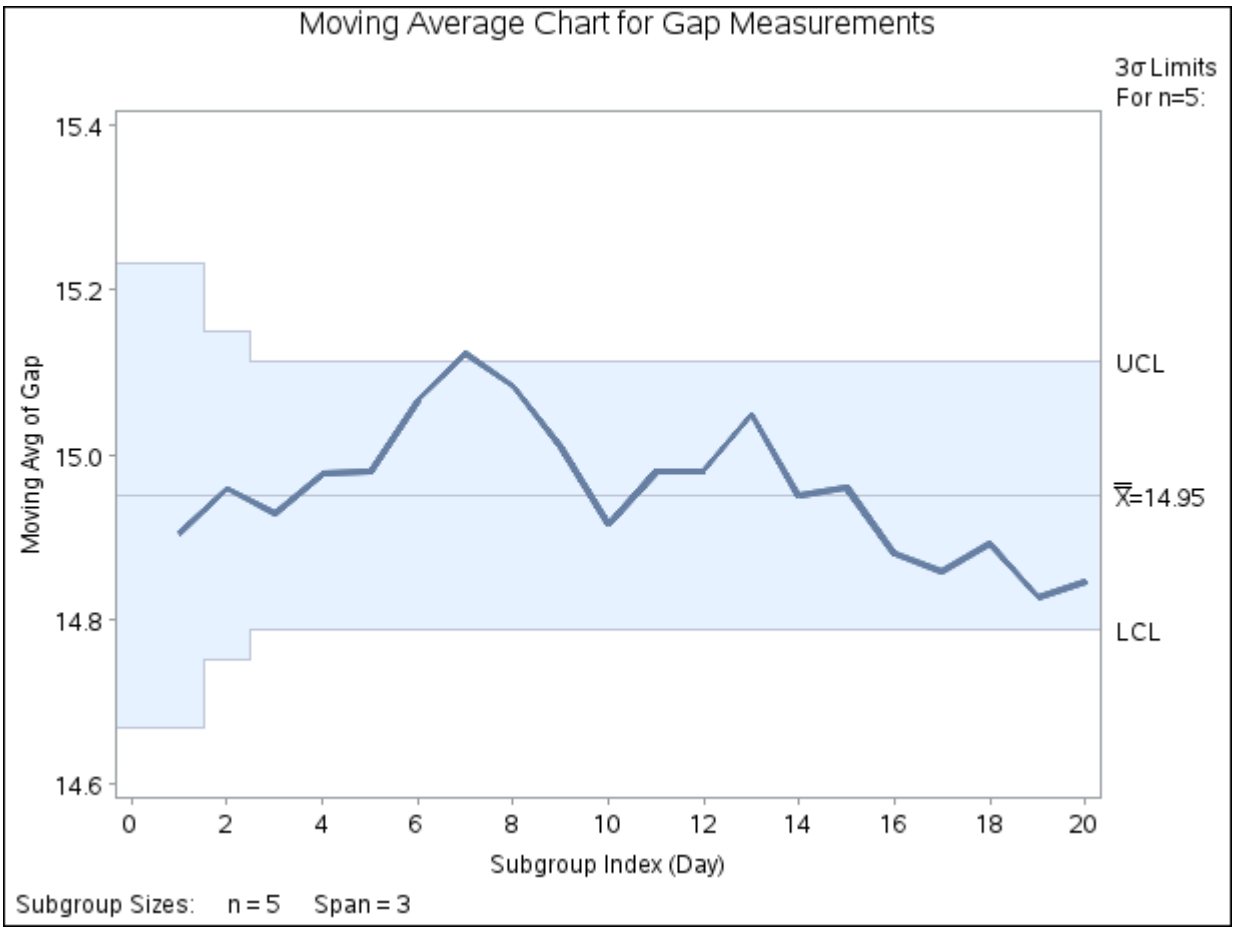
The within-subgroup variability of the gap measurements is known to be stable. You can use a uniformly weighted moving average chart to determine whether the mean level is in control. The following statements create the chart shown in Figure 10.11:

```
ods graphics off;
title 'Moving Average Chart for Gap Measurements';
proc macontrol data=Clips1;
   machart Gap*Day / span=3;
run;
```

This example illustrates the basic form of the MACHART statement. After the keyword MACHART, you specify the *process* to analyze (in this case, Gap) followed by an asterisk and the *subgroup-variable* (Day). The SPAN= option specifies the number of terms to include in the moving average. Options such as SPAN= are specified after the slash (/) in the MACHART statement. A complete list of options is presented in the section "Syntax: MACHART Statement" on page 861. You must provide the span of the moving average. As an alternative to specifying the SPAN= option, you can read the span from an input data set; see "Reading Preestablished Control Limit Parameters" on page 859.

The input data set is specified with the DATA= option in the PROC MACONTROL statement.

Figure 10.11 Uniformly Weighted Moving Average Chart for Gap Data



Each point on the chart represents the uniformly weighted moving average for a particular day. The moving average A_1 plotted at Day=1 is simply the subgroup mean for Day=1. The moving average A_2 plotted at Day=2 is the average of the subgroup means for Day=1 and Day=2. The moving average A_3 plotted at Day=3 is the average of the subgroup means for Day=1, Day=2, and Day=3.

$$
\begin{aligned}
A_1 &= \frac{14.76 + 14.82 + 14.88 + 14.83 + 15.23}{5} = 14.904 \text{ mm} \\
A_2 &= \frac{14.904 + 15.014}{2} = 14.959 \text{ mm} \\
A_3 &= \frac{14.904 + 15.014 + 14.866}{3} = 14.928 \text{ mm}
\end{aligned}
$$

For succeeding days, the moving average is similarly calculated as the average of the present and the two previous subgroup means (since a span of three is specified with the SPAN= option).

Note that the moving average for the seventh day lies above the upper control limit, signaling an out-of-control process.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 10.15.

For computational details, see "Constructing Uniformly Weighted Moving Average Charts" on page 874. For more details on reading from a DATA= data set, see "DATA= Data Set" on page 883.

Creating Moving Average Charts from Subgroup Summary Data

NOTE: See *Uniformly Weighted Moving Average Chart* in the SAS/QC Sample Library.

The previous example illustrates how you can create moving average charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the MACHART statement with data of this type. The following data set (Clipsum) provides the data from the preceding example in summarized form:

```
data Clipsum;
   input Day GapX GapS;
   GapN=5;
   datalines;
 1  14.904  0.18716
 2  15.014  0.09317
 3  14.866  0.25006
 4  15.048  0.23732
 5  15.024  0.26792
 6  15.126  0.12260
 7  15.220  0.23098
 8  14.902  0.17254
 9  14.910  0.19824
10  14.932  0.24035
11  15.096  0.25618
12  14.912  0.16903
13  15.138  0.15928
14  14.798  0.26329
15  14.944  0.20876
16  14.896  0.09965
17  14.734  0.22512
18  15.046  0.24141
19  14.702  0.17880
20  14.788  0.16634
;
```

A partial listing of Clipsum is shown in Figure 10.12. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day). The variable GapX contains the subgroup means, the variable GapS contains the subgroup standard deviations, and the variable GapN contains the subgroup sample sizes (these are all five).

Figure 10.12 The Summary Data Set Clipsum

The Data Set Clipsum

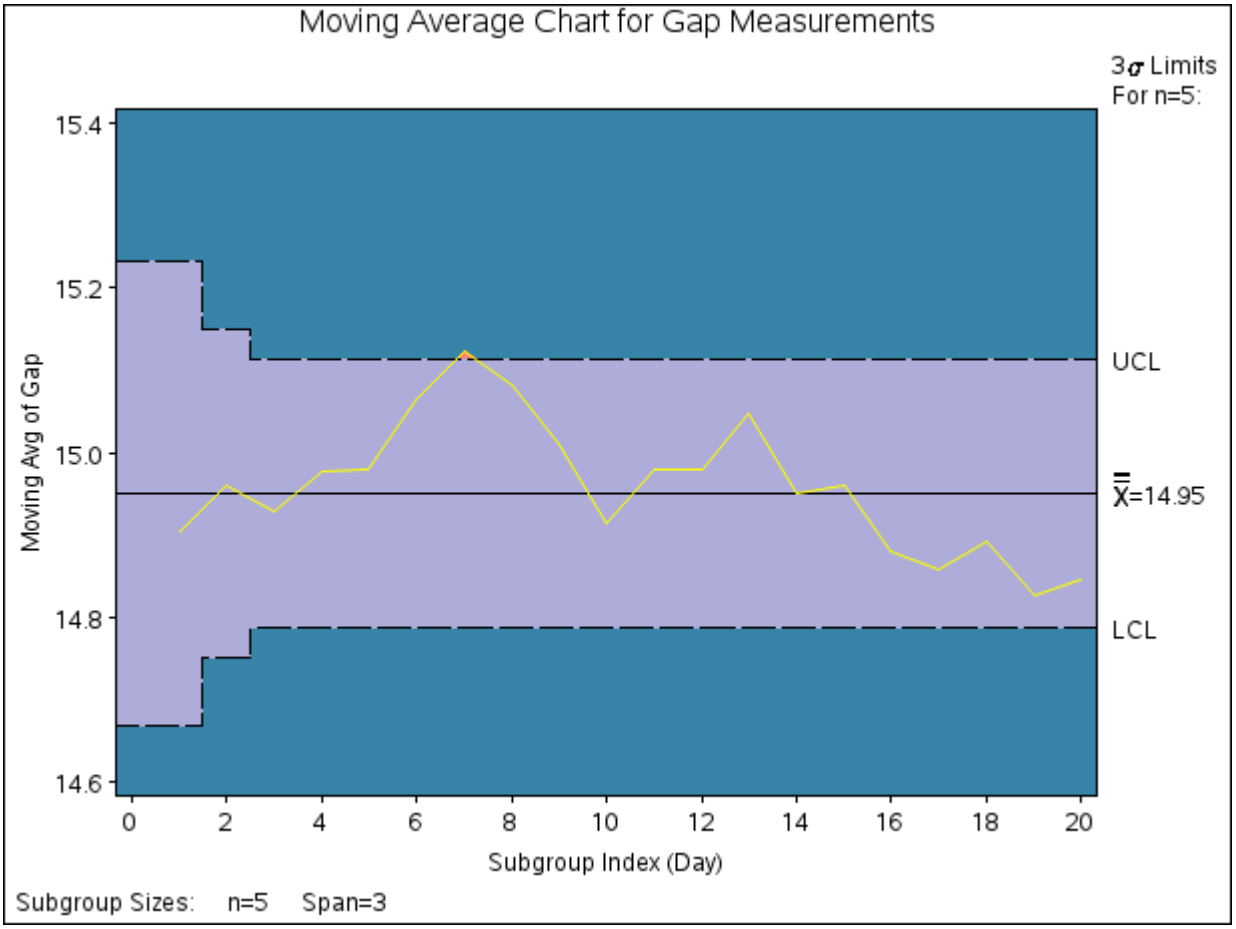
Day	GapX	GapS	GapN
1	14.904	0.18716	5
2	15.014	0.09317	5
3	14.866	0.25006	5
4	15.048	0.23732	5
5	15.024	0.26792	5

You can read this data set by specifying it as a HISTORY= data set in the PROC MACONTROL statement, as follows:

```
options nogstyle;
symbol color=salmon h=0.8;
title 'Moving Average Chart for Gap Measurements';
proc macontrol history=Clipsum;
   machart Gap*Day / span     = 3
                     cframe   = steel
                     cinfill  = vpab
                     cconnect = yellow
                     coutfill = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and MACHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting moving average chart is shown in Figure 10.13.

Figure 10.13 Uniformly Weighted Moving Average Chart from Summary Data



Note that Gap is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables GapX, GapS, and GapN. The suffix characters *X*, *S*, and *N* indicate *mean*,

standard deviation, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (Gap), which is referred to as the *process*. The variables GapX, GapS, and GapN are all required. The name Day specified after the asterisk is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the MACHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the MACHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC MACONTROL statement to rename the variables for the duration of the MACONTROL procedure step (see "" on page 1887 for an example).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 885.

Saving Summary Statistics

NOTE: See *Uniformly Weighted Moving Average Chart* in the SAS/QC Sample Library.

In this example, the MACHART statement is used to create a summary data set that can be read later by the MACONTROL procedure (as in the preceding example). The following statements read measurements from the data set Clips1 and create a summary data set named Cliphist:

```
title 'Summary Data Set for Gap Measurements';
proc macontrol data=Clips1;
   machart Gap*Day / span       = 3
                     outhistory = Cliphist
                     nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 10.11.

Figure 10.14 contains a partial listing of Cliphist.

Figure 10.14 The Summary Data Set Cliphist

Summary Data Set for Gap Measurements

Day	GapX	GapS	GapA	GapN
1	14.904	0.18716	14.9040	5
2	15.014	0.09317	14.9590	5
3	14.866	0.25006	14.9280	5
4	15.048	0.23732	14.9760	5
5	15.024	0.26792	14.9793	5

There are five variables in the data set Cliphist.

- Day contains the subgroup index.
- GapX contains the subgroup means.
- GapS contains the subgroup standard deviations.
- GapA contains the subgroup moving averages.
- GapN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, *A*, and *N* to the *process* Gap specified in the MACHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 880.

Saving Control Limit Parameters

NOTE: See *Uniformly Weighted Moving Average Chart* in the SAS/QC Sample Library.

You can save the control limit parameters used for a moving average chart in a SAS data set; this enables you to use these parameters with future data (see "Reading Preestablished Control Limit Parameters" on page 859) or modify the parameters with a DATA step program.

The following statements read measurements from the data set Clips1 (see "Creating Moving Average Charts from Raw Data" on page 849) and save the control limit parameters in a data set named Cliplim:

```
title 'Control Limit Parameters';
proc macontrol data=Clips1;
   machart Gap*Day / span      = 3
                     outlimits = Cliplim
                     nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Cliplim is listed in Figure 10.15.

Figure 10.15 The Data Set Cliplim Containing Control Limit Information

Control Limit Parameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_MEAN_	_STDDEV_	_SPAN_
Gap	Day	ESTIMATE	5	.002699796	3	14.95	0.21108	3

Note that the data set Cliplim does not contain the actual control limits, but rather the parameters required to compute the limits.

The data set contains one observation with the parameters for *process* Gap. The variable _SPAN_ contains the number of terms used to calculate the moving average. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates that the values of _MEAN_ and _STDDEV_ are estimates rather than standard values. For more information, see "OUTLIMITS= Data Set" on page 879.

You can create an output data set containing the control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limits';
proc macontrol data=Clips1;
   machart Gap*Day / span     = 3
                     outtable = Cliptab
                     nochart;
run;
```

The data set Cliptab is listed in Figure 10.16.

This data set contains one observation for each subgroup sample. The variable _UWMA_ contains the uniformly weighted moving average. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes, respectively. The variables _LCLA_ and _UCLA_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Day contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 881.

Figure 10.16 The OUTTABLE= Data Set Cliptab

Summary Statistics and Control Limits

VAR	Day	_SIGMAS_	_LIMITN_	_SPAN_	_SUBN_	_SUBX_	_SUBS_	_LCLA_	_UWMA_
Gap	1	3	5	3	5	14.904	0.18716	14.6668	14.9040
Gap	2	3	5	3	5	15.014	0.09317	14.7498	14.9590
Gap	3	3	5	3	5	14.866	0.25006	14.7865	14.9280
Gap	4	3	5	3	5	15.048	0.23732	14.7865	14.9760
Gap	5	3	5	3	5	15.024	0.26792	14.7865	14.9793
Gap	6	3	5	3	5	15.126	0.12260	14.7865	15.0660
Gap	7	3	5	3	5	15.220	0.23098	14.7865	15.1233
Gap	8	3	5	3	5	14.902	0.17254	14.7865	15.0827
Gap	9	3	5	3	5	14.910	0.19824	14.7865	15.0107
Gap	10	3	5	3	5	14.932	0.24035	14.7865	14.9147
Gap	11	3	5	3	5	15.096	0.25618	14.7865	14.9793
Gap	12	3	5	3	5	14.912	0.16903	14.7865	14.9800
Gap	13	3	5	3	5	15.138	0.15928	14.7865	15.0487
Gap	14	3	5	3	5	14.798	0.26329	14.7865	14.9493
Gap	15	3	5	3	5	14.944	0.20876	14.7865	14.9600
Gap	16	3	5	3	5	14.896	0.09965	14.7865	14.8793
Gap	17	3	5	3	5	14.734	0.22512	14.7865	14.8580
Gap	18	3	5	3	5	15.046	0.24141	14.7865	14.8920
Gap	19	3	5	3	5	14.702	0.17880	14.7865	14.8273
Gap	20	3	5	3	5	14.788	0.16634	14.7865	14.8453

MEAN	_UCLA_	_STDDEV_	_EXLIM_
14.95	15.2332	0.21108	
14.95	15.1502	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	UPPER
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	
14.95	15.1135	0.21108	

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Cliptab and display a moving average chart (not shown here) identical to Figure 10.11:

```
title 'Moving Average Chart for Gap Measurements';
proc macontrol table=Cliptab;
   machart Gap*Day;
run;
```

For more information, see "TABLE= Data Set" on page 886.

Reading Preestablished Control Limit Parameters

NOTE: See *Uniformly Weighted Moving Average Chart* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set saved the control limit parameters in the data set Cliplim. This example shows how to apply these parameters to new data provided in the following data set:

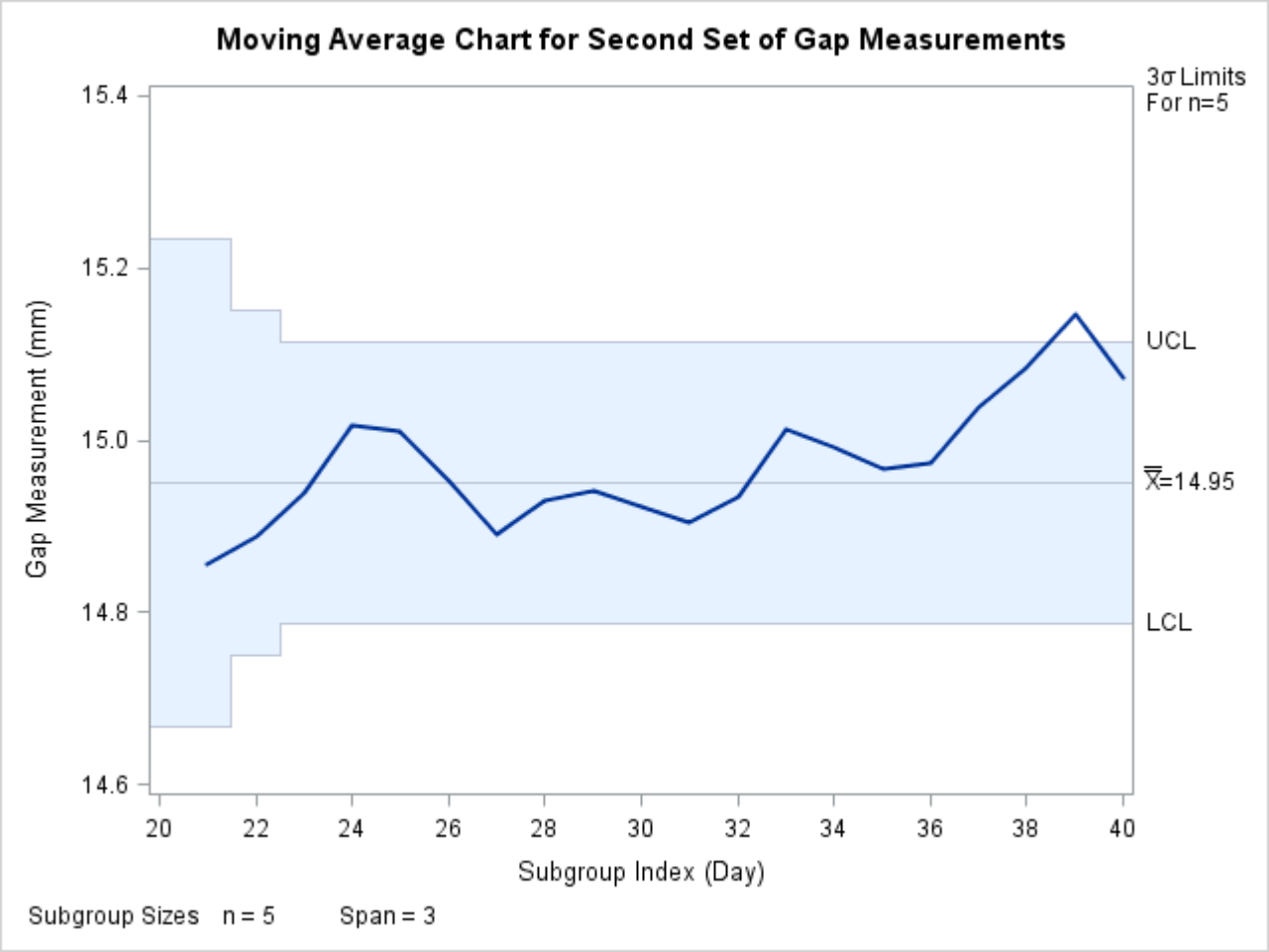
```
data Clips1a;
   label Gap='Gap Measurement (mm)';
   input Day @;
   do i=1 to 5;
      input Gap @;
      output;
   end;
   drop i;
   datalines;
21  14.86 15.01 14.67 14.67 15.07
22  14.93 14.53 15.07 15.10 14.98
23  15.27 14.90 15.12 15.10 14.80
24  15.02 15.21 14.93 15.11 15.20
25  14.90 14.81 15.26 14.57 14.94
26  14.78 15.29 15.13 14.62 14.54
27  14.78 15.15 14.61 14.92 15.07
28  14.92 15.31 14.82 14.74 15.26
29  15.11 15.04 14.61 15.09 14.68
30  15.00 15.04 14.36 15.20 14.65
31  14.99 14.76 15.18 15.04 14.82
32  14.90 14.78 15.19 15.06 15.06
33  14.95 15.10 14.86 15.27 15.22
34  15.03 14.71 14.75 14.99 15.02
35  15.38 14.94 14.68 14.77 14.83
36  14.95 15.43 14.87 14.90 15.34
37  15.18 14.94 15.32 14.74 15.29
38  14.91 15.15 15.06 14.78 15.42
39  15.34 15.34 15.41 15.36 14.96
40  15.12 14.75 15.05 14.70 14.74
;
```

The following statements create a moving average chart for the data in Clips1a using the control limit parameters in Cliplim:

```
ods graphics on;
title 'Moving Average Chart for Second Set of Gap Measurements';
proc macontrol data=Clips1a limits=Cliplim;
   machart Gap*Day / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC MACONTROL statement enables ODS Graphics, so the moving average chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 10.17.

Figure 10.17 Using Control Limit Parameters from a LIMITS= Data Set



The LIMITS= option in the PROC MACONTROL statement specifies the data set containing the control limits parameters. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Gap
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

Note that the moving average plotted for the 39th day lies above the upper control limit, signalling an out-of-control process.

In this example, the LIMITS= data set was created in a previous run of the MACONTROL procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 884 for details concerning the variables that you must provide, and see Example 10.6 for an illustration.

Syntax: MACHART Statement

The basic syntax for the MACHART statement is as follows:

MACHART *process* ∗ *subgroup-variable* / **SPAN=***value* < *options* > **;**

The general form of this syntax is as follows:

MACHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / **SPAN=***value* < *options* > **;**

Note that the SPAN= option is required unless its *value* is read from a LIMITS= data set. You can use any number of MACHART statements in the MACONTROL procedure. The components of the MACHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC MACONTROL statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Moving Average Charts from Raw Data" on page 849.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Moving Average Charts from Subgroup Summary Data" on page 853.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limit Parameters" on page 856.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct moving average charts (each with a span of 3) for Weight, Length, and Width:

```
proc macontrol data=Measures;
   machart (Weight Length Width)*Day / span=3;
run;
```

subgroup-variable

is the variable that classifies the data into subgroups. The *subgroup-variable* is required. In the preceding MACHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or plotting character used to plot the moving averages.

- If you produce a line printer chart, an 'A' is displayed for points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a moving average chart using an asterisk (*) to plot the points:

```
proc macontrol data=Values lineprinter;
   machart Weight*Hour='*' / span=3;
run;
```

options

specify chart parameters, enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" on page 862, which follows, lists all options by function.

Summary of Options

The following tables list the MACHART statement options by function. Options unique to the MACONTROL procedure are listed in Table 10.12, and are described in detail in the section "Dictionary of Special Options" on page 871. Options that are common to both the MACONTROL and SHEWHART procedures are listed in Table 10.13. They are described in detail in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 10.12 MACHART Statement Special Options

Option	Description
Options for Specifying Uniformly Weighted Moving Average Charts	
ALPHA=	Requests probability limits for control charts
ASYMPTOTIC	Requests constant control limits
LIMITN=	Specifies either a fixed nominal sample size (n) for control limits or allows the control limits to vary with subgroup sample size
MU0=	Specifies a standard (known) value μ_0 for the process mean
NOREADLIMITS	Specifies that control limit parameters are not to be read from LIMITS= data set
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from LIMITS= data set when both variables are available
READINDEX=	Reads control limit parameters from the first observation in the LIMITS= data set where the variable _INDEX_ equals *value*

Table 10.12 *continued*

Option	Description
READLIMITS	Reads control limit parameters from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMA0=	Specifies standard (known) value σ_0 for process standard deviation
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted moving averages
SPAN=	Specifies the number of terms in the moving average
Options for Plotting Subgroup Means	
CMEANSYMBOL=	Specifies color for MEANSYMBOL= symbol
MEANCHAR=	Specifies *character* to plot subgroup means on line printer charts
MEANSYMBOL=	Specifies symbol to plot subgroup means in traditional graphics

Table 10.13 MACHART Statement General Options

Option	Description
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line

Table 10.13 *continued*

Option	Description
Process Mean and Standard Deviation Options	
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on moving average chart
ALLLABEL2=	Labels every point on trend chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
NOTRENDCONNECT	Suppresses line segments that connect points on trend chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text

Table 10.13 *continued*

Option	Description
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of moving average chart
VAXIS2=	Specifies major tick mark values for vertical axis of trend chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	Specifies the natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable

Table 10.13 *continued*

Option	Description
MAXPANELS=	Specifies the maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
TRENDVAR=	Specifies list of trend variables
YPCT1=	Specifies length of vertical axis on moving average chart as a percentage of sum of lengths of vertical axes for moving average and trend charts
ZEROSTD	Displays moving average chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on moving average chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on moving average chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on moving average chart

Table 10.13 *continued*

Option	Description
VREF2=	Specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to moving average chart
ANNOTATE2=	Specifies annotate data set that adds features to trend chart
DESCRIPTION=	Specifies description of moving average chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of moving average chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
WTREND=	Specifies width of line segments connecting points on trend chart

Table 10.13 *continued*

Option	Description
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	Draws portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart

Table 10.13 *continued*

Option	Description
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	Creates all the tables that are produced by the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS options
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend

Table 10.13 *continued*

Option	Description
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	Delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	Superimposes star at each point on moving average chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups

Table 10.13 *continued*

Option	Description
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
VREFCHAR=	Specifies line character for VREF= and VREF2= lines

Dictionary of Special Options

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a single moving average exceeds its control limits. The value of α can range between 0 and 1. This assumes that the process is in statistical control and that the data follow a normal distribution. For the equations used to compute probability limits, see "Control Limits" on page 874.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option.
- As an alternative to specifying ALPHA=α (or reading _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

ASYMPTOTIC

requests constant upper and lower control limits for all subgroups having the following values:

$$\begin{aligned} \text{LCL} &= \overline{\overline{X}} - \frac{k\hat{\sigma}}{\sqrt{nw}} \\ \text{UCL} &= \overline{\overline{X}} + \frac{k\hat{\sigma}}{\sqrt{nw}} \end{aligned}$$

Here w is the span of the moving average, and n is the nominal sample size associated with the control limits. Substitute $\Phi^{-1}(1-\alpha/2)$ for k if you specify probability limits with the ALPHA= option. When you do not specify the ASYMPTOTIC option, the control limits are computed using the exact formulas

in Table 10.15. Use the ASYMPTOTIC option only if all the subgroup sample sizes are the same or if you specify LIMITN=*n*.

CMEANSYMBOL=*color*
specifies the *color* used for the symbol requested with the MEANSYMBOL= option in traditional graphics. This option is ignored unless you are producing traditional graphics.

LIMITN=*n*
LIMITN=VARYING
specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=*n*, moving averages are calculated and displayed only for those subgroups with a sample size equal to *n*, unless you also specify the ALLN option, which causes all the moving averages to be calculated and displayed. By default (or if you specify LIMITN=VARYING), moving averages are calculated and displayed for all subgroups, regardless of sample size.

MEANCHAR='*character***'**
specifies a *character* used in legacy line printer charts to plot the subgroup mean for each subgroup. By default, subgroup means are not plotted. This option is ignored unless you specify the LINEPRINTER option in the PROC MACONTROL statement.

MEANSYMBOL=*keyword*
specifies a symbol used to plot the subgroup mean for each subgroup in traditional graphics. By default, subgroup means are not plotted. This option is ignored unless you are producing traditional graphics.

MU0=*value*
specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data.

NOTE: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set.

See Example 10.6.

NOREADLIMITS
specifies that control limit parameters for each *process* listed in the MACHART statement are *not* to be read from the LIMITS= data set specified in the PROC MACONTROL statement.

The following example illustrates the NOREADLIMITS option:

```
proc macontrol data=Pistons limits=Diamlim;
   machart Diameter*Hour;
   machart Diameter*Hour / noreadlimits span=3;
run;
```

The first MACHART statement reads the control limits from the first observation in the data set Diamlim for which the variable _VAR_ is equal to 'Diameter' and the variable _SUBGRP_ is equal to 'Hour'. The second MACHART statement computes estimates of the process mean and standard deviation for the control limits from the measurements in the data set Pistons. Note that the second MACHART statement is equivalent to the following statements, which would be more commonly used:

```
proc macontrol data=Pistons;
   machart Diameter*Hour / span=3;
run;
```

For more information about reading control limit parameters from a LIMITS= data set, see the READLIMITS option later in this list.

READALPHA
specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from a LIMITS= data set when both variables are available in the data set. Thus the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default.

READINDEX=*'value'*
reads control limit parameters from a LIMITS= data set (specified in the PROC MACONTROL statement) for each *process* listed in the MACHART statement. The control limit parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup-variable*
- the value of _INDEX_ matches *value*

The *value* can be up to 48 characters and must be enclosed in quotes.

READLIMITS
specifies that control limit parameters are to be read from a LIMITS= data set specified in the PROC MACONTROL statement. The parameters for a particular *process* are read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

NOTE: In SAS 6.10 and later releases, the READLIMITS option is not necessary.

SIGMA0=*value*
specifies a known (standard) value σ_0 for the process standard deviation σ. The *value* must be positive. By default, the MACONTROL procedure estimates σ from the data using the formulas given in "Methods for Estimating the Standard Deviation" on page 886.

NOTE: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set.

SIGMAS=*value*
specifies the width of the control limits in terms of the multiple k of the standard error of the plotted moving averages on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are 3σ limits.

SPAN=*value*

specifies the number of terms used to calculate the moving average (*value* is an integer greater than 1). The SPAN= option is required unless you read control limit parameters from a LIMITS= data set or a TABLE= data set. See "Plotted Points" on page 874 and "Choosing the Span of the Moving Average" on page 876 for details.

Details: MACHART Statement

Constructing Uniformly Weighted Moving Average Charts

The following notation is used in this section:

A_i	Uniformly weighted moving average for the ith subgroup
w	Span parameter (number of terms in moving average)
μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
x_{ij}	jth measurement in ith subgroup, with j =1, 2, 3, . . . , n_i
n_i	Sample size of ith subgroup
$\overline{X}_i$	Mean of measurements in ith subgroup. If $n_i = 1$, then the subgroup mean reduces to the single observation in the subgroup.
$\overline{\overline{X}}$	Weighted average of subgroup means
$\Phi^{-1}(\cdot)$	Inverse standard normal function

Plotted Points

Each point on the chart indicates the value of the uniformly weighted moving average for that subgroup. The moving average for the ith subgroup (A_i) is defined as

$$A_i = (\overline{X}_1 + \ldots + \overline{X}_i)/i \qquad \text{if } i < w$$

$$A_i = (\overline{X}_i + \ldots + \overline{X}_{i-w+1})/w \quad \text{if } i \geq w$$

where w is the span, or number of terms, of the moving average. You can specify the span with the SPAN= option in the MACHART statement or with the value of _SPAN_ in a LIMITS= data set.

Central Line

By default, the central line on a moving average chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X_N}}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of A_i above and below the central line. The default limits are computed with k = 3 (these are referred to as 3σ *limits*).

- as probability limits defined in terms of α, a specified probability that A_i exceeds the limits

The following table presents the formulas for the limits:

Table 10.15 Limits for Moving Average Chart

Control Limits
$\text{LCL} = \overline{\overline{X}} - k(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$
$\text{UCL} = \overline{\overline{X}} + k(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$
Probability Limits
$\text{LCL} = \overline{\overline{X}} - \Phi^{-1}(1-\alpha/2)(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$
$\text{UCL} = \overline{\overline{X}} + \Phi^{-1}(1-\alpha/2)(\hat{\sigma}/\min(i,w))\sqrt{(1/n_i) + (1/n_{i-1}) + \ldots + (1/n_{1+\max(i-w,0)})}$

These formulas assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and replace $\hat{\sigma}$ with σ_0 in Table 10.15. Note that the limits vary with both n_i and i.

If the subgroup sample sizes are constant ($n_i = n$), the formulas for the control limits simplify to

$$\begin{aligned} \text{LCL} &= \overline{\overline{X}} - \frac{k\hat{\sigma}}{\sqrt{n\min(i,w)}} \\ \text{UCL} &= \overline{\overline{X}} + \frac{k\hat{\sigma}}{\sqrt{n\min(i,w)}} \end{aligned}$$

Refer to Montgomery (1996) for more details. When the subgroup sample sizes are constant, the width of the control limits for the first w moving averages decreases monotonically because each of the first w moving averages includes one more term than the preceding moving average.

If you specify the ASYMPTOTIC option, constant control limits with the following values are displayed:

$$\begin{aligned} \text{LCL} &= \overline{\overline{X}} - \frac{k\hat{\sigma}}{\sqrt{nw}} \\ \text{UCL} &= \overline{\overline{X}} + \frac{k\hat{\sigma}}{\sqrt{nw}} \end{aligned}$$

For asymptotic probability limits, replace k with $\Phi^{-1}(1-\alpha/2)$ in these equations. You can display asymptotic limits by specifying the ASYMPTOTIC option.

You can specify parameters for the moving average limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.

- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify w with the SPAN= option or with the variable _SPAN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Choosing the Span of the Moving Average

There are few published guidelines for choosing the span w. In some applications, practical experience may dictate the choice of w. A more systematic approach is to choose w by considering its effect on the average run length (the expected number of points plotted before a shift is detected). This effect was studied by Roberts (1959), who used simulation methods.

You can use Table 10.16 and Table 10.17 to find a combination of k and w that yields a desired ARL for an in-control process ($\delta = 0$) and for a specified shift of δ.

Table 10.16 Average Run Lengths for One-Sided Uniformly Weighted Moving Average Charts

		w (span)						
k	δ	**2**	**3**	**4**	**5**	**6**	**8**	**10**
2.0	0.00	51.58	60.97	70.58	80.18	89.78	108.65	127.47
2.0	0.25	25.01	26.47	28.00	29.33	30.76	33.08	35.18
2.0	0.50	13.41	13.31	13.40	13.69	14.01	14.66	15.17
2.0	0.75	8.00	7.75	7.78	7.97	8.15	8.60	9.06
2.0	1.00	5.27	5.20	5.29	5.45	5.67	6.15	6.69
2.0	1.50	2.90	3.03	3.24	3.50	3.73	4.23	4.66
2.0	2.00	2.04	2.27	2.51	2.73	2.95	3.32	3.65
2.0	2.50	1.68	1.91	2.11	2.31	2.48	2.78	3.04
2.0	3.00	1.46	1.68	1.85	2.01	2.16	2.40	2.63
2.0	4.00	1.20	1.38	1.52	1.64	1.75	1.94	2.10
2.0	5.00	1.06	1.18	1.31	1.41	1.50	1.65	1.79
2.5	0.00	179.92	204.43	230.32	259.32	287.08	339.71	394.43
2.5	0.25	72.62	71.56	72.48	72.93	73.40	75.54	77.47
2.5	0.50	33.67	30.13	28.54	27.49	26.93	26.29	26.03
2.5	0.75	17.28	15.01	13.91	13.42	13.13	13.00	13.10
2.5	1.00	9.94	8.66	8.20	8.01	7.96	8.24	8.63
2.5	1.50	4.43	4.13	4.21	4.39	4.64	5.17	5.69
2.5	2.00	2.65	2.77	3.03	3.29	3.54	4.01	4.43
2.5	2.50	1.98	2.24	2.50	2.74	2.95	3.32	3.67
2.5	3.00	1.70	1.95	2.17	2.37	2.55	2.86	3.14
2.5	4.00	1.37	1.59	1.76	1.90	2.03	2.28	2.49
2.5	5.00	1.15	1.35	1.51	1.62	1.73	1.92	2.08

Table 10.16 (continued)

k	δ	2	3	4	5	6	8	10
3.0	0.00	792.24	867.57	963.95	1051.77	1150.79	1345.96	1539.75
3.0	0.25	269.28	244.26	231.50	226.25	220.89	209.87	204.74
3.0	0.50	104.18	83.86	72.84	65.43	60.85	54.62	50.34
3.0	0.75	45.69	34.45	28.79	25.69	23.66	21.24	20.15
3.0	1.00	22.73	16.74	14.20	12.89	12.12	11.52	11.45
3.0	1.50	7.65	6.16	5.70	5.64	5.75	6.23	6.78
3.0	2.00	3.77	3.49	3.63	3.89	4.17	4.71	5.20
3.0	2.50	2.46	2.63	2.90	3.18	3.43	3.88	4.28
3.0	3.00	1.96	2.23	2.50	2.74	2.95	3.33	3.65
3.0	4.00	1.57	1.81	2.00	2.18	2.34	2.62	2.87
3.0	5.00	1.30	1.55	1.72	1.85	1.97	2.20	2.40
3.5	0.00	4275.15	4536.99	4853.63	5168.75	5485.97	6088.03	6613.01
3.5	0.25	1281.12	1078.59	964.86	886.26	830.03	751.66	684.98
3.5	0.50	413.30	294.47	235.00	197.27	169.50	136.01	115.48
3.5	0.75	153.50	98.31	73.49	59.29	50.49	40.45	34.53
3.5	1.00	63.68	39.34	29.37	24.06	20.88	17.70	16.12
3.5	1.50	15.84	10.44	8.50	7.78	7.47	7.51	7.97
3.5	2.00	6.06	4.73	4.49	4.61	4.86	5.43	6.01
3.5	2.50	3.27	3.13	3.34	3.63	3.92	4.45	4.91
3.5	3.00	2.31	2.54	2.83	3.11	3.36	3.80	4.19
3.5	4.00	1.77	2.02	2.25	2.45	2.64	2.97	3.27
3.5	5.00	1.48	1.74	1.91	2.06	2.21	2.48	2.71

Table 10.17 Average Run Lengths for Two-Sided Uniformly Weighted Moving Average Charts

		w (span)						
k	δ	2	3	4	5	6	8	10
2.0	0.00	25.46	29.62	33.94	38.08	42.35	51.20	59.48
2.0	0.25	20.43	22.38	24.21	25.87	27.35	30.08	32.33
2.0	0.50	12.73	12.80	13.02	13.29	13.57	14.19	14.84
2.0	0.75	7.87	7.68	7.71	7.86	8.03	8.44	8.90
2.0	1.00	5.24	5.14	5.22	5.40	5.59	6.09	6.60
2.0	1.50	2.90	3.02	3.24	3.48	3.71	4.19	4.63
2.0	2.00	2.04	2.26	2.51	2.73	2.94	3.31	3.63
2.0	2.50	1.67	1.91	2.12	2.30	2.47	2.77	3.03
2.0	3.00	1.46	1.67	1.85	2.01	2.15	2.40	2.63
2.0	4.00	1.20	1.38	1.52	1.64	1.75	1.94	2.10
2.0	5.00	1.06	1.19	1.31	1.41	1.50	1.65	1.79
2.5	0.00	89.48	101.24	114.35	127.74	140.88	166.98	192.93
2.5	0.25	63.12	64.91	67.00	68.75	69.84	72.22	74.49

Table 10.17 *continued*

k	δ	2	3	4	5	6	8	10
2.5	0.50	32.46	29.54	28.20	27.33	26.72	25.92	25.72
2.5	0.75	17.28	14.97	13.85	13.29	13.02	12.81	12.98
2.5	1.00	9.94	8.61	8.16	7.99	8.01	8.23	8.63
2.5	1.50	4.42	4.14	4.20	4.38	4.62	5.16	5.67
2.5	2.00	2.65	2.77	3.03	3.29	3.54	4.00	4.43
2.5	2.50	1.99	2.24	2.50	2.73	2.95	3.33	3.65
2.5	3.00	1.69	1.95	2.17	2.37	2.54	2.86	3.14
2.5	4.00	1.37	1.59	1.76	1.90	2.04	2.27	2.49
2.5	5.00	1.15	1.35	1.51	1.63	1.73	1.92	2.09
3.0	0.00	397.12	436.27	481.16	527.14	574.05	667.68	762.89
3.0	0.25	245.51	228.67	222.75	216.07	213.79	207.03	201.71
3.0	0.50	103.15	83.49	72.47	65.67	60.67	53.93	50.30
3.0	0.75	45.56	34.25	29.01	25.72	23.59	21.12	19.93
3.0	1.00	22.68	16.81	14.19	12.92	12.18	11.54	11.48
3.0	1.50	7.68	6.14	5.71	5.65	5.77	6.23	6.77
3.0	2.00	3.74	3.49	3.63	3.88	4.17	4.71	5.21
3.0	2.50	2.46	2.63	2.90	3.18	3.43	3.89	4.29
3.0	3.00	1.96	2.23	2.50	2.73	2.95	3.32	3.66
3.0	4.00	1.57	1.81	2.00	2.18	2.34	2.62	2.88
3.0	5.00	1.30	1.55	1.72	1.85	1.97	2.20	2.40
3.5	0.00	2217.61	2372.09	2567.27	2775.06	2983.70	3398.08	3810.50
3.5	0.25	1186.27	1027.67	940.30	875.91	826.53	744.59	676.61
3.5	0.50	411.69	295.62	232.68	195.65	169.21	135.73	116.06
3.5	0.75	152.52	97.33	72.30	58.98	50.59	40.22	34.71
3.5	1.00	64.03	39.46	29.18	24.08	20.80	17.54	16.16
3.5	1.50	15.83	10.36	8.47	7.73	7.46	7.56	8.00
3.5	2.00	6.05	4.71	4.49	4.61	4.85	5.44	6.00
3.5	2.50	3.27	3.12	3.34	3.64	3.92	4.44	4.91
3.5	3.00	2.32	2.54	2.83	3.11	3.36	3.80	4.19
3.5	4.00	1.77	2.02	2.25	2.46	2.65	2.97	3.26
3.5	5.00	1.49	1.74	1.91	2.06	2.21	2.48	2.71

For example, suppose you want to construct a two-sided moving average chart with an in-control ARL of 100 and an ARL of 9 for detecting a shift of $\delta = 1$. Table 10.17 shows that the combination $w = 3$ and $k = 2.5$ yields an in-control ARL of 101.24 and an ARL of 8.61 for $\delta = 1$.

Note that you can also use Table 10.16 and Table 10.17 to evaluate an existing moving average chart (see Example 10.7).

The following SAS program computes the average run length for a two-sided moving average chart for various shifts in the mean. This program can be adapted to compute averages run lengths for various combinations of k and w.

```
data sim;
   drop span delta time j y x;
   span=4;
   do shift=0,.25,.5,.75,1,1.5,2,2.5,3,4,5;
      do j=1 to 50000;
         do time=1 to 15000;
            if time<=100 then
               delta=0;
            else
               delta=shift;
            y=delta+rannor(234);
            if time<span then
               x=.;
            else
               x=(y+lag1(y)+lag2(y)+lag3(y))/span;
            if time>=101 and abs(x)>3/sqrt(span)
              then leave;
         end;
         arl=time-100;
         output;
      end;
   end;

proc means;
   class shift;
run;
```

In the preceding program, the size of the span w (SPAN) is 4 and the shifts in the mean are introduced to the variable (Y) $y \sim N(0,1)$ after the first 100 observations. The first DO loop specifies shifts of various magnitude, the second DO loop performs 50000 simulations for each shift, and the third DO loop counts the run length (TIME), that is, the number of samples observed before the control chart signals. A large upper bound (15000) for TIME is specified so that the run length is uncensored.

The program can be generalized for various span sizes by assigning a different value for the variable SPAN and changing the expression for X appropriately. Optionally, you can compute the ARL for a one-sided chart by changing the limits, that is, `x>3/sqrt(span)`. This was the technique used to construct Table 10.16 and Table 10.17.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves the control limit parameters. Table 10.18 lists the variables that can be saved.

Table 10.18 OUTLIMITS= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding limits
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LIMITN	Sample size associated with the control limits

Table 10.18 *continued*

Variable	Description
MEAN	Process mean ($\overline{\overline{X}}$ or μ_0)
SIGMAS	Multiple (k) of standard error of A_i
SPAN	Number of terms in the moving average
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the MACHART statement
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
VAR	*Process* specified in the MACHART statement

The OUTLIMITS= data set does not contain the control limits; instead, it contains control limit parameters that can be used to recompute the control limits.

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variable _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of A_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MACHART statement.

You can use OUTLIMITS= data sets

- to keep a permanent record of the control limit parameters
- to write reports. You may prefer to use OUTTABLE= data sets for this purpose.
- as LIMITS= data sets in subsequent runs of PROC MACONTROL

For an example of an OUTLIMITS= data set, see "Saving Control Limit Parameters" on page 856.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*

- a subgroup moving average variable named by *process* suffixed with *A*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the MACHART statement. For example, consider the following statements:

```
proc macontrol data=Clips;
   machart (Gap Yieldstrength)*Day / span      =3
                                     outhistory=Cliphist;
run;
```

The data set Cliphist would contain nine variables named Day, GapX, GapS, GapA, GapN, YieldstrengthX, YieldstrengthS, YieldstrengthA, and YieldstrengthN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 855.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 10.19 lists the variables that can be saved.

Table 10.19 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on moving average chart
LCLA	Lower control limit for moving average
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
SIGMAS	Multiple (k) of the standard error associated with control limits
SPAN	Number of terms in the moving average
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
SUBX	Subgroup mean

Table 10.19 *continued*

Variable	Description
UCLA	Upper control limit for moving average
UWMA	Uniformly weighted moving average
VAR	*Process* specified in MACHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- *symbol-variable*

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively; or with the corresponding variables in a LIMITS= data set).
2. The variables _VAR_ and _EXLIM_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limit Parameters" on page 856.

ODS Tables

The following table summarizes the ODS tables that you can request with the MACHART statement.

Table 10.20 ODS Tables Produced with the MACHART Statement

Table Name	Description	Options
MAChartSummary	Uniformly weighted moving average chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT
Parameters	Uniformly weighted moving average parameters	TABLE, TABLEALL, TABLEC, TABLEID, TABLEOUT

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. MACHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993

When ODS Graphics is in effect, the MACHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 10.21.

Table 10.21 ODS Graphics Produced by the MACHART Statement

ODS Graph Name	Plot Description
MAChart	Moving average chart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC MACONTROL statement. Each *process* specified in the MACHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the MACHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the ith subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the ith subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating Moving Average Charts from Raw Data" on page 849.

LIMITS= Data Set

You can read preestablished control limits parameters from a LIMITS= data set specified in the PROC MACONTROL statement. The LIMITS= data set used by the MACONTROL procedure does not contain the actual control limits, but rather it contains the parameters required to compute the limits. For example, the following statements read control limit parameters from the data set Parms:

```
proc macontrol data=Parts limits=Parms;
   machart Gap*Day;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the MACONTROL procedure. Such data sets always contain the variables required for a LIMITS= data set; see the section "OUTLIMITS= Data Set" on page 879. The LIMITS= data set can also be created directly using a DATA step.

When you create a LIMITS= data set, you must provide the variable _SPAN_, which specifies the number of terms to use in the moving average. In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables of length 8.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8. Valid values are 'ESTIMATE', 'STANDARD', 'STDMEAN', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

Some advantages of working with a LIMITS= data set are that

- it facilitates reusing a permanently saved set of parameters
- a distinct set of parameters can be read for each *process* specified in the MACHART statement
- it facilitates keeping track of multiple sets of parameters that accumulate for the same *process* as the process evolves over time

For an example, see "Reading Preestablished Control Limit Parameters" on page 859.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC MACONTROL statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the MACONTROL, SHEWHART, or CUSUM procedures or to read output data sets created with SAS summarization procedures such as PROC MEANS.

A HISTORY= data set used with the MACHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- a subgroup standard deviation variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters *X*, *S*, and *N*, respectively.

For example, consider the following statements:

```
proc macontrol history=Cliphist;
   machart (Gap Diameter)*Day  / span=3;
run;
```

The data set Cliphist must include the variables Day, GapX, GapS, GapN, DiameterX, DiameterS, and DiameterN.

Although a moving average variable (named by the *process* name suffixed with *A*) is saved in an OUTHISTORY= data set, it is not required in a HISTORY= data set, because the subgroup mean variable is sufficient to compute the moving averages.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the MACONTROL procedure reads all the observations in a HISTORY= data set. However, if the HISTORY= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Moving Average Charts from Subgroup Summary Data" on page 853.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC MACONTROL statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the MACONTROL procedure.

Table 10.22 lists the variables required in a TABLE= data set used with the MACHART statement:

Table 10.22 TABLE= Data Set Variables

Variable	Description
LCLE	Lower control limit for Moving Average
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
SPAN	Number of terms in the moving average
Subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
SUBX	Subgroup mean
UCLA	Upper control limit for moving average
UWMA	Uniformly weighted moving average

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable of length 8.

For an example of a TABLE= data set, see "Saving Control Limit Parameters" on page 856.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, four methods are available for estimating the process standard deviation σ. Three methods (referred to as the default, MVLUE, and RMSDF) are available with subgrouped data. A fourth method is used if the data are individual measurements (see "Default Method for Individual Measurements" on page 888).

Default Method for Subgroup Samples

This method is the default for moving average charts using subgrouped data. The default estimate of σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \ldots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method for Subgroup Samples

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \ldots + h_N s_N/c_4(n_N)}{h_1 + \ldots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method for Subgroup Samples

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ as follows:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method for Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as

$$\hat{\sigma} = \sqrt{\frac{1}{2(N-1)} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$$

where N is the number of observations, and $x_1, x_2, \ldots, x_N$ are the individual measurements. This formula is given by Wetherill (1977), who states that the estimate of the variance is biased if the measurements are autocorrelated.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	All	*Subgroup-variable*
Vertical	DATA=	*Process*
Vertical	HISTORY=	Subgroup mean variable
Vertical	TABLE=	_UWMA_

For example, the following sets of statements specify the label *Moving Average of Clip Gaps* for the vertical axis and the label *Day* for the horizontal axis of the moving average chart:

```
proc macontrol data=Clips1;
   machart Gap*Day / span=4;
   label Gap = 'Moving Average of Clip Gaps';
   label Day = 'Day';
run;

proc macontrol history=cliphist;
   machart Gap*Day / span=4;
   label GapX = 'Moving Average of Clip Gaps';
   label Day  = 'Day';
run;

proc macontrol table=cliptab;
   machart Gap*Day;
   label _uwma_   = 'Moving Average of Clip Gaps';
   label Day      = 'Day';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is

not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: MACHART Statement

This section provides advanced examples of the MACHART statement.

Example 10.6: Specifying Standard Values for the Process Mean and Process Standard Deviation

NOTE: See *Standard Values for Moving Average Charts* in the SAS/QC Sample Library.

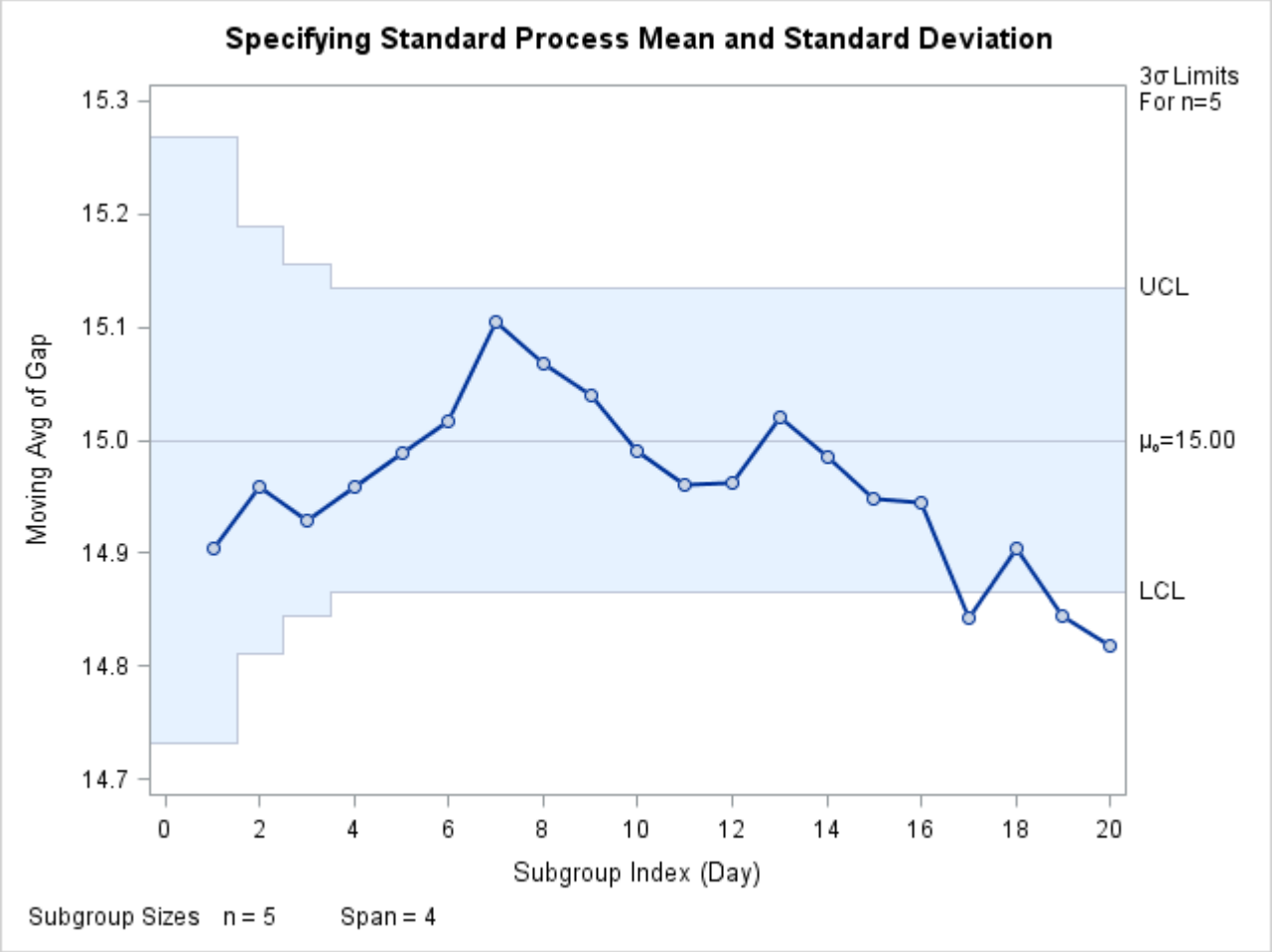
By default, the MACHART statement estimates the process mean (μ) and standard deviation (σ) from the data. This is illustrated in "Getting Started: MACHART Statement" on page 849. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

For example, suppose it is known that the metal clip manufacturing process (introduced in "Creating Moving Average Charts from Raw Data" on page 849) has a mean of 15 and standard deviation of 0.2. The following statements specify these standard values:

```
ods graphics on;
title 'Specifying Standard Process Mean and Standard Deviation';
proc macontrol data=Clips1;
   machart Gap*Day /
        odstitle = title
        mu0      = 15
        sigma0   = 0.2
        span     = 4
        xsymbol  = mu0
        markers;
run;
```

The XSYMBOL= option specifies the label for the central line. The resulting chart is shown in Output 10.6.1.

Output 10.6.1 Specifying Standard Values with MU0= and SIGMA0=



The central line and control limits are determined using μ_0 and σ_0 (see the equations in Table 10.15). Output 10.6.1 indicates that the process is out-of-control since the moving averages for Day=17, Day=19, and Day=20 lie below the lower control limit.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data Cliplim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Gap';
   _subgrp_ = 'Day';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 15;
   _stddev_ = 0.2;
   _span_   = 4;
run;

proc macontrol data=Clips1 limits=Cliplim;
   machart Gap*Day / xsymbol=mu0
                     odstitle = title
                     markers;
run;
```

The variable _SPAN_ is required, and its value provides the number of terms in the moving average. The variables _VAR_ and _SUBGRP_ are also required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the MACHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting chart (not shown here) is identical to the one shown in Output 10.6.1.

Example 10.7: Annotating Average Run Lengths on the Chart

NOTE: See *ARLs Shown on a Moving Average Chart* in the SAS/QC Sample Library.

You can use Table 10.16 and Table 10.17 to find a moving average chart scheme with the desired average run length properties. Specifically, you can find a combination of k and w that yields a desired ARL for an in-control process ($\delta = 0$) and for a specified shift of δ.

You can also use these tables to evaluate an existing moving average chart scheme. For example, the moving average chart shown in Output 10.6.1 has a two-sided scheme with $w = 4$ and $k = 3$. Suppose you want to detect a shift of $\delta = .5$. From Table 10.17, the average run length with $w = 4$, $k = 3$, and $\delta = .5$ is 72.47. The in-control average run length ($\delta = 0$) for this scheme is 481.16.

The following statements create an inset data set that can be read to display these ARL values on the moving average chart:

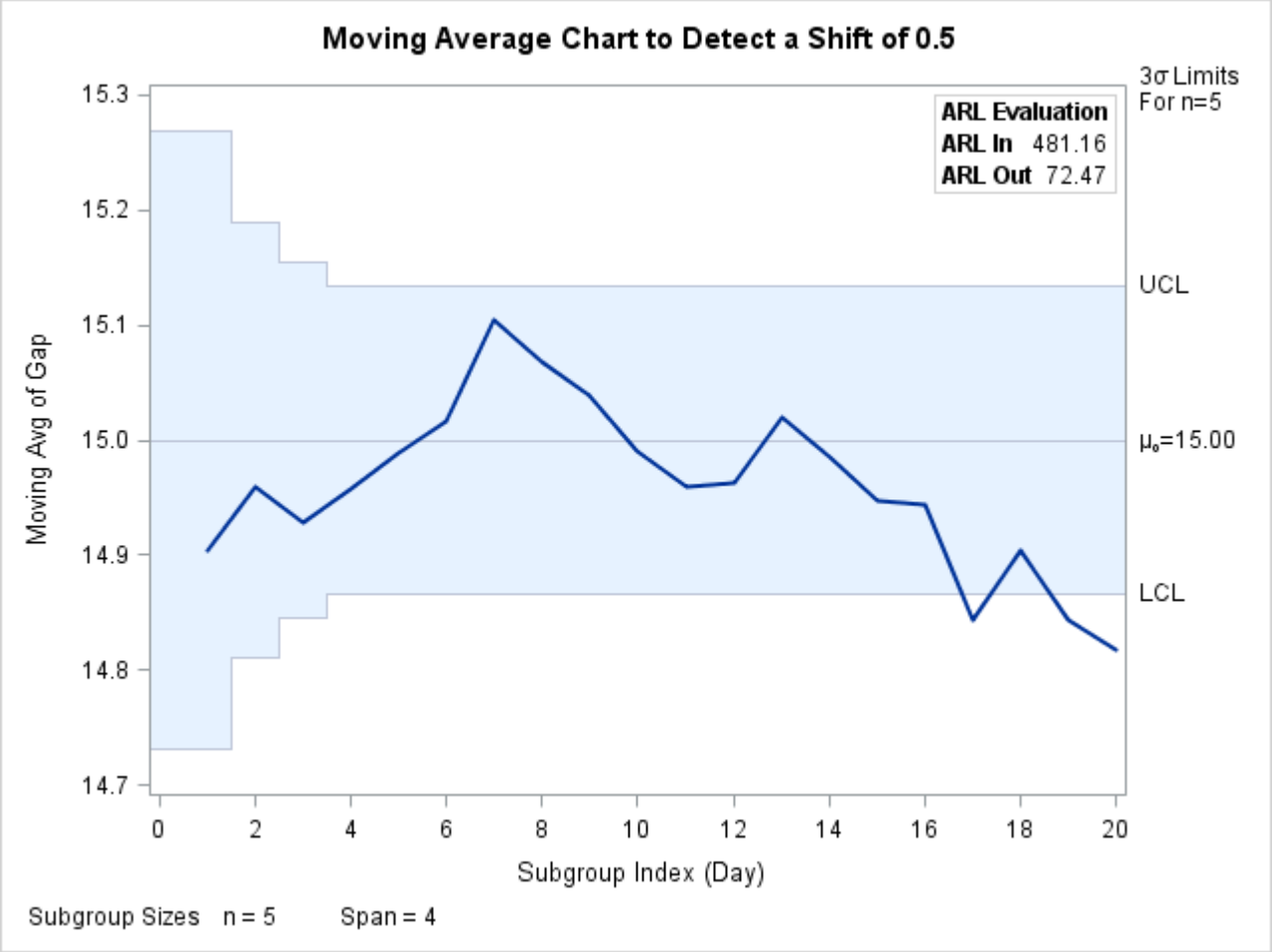
```
data ARLinset;
   length _label_ $ 8;
   _label_ = 'ARL In';
   _value_ = 481.16;
   output;
   _label_ = 'ARL Out';
   _value_ = 72.47;
   output;
run;
```

The following statements create the moving average chart shown in Output 10.7.1.

```
title 'Moving Average Chart to Detect a Shift of 0.5';
ods graphics on;
proc macontrol data=Clips1;
   machart Gap*Day / mu0      = 15
                     sigma0   = 0.2
                     span     = 4
                     xsymbol  = mu0
                     odstitle = title;
   inset data = ARLinset / header = 'ARL Evaluation'
                           pos    = ne;
run;
```

The average run lengths in this example (481.16 and 72.27) are simply copied from Table 10.17. You can generalize the preceding program so that it computes the average run lengths by incorporating the simulation program from the section "Choosing the Span of the Moving Average" on page 876.

Output 10.7.1 Displaying Average Run Lengths on Chart



For more information on annotating charts with insets, refer to "INSET Statement: MACONTROL Procedure" on page 892.

INSET Statement: MACONTROL Procedure

Overview: INSET Statement

The INSET statement enables you to enhance a moving average control chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. A possible application of an inset is to present moving average parameters on the chart rather than displaying them in a legend. An inset can also display arbitrary values provided in a SAS data set.

Note that the INSET statement by itself does not produce a display but must be used in conjunction with an MACHART or EWMACHART statement. Insets are not available with line printer charts, so the INSET

statement is not applicable when the LINEPRINTER option is specified in the PROC MACONTROL statement.

You can use options in the INSET statement to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started: INSET Statement

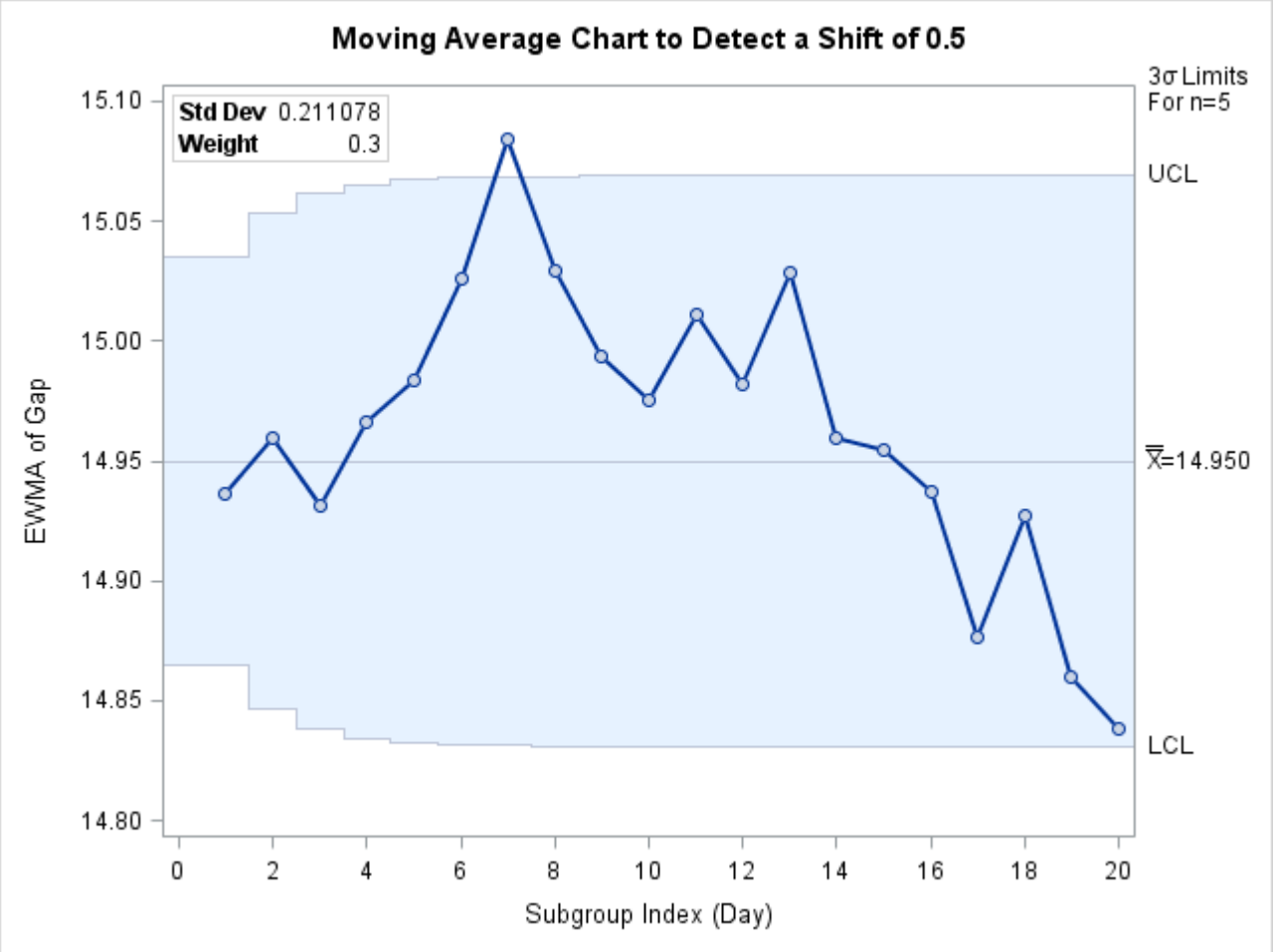
This section introduces the INSET statement with a basic example showing how it is used. See "INSET and INSET2 Statements: SHEWHART Procedure" on page 1975 for a complete description of the INSET statement.

This example is based on the same scenario as the first example in the "Getting Started" section of "EWMACHART Statement: MACONTROL Procedure" on page 795. An EWMA chart is used to analyze data from the manufacture of metal clips. The following statements create a data set containing measurements to be analyzed and the EWMA chart shown in Figure 10.18.

```
data Clips1;
   input Day @ ;
   do i=1 to 5;
      input Gap @ ;
      output;
   end;
   drop i;
   datalines;
 1  14.76  14.82  14.88  14.83  15.23
 2  14.95  14.91  15.09  14.99  15.13
 3  14.50  15.05  15.09  14.72  14.97
 4  14.91  14.87  15.46  15.01  14.99
 5  14.73  15.36  14.87  14.91  15.25
 6  15.09  15.19  15.07  15.30  14.98
 7  15.34  15.39  14.82  15.32  15.23
 8  14.80  14.94  15.15  14.69  14.93
 9  14.67  15.08  14.88  15.14  14.78
10  15.27  14.61  15.00  14.84  14.94
11  15.34  14.84  15.32  14.81  15.17
12  14.84  15.00  15.13  14.68  14.91
13  15.40  15.03  15.05  15.03  15.18
14  14.50  14.77  15.22  14.70  14.80
15  14.81  15.01  14.65  15.13  15.12
16  14.82  15.01  14.82  14.83  15.00
17  14.89  14.90  14.60  14.40  14.88
18  14.90  15.29  15.14  15.20  14.70
19  14.77  14.60  14.45  14.78  14.91
20  14.80  14.58  14.69  15.02  14.85
;
```

```
ods graphics on;
proc macontrol data=Clips1;
   ewmachart Gap*Day / weight    = 0.3
                       odstitle = title
                       markers
                       nolegend;
   inset stddev weight;
run;
```

Figure 10.18 Exponentially Weighted Moving Average Chart with an Inset



Syntax: INSET Statement

The syntax for the INSET statement is as follows:

INSET *keyword-list* < / *options* > **;**

You can use any number of INSET statements in the MACONTROL procedure. However, when ODS Graphics is enabled, at most two insets are displayed inside the plot area and at most two are displayed in the chart margins. Each INSET statement produces a separate inset and must follow an EWMACHART

or MACHART statement. The inset appears on every panel (page) produced by the last chart statement preceding it.

Keywords specify the statistics to be displayed in an inset; options control the inset's location and appearance. A complete description of the INSET statement syntax is given in the section "Syntax: INSET and INSET2 Statements" on page 1981 of Chapter 19, "The SHEWHART Procedure." The INSET statement options are identical in the MACONTROL and SHEWHART procedures, but the available keywords are different. The options are listed in Table 19.89. The keywords available with the MACONTROL procedure are listed in Table 10.23 to Table 10.26.

Table 10.23 Summary Statistics

Keyword	Description
MEAN	estimated or specified process mean
N	nominal subgroup size
NMIN	minimum subgroup size
NMAX	maximum subgroup size
NOUT	number of subgroups outside control limits
NLOW	number of subgroups below lower control limit
NHIGH	number of subgroups above upper control limit
STDDEV	estimated or specified process standard deviation
DATA=	arbitrary values from *SAS-data-set*

Table 10.24 Parameter for Uniformly Weighted Moving Average Charts

Keyword	Description
SPAN	number of terms used to calculate moving average

Table 10.25 Parameter for Exponentially Weighted Moving Average Charts

Keyword	Description
WEIGHT	weight assigned to most recent subgroup mean in computation of the EWMA

You can use the keywords in Table 10.26 only when producing ODS Graphics output. The labels for the statistics use Greek letters.

Table 10.26 Keywords Specific to ODS Graphics Output

Keyword	Description
UMU	estimated or specified process mean
USIGMA	estimated or specified process standard deviation

References

American Society for Quality Control (1983). *ASQC Glossary and Tables for Statistical Quality Control*. Milwaukee: ASQC.

American Society for Testing and Materials (1976). *ASTM Manual on Presentation of Data and Control Chart Analysis*. Philadelphia: ASTM.

Burr, I. W. (1969). "Control Charts for Measurements with Varying Sample Sizes." *Journal of Quality Technology* 1:163–167.

Burr, I. W. (1976). *Statistical Quality Control Methods*. New York: Marcel Dekker.

Crowder, S. V. (1987a). "Average Run Lengths of Exponentially Weighted Moving Average Charts." *Journal of Quality Technology* 19:161–164.

Crowder, S. V. (1987b). "A Simple Method for Studying Run-Length Distributions of Exponentially Weighted Moving Average Charts." *Technometrics* 29:401–408.

Hunter, J. S. (1986). "The Exponentially Weighted Moving Average." *Journal of Quality Technology* 18:203–210.

Kume, H. (1985). *Statistical Methods for Quality Improvement*. Tokyo: AOTS Chosakai.

Montgomery, D. C. (1996). *Introduction to Statistical Quality Control*. 3rd ed. New York: John Wiley & Sons.

Nelson, L. S. (1983). "The Deceptiveness of Moving Averages." *Journal of Quality Technology* 15:99–100.

Nelson, L. S. (1989). "Standardization of Shewhart Control Charts." *Journal of Quality Technology* 21:287–289.

Nelson, L. S. (1994). "Shewhart Control Charts with Unequal Subgroup Sizes." *Journal of Quality Technology* 26:64–67.

Roberts, S. W. (1959). "Control Chart Tests Based on Geometric Moving Averages." *Technometrics* 1:239–250.

Robinson, P. B., and Ho, T. Y. (1978). "Average Run Lengths of Geometric Moving Average Charts by Numerical Methods." *Technometrics* 20:85–93.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986). *Modern Methods for Quality Control and Improvement*. New York: John Wiley & Sons.

Wetherill, G. B. (1977). *Sampling Inspection and Quality Control*. 2nd ed. New York: Chapman & Hall.

Wortham, A. W., and Heinrich, G. F. (1972). "Control Charts Using Exponential Smoothing Techniques." *ASQC Annual Conference Transactions* 26:451–458.

Wortham, A. W., and Ringer, L. J. (1971). "Control via Exponential Smoothing." *Logistics Review* 7:33–40.

Chapter 11
Introduction to Multivariate Process Monitoring Procedures

Contents

Overview: MVP Procedures **897**
MVP Analysis Phases **898**
Phase I Analysis: Building a Model of a Process 899
Phase II Analysis: Process Monitoring 899
References **900**

Overview: MVP Procedures

This chapter provides an overview of the SAS/QC procedures that perform multivariate process monitoring. They are the MVPMODEL, MVPMONITOR, and MVPDIAGNOSE procedures, referred to collectively as the *MVP procedures*. The MVP procedures are used to monitor multivariate process variation over time in order to determine whether a process is stable, to detect changes in a stable process, and to investigate causes of unusual variation.

The MVPMODEL procedure builds a principal component model from multivariate process data. It uses principal component analysis (PCA) techniques that evolved in the field of chemometrics for monitoring hundreds or even thousands of correlated process variables; see Kourti and MacGregor (1995, 1996) for an introduction. A principal component model reduces the dimensionality of the data by projecting the process measurements to a low-dimensional subspace that is defined by a small number of principal components. This subspace is known as the *model hyperplane*. PROC MVPMODEL computes T^2 and squared prediction error (SPE) statistics based on these principal components. See Chapter 13, "The MVPMODEL Procedure," for details.

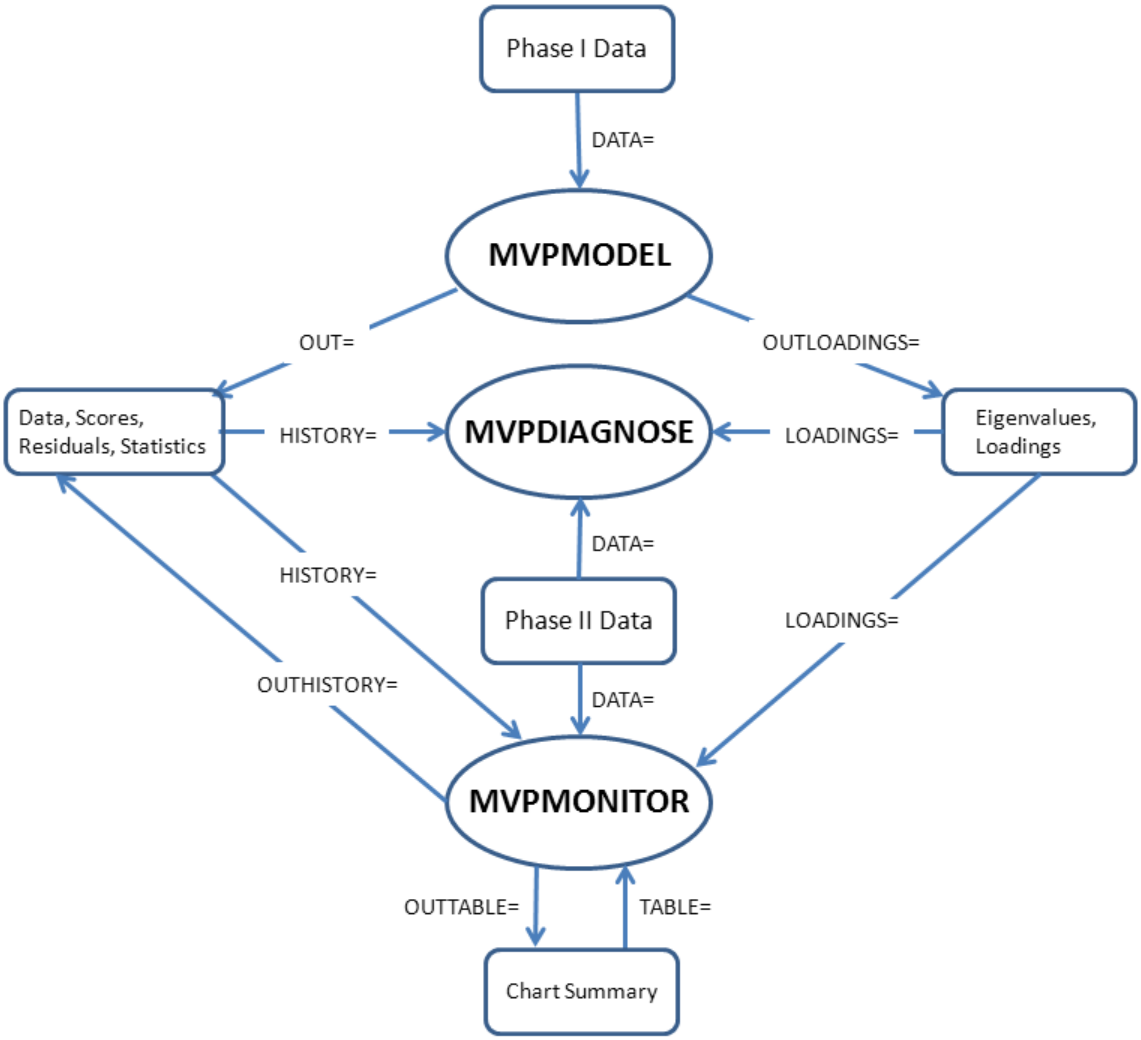
The principal component model and the computed statistics that PROC MVPMODEL produces serve as input to the MVPMONITOR and MVPDIAGNOSE procedures.

The MVPMONITOR procedure creates multivariate control charts of the T^2 and SPE statistics. These control charts are used to monitor and categorize variation in the process. Multivariate control charts can detect unusual variation that is not uncovered by individually monitoring the variables with univariate control charts, such as Shewhart charts. See Chapter 14, "The MVPMONITOR Procedure," for details.

The MVPDIAGNOSE procedure produces score plots and contribution plots that can be used to investigate the unusual variation. A score plot is a scatter plot of scores that are associated with a pair of principal components for the multivariate process data. A contribution plot shows the contributions of the process variables to the T^2 or SPE statistic for a single observation. Both types of plots can be useful in understanding the unusual variation in the process. See Chapter 12, "The MVPDIAGNOSE Procedure," for details.

Figure 11.1 shows the MVP procedures and their various inputs and outputs.

Figure 11.1 The MVP Procedures



The MVP procedures are represented by ovals. Input and output data sets are represented by rounded rectangles and labeled with brief descriptions of their contents. The arrows between procedures and data sets are labeled with the names of the procedure options that are used to specify the data sets.

The distinction between the Phase I and Phase II data shown in Figure 11.1 is discussed in the next section.

MVP Analysis Phases

The two primary scenarios for using the MVP procedures are referred to as Phase I and Phase II analysis. In Phase I analysis you build a principal component model of a process and determine whether the process is stable. In Phase II analysis you apply the model to new data to monitor the process over time.

Phase I Analysis: Building a Model of a Process

The first step in using the MVP procedures is to build a principal component model by running PROC MVPMODEL on an initial sample of process data, which is labeled "Phase I Data" in Figure 11.1. PROC MVPMODEL produces tabular and graphical output to enable you to determine an appropriate number of principal components for the model. You can also use cross validation to have the procedure automatically select the number of principal components.

After you determine the number of components, you use PROC MVPMODEL to save the eigenvalues, principal component loadings, and other information that describes the model to an OUTLOADINGS= data set. At the same time, you can save the Phase I data and the corresponding principal component scores, residuals, and T^2 and SPE statistics to an OUT= data set.

Next, you use the MVPMONITOR procedure to determine whether the process is stable. PROC MVPMONITOR reads the OUT= data set that is produced by PROC MVPMODEL as a HISTORY= data set and creates control charts for the T^2 and SPE statistics that are computed from the Phase I data. A point that falls outside a chart's control limits indicates unusual variation.

The purpose of using a control chart to signal unusual variation is to distinguish *special cause* variation from *common cause* variation. Special causes, also referred to as *assignable causes*, are local, sporadic, or transient causes of variation, whereas common causes are inherent in a process. You should investigate out-of-control points to determine whether they signal assignable causes of variation.

Contribution plots show how the original process variables contribute to variation displayed by the T^2 and SPE charts. PROC MVPMONITOR can produce contribution plots automatically for out-of-control points. You can also use the MVPDIAGNOSE procedure to produce contribution plots of observations from the HISTORY= data set.

Based on your Phase I analysis, you can proceed to monitor the process by using the existing principal component model or collect more Phase I data and build a new model. The goal is to have a model of a stable process before you move on to Phase II analysis.

Phase II Analysis: Process Monitoring

In Phase II analysis you apply the principal component model that you built in Phase I to new process measurements, labeled "Phase II Data" in Figure 11.1. The MVPMONITOR and MVPDIAGNOSE procedures read Phase II data from a data set that is specified with the DATA= option. You must also specify the OUTLOADINGS= data set that was produced by PROC MVPMODEL as a LOADINGS= data set. The loadings are used to compute principal component scores, residuals, and statistics for the Phase II data.

PROC MVPMONITOR can save the Phase II data and the corresponding computed values in an OUTHISTORY= data set. An OUTHISTORY= data set has the same structure as an OUT= data set that is produced by PROC MVPMODEL, and it can be read by the MVPMONITOR and MVPDIAGNOSE procedures as a HISTORY= data set, as shown in Figure 11.1.

You use PROC MVPMONITOR to produce control charts of Phase II data in order to monitor the process for continued stability. Unusual variation might indicate that the process is no longer stable. T^2 charts detect unusual variation within the model hyperplane, whereas SPE charts detect unusual variation outside the model hyperplane. Unexpected SPE chart variation might indicate that the process variation has changed

such that the principal component model no longer adequately represents it. You can save a summary of a control chart by using the OUTTABLE= option in an SPECHART or TSQUARECHART statement in PROC MVPMONITOR. This enables you to "replay" a chart by specifying a TABLE= input data set. You can also read a TABLE= data set that was created outside PROC MVPMONITOR to display a chart with customized control limits.

In addition to producing contribution plots of Phase II data, the MVPDIAGNOSE procedure also produces score plots. Score plots can provide insight into the nature of the variation that is represented by the components. You can compare score plots of Phase II data that are produced by PROC MVPDIAGNOSE to score plots of Phase I data that are produced by PROC MVPMODEL.

References

Kourti, T., and MacGregor, J. F. (1995). "Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods." *Chemometrics and Intelligent Laboratory Systems* 28:3–21.

Kourti, T., and MacGregor, J. F. (1996). "Multivariate SPC Methods for Process and Product Monitoring." *Journal of Quality Technology* 28:409–428.

Miller, P., Swanson, R. E., and Heckler, C. H. E. (1998). "Contribution Plots: A Missing Link in Multivariate Quality Control." *Applied Mathematics and Computer Science* 8:775–792.

Chapter 12
The MVPDIAGNOSE Procedure

Contents

Overview: MVPDIAGNOSE Procedure **901**
Getting Started: MVPDIAGNOSE Procedure **902**
Syntax: MVPDIAGNOSE Procedure **906**
PROC MVPDIAGNOSE Statement 906
BY Statement 907
CONTRIBUTIONPANEL Statement 908
CONTRIBUTIONPLOT Statement 909
ID Statement 910
SCOREMATRIX Statement 910
SCOREPLOT Statement 911
TIME Statement 912
Common Plot Statement Options 913
Details: MVPDIAGNOSE Procedure **914**
Contribution Plots 914
Paneled Contribution Plot Layouts 914
Input Data Sets 915
ODS Graphics 917
Examples: MVPDIAGNOSE Procedure **917**
Example 12.1: Phase II Analysis with MVPDIAGNOSE 917
References **922**

Overview: MVPDIAGNOSE Procedure

The MVPDIAGNOSE procedure is used in conjunction with the MVPMODEL and MVPMONITOR procedures to monitor multivariate process variation over time, to determine whether the process is stable, and to detect and diagnose changes in a stable process. Collectively these three procedures are referred to as the *MVP procedures*. See Chapter 11, "Introduction to Multivariate Process Monitoring Procedures," for a description of how the MVP procedures work together, and Chapter 13, "The MVPMODEL Procedure," and Chapter 14, "The MVPMONITOR Procedure," for details about the other MVP procedures.

The MVPDIAGNOSE procedure produces the following graphs that can provide insight into the variation in a process:

- score plots for pairs of principal components

- score plot matrices containing pairwise plots for multiple pairs of principal components
- contribution plots for individual observations
- paneled contribution plots for multiple observations

Each point in a score plot corresponds to a single observation from the input data set. A contribution plot displays the process variable contributions to a squared prediction error (SPE) or T^2 statistic from a single observation in the input data set. Therefore, each observation in the input data is independent in how PROC MVPDIAGNOSE handles it. This enables you to preprocess the input data flexibly by using the DATA step, WHERE expressions, and other SAS language elements to select the data to plot.

NOTE: ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement before invoking the procedure) in order for the MVPDIAGNOSE procedure to produce graphical output.

Getting Started: MVPDIAGNOSE Procedure

This example illustrates the basic features of the MVPDIAGNOSE procedure by using airline flight delay data available from the U.S. Bureau of Transportation Statistics at http://www.transtats.bts.gov. Suppose you want to compare process variable contributions for an out-of-control T^2 statistic with contributions for adjacent observations. This kind of comparison can help you understand the underlying causes of unusual variation in the process.

The following statements create a SAS data set named MWflightDelays that provides the delays for flights that originated in the midwestern United States. The data set contains variables for nine airlines: AA (American Airlines), CO (Continental Airlines), DL (Delta Airlines), F9 (Frontier Airlines), FL (AirTran Airways), NW (Northwest Airlines), UA (United Airlines), US (US Airways), and WN (Southwest Airlines).

```
data MWflightDelays;
   format flightDate MMDDYY8.;
   label flightDate='Date';
   input flightDate :MMDDYY8. AA CO DL F9 FL NW UA US WN;
   datalines;
02/01/07 14.9  7.1  7.9  8.5 14.8  4.5  5.1 13.4  5.1
02/02/07 14.3  9.6 14.1  6.2 12.8  6.0  3.9 15.3 11.4
02/03/07 23.0  6.1  1.7  0.9 11.9 15.2  9.5 18.4  7.6
02/04/07  6.5  6.3  3.9 -0.2  8.4 18.8  6.2  8.8  8.0
02/05/07 12.0 14.1  3.3 -1.3 10.0 13.1 22.8 16.5 11.5
02/06/07 31.9  8.6  4.9  2.0 11.9 21.9 29.0 15.5 15.2
02/07/07 14.2  3.0  2.1 -0.9 -0.6  7.8 19.9  8.6  6.4
02/08/07  6.5  6.8  1.8  7.7  1.3  6.9  6.1  9.2  5.4
02/09/07 12.8  9.4  5.5  9.3 -0.2  4.6  7.6  7.8  7.5
02/10/07  9.4  3.5  1.5 -0.2  2.2  9.9  3.1 12.5  3.0
02/11/07 12.9  5.4  0.9  6.8  2.1  7.9  3.7 10.7  5.6
02/12/07 34.6 15.9  1.8  1.0  4.5 10.2 14.0 19.1  4.9
02/13/07 34.0 16.0  4.4  6.1 18.3  9.1 30.2 46.3 50.6
02/14/07 21.2 45.9 16.6 12.5 35.1 23.8 40.4 43.6 35.2
02/15/07 46.6 36.3 23.9 20.8 30.4 24.3 30.3 59.9 25.6
02/16/07 31.2 20.8 15.2 20.1  9.1 12.9 22.9 36.4 16.4
;
```

The observations for a given date are the average delays in minutes for flights that depart from the Midwest. For example, on February 2, 2007, F9 (Frontier Airlines) flights departed an average of 6.2 minutes late.

The first step in multivariate process monitoring of the data is to build a principal component model of the process variation. The following statements use PROC MVPMODEL to create a model with three principal components. (See Chapter 13, "The MVPMODEL Procedure," for details.)

```
proc mvpmodel data=MWflightDelays ncomp=3 noprint
              out=mvpair outloadings=mvpairloadings;
   var AA CO DL F9 FL NW UA US WN;
run;
```

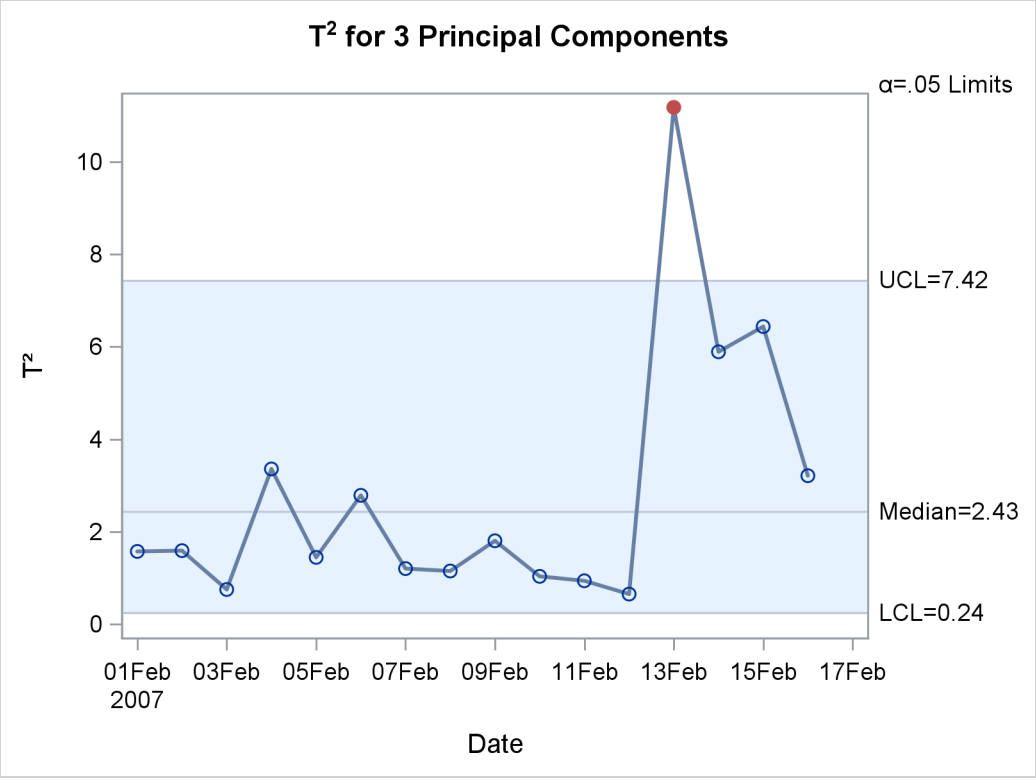
The mvpair data set contains the process data and associated principal component scores. The mvpairloadings data set contains the principal component loadings for the process variables and other data that describe the model.

The following statements create a T^2 control chart by using the principal components. (See Chapter 14, "The MVPMONITOR Procedure," for details.)

```
ods graphics on;
proc mvpmonitor history=mvpair loadings=mvpairloadings;
   time flightDate;
   tsquarechart / contributions;
run;
```

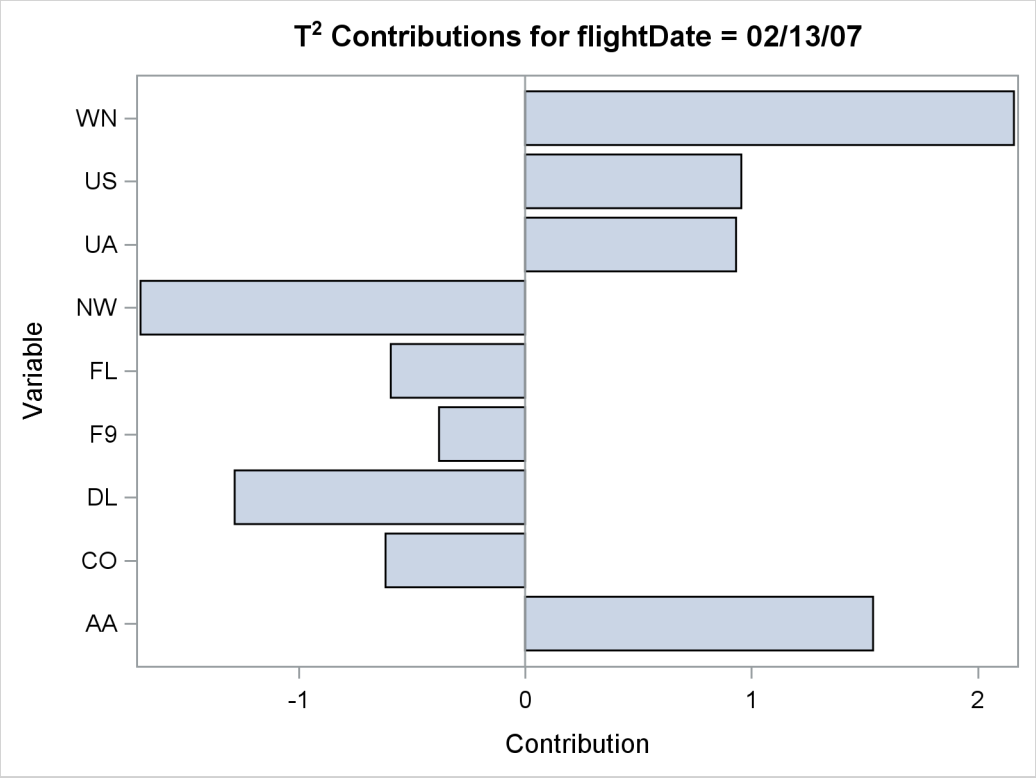
The CONTRIBUTIONS option produces contribution plots for any out-of-control points in the T^2 chart. Figure 12.1 shows the T^2 chart.

Figure 12.1 Multivariate Control Chart for T^2 Statistics



The T^2 chart shows an out-of-control point on February 13, 2007. Figure 12.2 shows the contribution plot for this date that was produced by the CONTRIBUTIONS option.

Figure 12.2 Contribution Plot for Out-of-Control Point

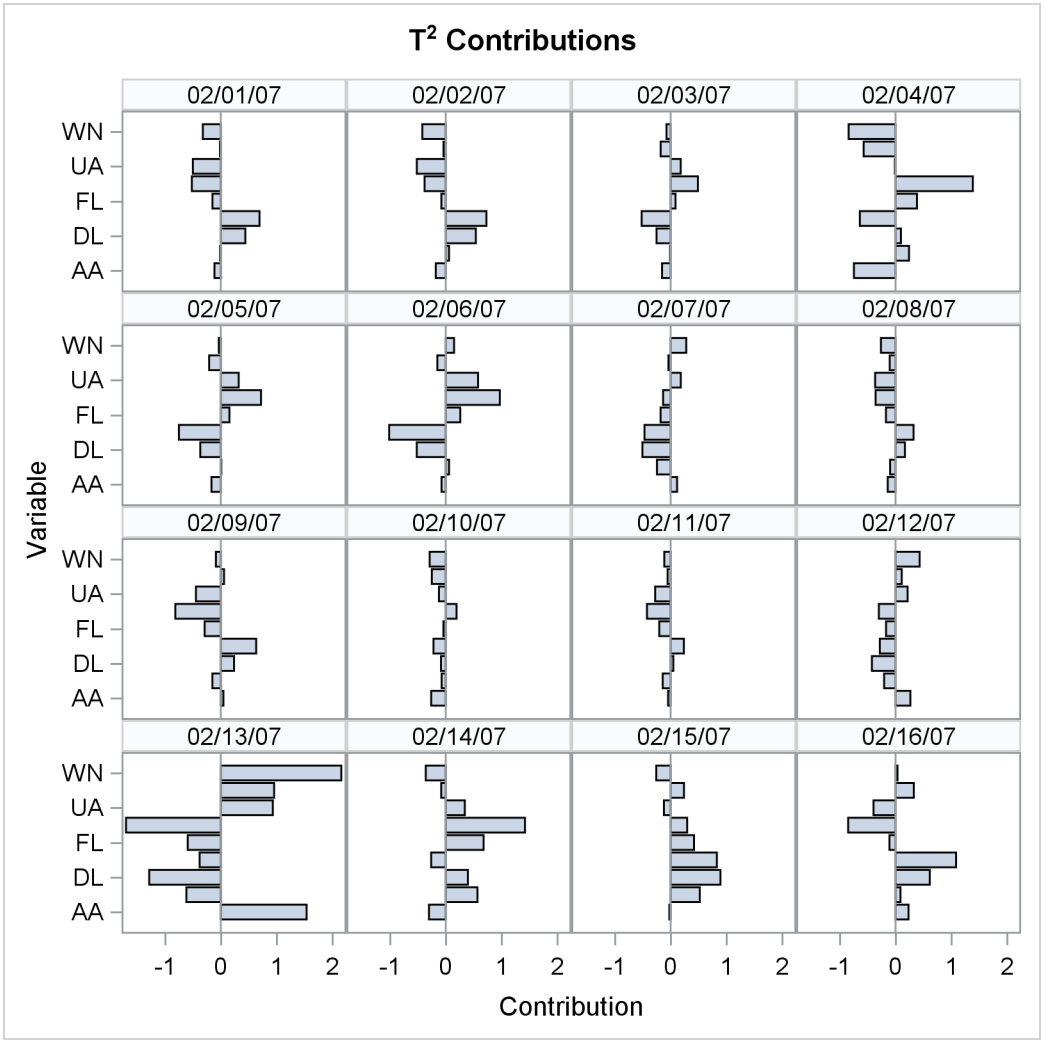


The contribution plot shows that the delays for airlines AA, DL, NW, and WN are the major contributors to the out-of-control point. You can use PROC MVPDIAGNOSE to compare the contributions for this point to those for adjacent points. The following statements produce paneled contribution plots of all the observations in mvpair:

```
proc mvpdiagnose history=mvpair loadings=mvpairloadings;
   time flightDate;
   contributionpanel / type=tsquare;
run;
```

Figure 12.3 shows the paneled contribution plots.

Figure 12.3 T^2 Contributions for Flight Delays



The contribution plot for February 13 is in the lower-left corner of the plot. Notice that the magnitudes of all the process variable contributions are quite large for this date compared to those for the other dates. All the process variables contributed strongly to the out-of-control T^2 statistic. This implies that something unusual occurred on February 13 that affected the flight delays for all the airlines.

In fact, on this day a strong winter storm battered the Midwest. This is an example of variation due to a *special cause*. Special causes, also referred to as *assignable causes*, are local, sporadic, or transient problems in a process. They are distinguished from *common causes* of variation, which are inherent in a system. Control charts are used to monitor the process for the occurrence of special causes and to measure and potentially to reduce the effects of common causes.

Syntax: MVPDIAGNOSE Procedure

PROC MVPDIAGNOSE < *options* > ;
BY *variables* ;
CONTRIBUTIONPANEL < / *options* > ;
CONTRIBUTIONPLOT < / *options* > ;
ID *variables* ;
SCOREMATRIX < / *options* > ;
SCOREPLOT < / *options* > ;
TIME *variable* ;

The following sections describe the PROC MVPDIAGNOSE statement and then describe the other statements in alphabetical order.

PROC MVPDIAGNOSE Statement

PROC MVPDIAGNOSE < *options* > ;

The PROC MVPDIAGNOSE statement invokes the MVPDIAGNOSE procedure and specifies input data sets. You can specify the following *options*:

DATA=*SAS-data-set*
specifies an input SAS data set that contains process measurement data for a Phase II analysis. If you specify a DATA= data set, you must also specify a LOADINGS= data set. You cannot specify both the HISTORY= option and the DATA= option. See the section "DATA= Data Set" on page 915 for details about DATA= data sets.

HISTORY=*SAS-data-set*
specifies an input SAS data set that contains process variable data that are augmented with principal component scores, multivariate summary statistics, and other calculated values. Usually you create a HISTORY= data set by using the OUT= option in the PROC MVPMODEL statement or the OUTHISTORY= option in the PROC MVPMONITOR statement. You cannot specify both the DATA= option and the HISTORY= option. See the section "HISTORY= Data Set" on page 915 for details about HISTORY= data sets.

LOADINGS=*SAS-data-set*
specifies an input SAS data set that contains eigenvalues, principal component loadings, and process variable means and standard deviations that are used to compute principal component scores and multivariate summary statistics for a Phase II analysis. Usually you create a LOADINGS= data set by using the OUTLOADINGS= option in the PROC MVPMODEL statement. See the section "LOADINGS= Data Set" on page 916 for details about LOADINGS= data sets.

MISSING=AVG | NONE
specifies how observations that have missing process variable values in the DATA= data set are to be handled. The option MISSING=AVG specifies that missing values for a given variable be replaced by the average of the nonmissing values for that variable. The default is MISSING=NONE, which excludes from the analysis any observation that has missing values for any of the process variables.

PREFIX=*name*

specifies the prefix that is used to identify variables that contain principal component scores in the HISTORY= data set. For example, if you specify PREFIX=ABC, PROC MVPDIAGNOSE attempts to read the score variables ABC1, ABC2, ABC3, and so on. The default prefix is Prin, which is also the default score variable prefix for data sets created by using the OUT= option in the PROC MVPMODEL statement. If you are using an OUT= data set from PROC MVPMODEL in the HISTORY= data set, the PREFIX= values must match. That is, the PREFIX= value that is specified in the PROC MVPDIAGNOSE statement must match the PREFIX= value in the data set that is specified in the OUT= option in the PROC MVPMODEL statement.

NOTE: The number of characters in the prefix plus the number of digits that are required to enumerate the principal components must not exceed the current name length defined by the VALIDVARNAME= system option.

RPREFIX=*name*

specifies the prefix that is used to identify variables that contain residuals in the HISTORY= data set. A residual variable name is formed by appending a process variable name to the prefix. The default prefix is R_, which is also the default residual variable prefix for data sets created by using the OUT= option in the PROC MVPMODEL statement. If you are using a data set produced with the OUT= option in the PROC MVPMODEL statement as a HISTORY= data set, the RPREFIX= value must match the RPREFIX= value specified when the OUT= data set was created by PROC MVPMODEL.

If the combined length of the residual prefix and a process variable name exceeds the maximum name length defined by the VALIDVARNAME= system option, characters are removed from the middle of the process variable name before it is appended to the residual prefix. For example, if you specify RPREFIX=*Residual_* (nine characters), the maximum variable name length is 32, and there is a process variable named PrimaryThermometerReading (25 characters), then two characters are dropped from the middle of the process variable name. The resulting residual variable name is Residual_PrimaryThermeterReading.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MVPDIAGNOSE to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the MVPDIAGNOSE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CONTRIBUTIONPANEL Statement

CONTRIBUTIONPANEL < */ options* > **;**

The CONTRIBUTIONPANEL statement displays a paneled layout of the contribution plots for each observation in the input data set, up to a maximum specified by the MAXNPLOTS= option. Individual contribution plots are displayed in panels from left to right and top to bottom, and they are identified by TIME variable values or observation numbers. You can use the CONTRIBUTIONPLOT statement to display each contribution plot as a separate graph.

Table 12.1 summarizes the *options* available in the CONTRIBUTIONPANEL statement.

Table 12.1 CONTRIBUTIONPANEL Statement Options

Option	Description
MAXNPLOTS=	Specifies the maximum number of contribution plots displayed
MAXNVARS=	Specifies the maximum number of process variable contributions displayed in each plot
NCOLS=	Specifies the number of columns in the panel layout
NROWS=	Specifies the number of rows in the panel layout
ODSFOOTNOTE=	Adds a footnote to the paneled contribution plots
ODSFOOTNOTE2=	Adds a secondary footnote to the paneled contribution plots
ODSTITLE=	Specifies a title for the paneled contribution plots
ODSTITLE2=	Specifies a secondary title for the paneled contribution plots
TYPE=	Specifies the type of contribution plots produced

You can specify the following *options* in the CONTRIBUTIONPANEL statement. The section "Common Plot Statement Options" on page 913 describes additional options that are available in all plot statements.

MAXNPLOTS=*n*

specifies the maximum number of contribution plots to be produced by the CONTRIBUTIONPANEL statement. The number of plots that are produced is the minimum of *n* and the number of observations in the input data set. When *n* is less than the number of observations, contribution plots are produced for the first *n* observations. The default is 50.

MAXNVARS=*n*

specifies the maximum number of process variable contributions to be displayed in the paneled layout. The magnitudes for each contribution are summed over the observations to be plotted, and the *n* contributions with the greatest total magnitudes are displayed. Therefore each plot displays contributions for the same process variables. By default, all contributions are displayed.

NCOLS=*c*

specifies the number of columns in the panel layout. See the section "Paneled Contribution Plot Layouts" on page 914 for a description of how the default numbers of columns and rows are calculated.

NROWS=*r*

specifies the number of rows in the panel layout. See the section "Paneled Contribution Plot Layouts" on page 914 for a description of how the default numbers of columns and rows are calculated.

TYPE=SPE | TSQUARE

specifies the type of contribution plots displayed. If you specify TYPE=SPE, the contribution plots are based on the SPE statistics; if you specify TYPE=TSQUARE, the contribution plots are based on the T^2 statistics. You must specify a LOADINGS= data set to create T^2 contribution plots. By default, TYPE=TSQUARE if a LOADINGS= data set is provided and TYPE=SPE otherwise.

CONTRIBUTIONPLOT Statement

CONTRIBUTIONPLOT < / *options* > **;**

The CONTRIBUTIONPLOT statement produces a contribution plot for each observation in the input data set, up to a maximum that is specified by the MAXNPLOTS= option. Each contribution plot is displayed as a separate graph. You can use the CONTRIBUTIONPANEL statement to display multiple contribution plots in a paneled layout.

Table 12.2 summarizes the *options* available in the CONTRIBUTIONPLOT statement.

Table 12.2 CONTRIBUTIONPLOT Statement Options

Option	Description
MAXNPLOTS=	Specifies the maximum number of contribution plots displayed
MAXNVARS=	Specifies the maximum number of process variable contributions displayed in each plot
ODSFOOTNOTE=	Adds a footnote to the contribution plots
ODSFOOTNOTE2=	Adds a secondary footnote to the contribution plots
ODSTITLE=	Specifies a title for the contribution plots
ODSTITLE2=	Specifies a secondary title for the contribution plots
TYPE=	Specifies the type of contribution plots produced

You can specify the following *options* in the CONTRIBUTIONPLOT statement. The section "Common Plot Statement Options" on page 913 describes additional options that are available in all plot statements.

MAXNPLOTS=*n*

specifies the maximum number of contribution plots to be produced by the CONTRIBUTIONPLOT statement. The number of plots that are produced is the minimum of *n* and the number of observations in the input data set. When *n* is less than the number of observations, contribution plots are produced for the first *n* observations. The default is 50.

MAXNVARS=*n*

specifies that only the *n* contributions that have the greatest magnitudes be displayed in each plot. The contributions are ranked independently for each plot, so different process variable contributions might be displayed in different plots. By default, all contributions are displayed.

TYPE=SPE | TSQUARE

specifies the type of contribution plot to be created. The option TYPE=TSQUARE specifies that the contribution plots be based on the T^2 statistics. The option TYPE=SPE specifies that the contribution plots be based on the SPE statistics. By default, TYPE=TSQUARE if a LOADINGS= data set is provided and TYPE=SPE otherwise. You can use more than one CONTRIBUTIONPLOT statement. You must specify a LOADINGS= data set to create T^2 contribution plots.

ID Statement

ID *variables* **;**

The first ID *variable* that is specified provides the labels for points in score plots. The values of all the ID variables are displayed in tooltips associated with points in a score plot when you create HTML output and specify the IMAGEMAP option in the ODS GRAPHICS statement. See Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*), for details.

SCOREMATRIX Statement

SCOREMATRIX < */ options* > **;**

The SCOREMATRIX statement produces a matrix of score plots, each of which is a scatter plot of scores for a pair of principal components. You can use the SCOREPLOT statement to display a single score plot in a graph by itself.

Table 12.3 summarizes the *options* available in the SCOREMATRIX statement.

Table 12.3 SCOREMATRIX Statement Options

Option	Description
ALPHA=	Specifies the α value for prediction ellipses
ELLIPSE	Requests prediction ellipses to be overlaid on score plots
GROUP=	Specifies a variable for grouping points in score plots
LABELS=	Specifies whether points in the score plots are labeled
NCOMP=	Specifies the number of principal components whose scores are plotted
ODSFOOTNOTE=	Adds a footnote to the score matrix
ODSFOOTNOTE2=	Adds a secondary footnote to the score matrix
ODSTITLE=	Specifies a title for the score matrix
ODSTITLE2=	Specifies a secondary title for the score matrix

You can specify the following *options* in the SCOREMATRIX statement. The section "Common Plot

Statement Options" on page 913 describes additional options that are available in all plot statements.

ALPHA=*α*
specifies the α value for prediction ellipses that are overlaid on the score plots. The probability that a new observation falls outside the ellipse is α. The default is 0.05. If you specify the ALPHA= option, you do not need to specify the ELLIPSE option.

ELLIPSE
requests that prediction ellipses be overlaid on the score plots. The probability that a new observation falls outside the ellipse is specified by the ALPHA= option.

GROUP=*variable*
specifies a *variable* in the input data set that is used to group the points in the score plots. Points that have different GROUP= values are plotted using different markers or colors (or both) to distinguish the groups.

LABELS=ON | OFF
specifies whether points in the score plots are labeled. Points are labeled with the values of the first variable listed in the ID statement, or the observation number if no ID statement is specified. The default is LABELS=OFF.

NCOMP=*n*
specifies the number of principal components whose scores are plotted in the matrix. The principal components that are plotted are always 1 through *n*. By default, the matrix contains score plots for all principal components.

SCOREPLOT Statement

SCOREPLOT < / *options* > **;**

The SCOREPLOT statement produces a single score plot, which is a scatter plot of the scores that are associated with a pair of principal components. You can use the SCOREMATRIX statement to display a matrix of score plots for more than two principal components.

Table 12.4 summarizes the *options* available in the SCOREPLOT statement.

Table 12.4 SCOREPLOT Statement Options

Option	Description
ALPHA=	Specifies the α value for the prediction ellipse
ELLIPSE	Requests a prediction ellipse to be overlaid on the score plot
GROUP=	Specifies a variable for grouping points in the score plot
LABELS=	Specifies which points in the score plot are to be labeled
ODSFOOTNOTE=	Adds a footnote to the score matrix
ODSFOOTNOTE2=	Adds a secondary footnote to the score matrix
ODSTITLE=	Specifies a title for the score matrix
ODSTITLE2=	Specifies a secondary title for the score matrix
XCOMP=	Specifies the principal component whose scores are plotted on the horizontal axis

Table 12.4 *(continued)*

Option	Description
YCOMP=	Specifies the number of principal components whose scores are plotted on the vertical axis

You can specify the following *options* in the SCOREPLOT statement. The section "Common Plot Statement Options" on page 913 describes additional options that are available in all plot statements.

ALPHA=*α*

specifies the α value for a prediction ellipse that is overlaid on the score plot. The probability that a new observation falls outside the ellipse is α. The default is 0.05. If you specify the ALPHA= option, you do not need to specify the ELLIPSE option.

ELLIPSE

requests that a prediction ellipse be overlaid on the principal component score plot. The probability that a new observation falls outside the ellipse is specified by the ALPHA= option.

GROUP=*variable*

specifies a *variable* in the input data set that is used to group the points in the score plot. Points that have different GROUP= values are plotted with different markers or colors (or both) to distinguish the groups.

LABELS=ON | OFF | OUTSIDE

specifies which points in the score plot to label. Points are labeled with the values of the first variable listed in the ID statement, or the observation number if no ID statement is specified. By default, LABELS=ON and all points are labeled. You can specify LABELS=OFF to suppress all point labels.

If you overlay a prediction ellipse on the score plot by specifying the ELLIPSE or ALPHA= option, you can specify LABELS=OUTSIDE to label only the points outside the prediction ellipse.

XCOMP=*x*

specifies an integer *x* that identifies the principal component whose scores are plotted on the horizontal axis of the score plot. The default is 1. You cannot specify the same principal component number in both the XCOMP= and YCOMP= options.

YCOMP=*y*

specifies an integer *y* that identifies the principal component whose scores are plotted on the vertical axis of the score plot. The default is $\mathrm{mod}(x, j) + 1$, where *x* is the value that is specified by the XCOMP= option and *j* is the number of principal components in the model. You cannot specify the same principal component number in both the XCOMP= and YCOMP= options.

TIME Statement

TIME *variable* **;**

The TIME *variable* is a numeric variable that provides the chronological order or time values for measurements in the input data set. The *variable* name and value are incorporated into contribution plot titles to identify the

observations represented. If you do not specify a TIME variable, the observation number from the input data set is used instead.

Common Plot Statement Options

You can specify the following *options* after a slash (/) in the CONTRIBUTIONPANEL, CONTRIBUTIONPLOT, SCOREMATRIX, and SCOREPLOT statements.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string*'
adds a footnote to the plot. If you specify the FOOTNOTE (or FOOTNOTE1) keyword, the value of the SAS FOOTNOTE statement is used as the plot footnote. If you specify a quoted string, that string is used as the footnote. The quoted string can contain the following escaped characters, which are replaced by the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the analysis TIME has no label).

ODSFOOTNOTE2=FOOTNOTE2 | '*string*'
adds a secondary footnote to the plot. If you specify the FOOTNOTE2 keyword, the value of the SAS FOOTNOTE2 statement is used as the secondary plot footnote. If you specify a quoted string, that string is used as the secondary footnote. The quoted string can contain the following escaped characters, which are replaced by the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the TIME variable has no label).

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | '*string*'
specifies a title for the plot. You can specify the following values:

TITLE (or TITLE1)	uses the value of the SAS TITLE statement as the plot title.
NONE	suppresses all titles from the plot.
DEFAULT	uses the default title.

If you specify a quoted string, that string is used as the graph title. The quoted string can contain the following escaped characters, which are replaced by the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the analysis variable has no label).

ODSTITLE2=TITLE2 | '*string*'
specifies a secondary title for the plot. If you specify the TITLE2 keyword, the value of the SAS TITLE2 statement is used as the secondary plot title. If you specify a quoted string, that string is used as the secondary title. The quoted string can contain the following escaped characters, which are replaced by the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the analysis variable has no label).

Details: MVPDIAGNOSE Procedure

Contribution Plots

One way to diagnose the behavior of out-of-control points in multivariate control charts is to use contribution plots (Miller, Swanson, and Heckler 1998). These plots tell you which variables contribute to the distance between the points in an SPE or T^2 chart and the sample mean of the data.

A contribution plot is a bar chart of the contributions of the process variables to the statistic. For the ith SPE statistic, the contribution of the kth variable is the kth entry of the vector $\mathbf{e}_i$, which is computed as

$$\mathbf{e}_i = \mathbf{x}_i \left(\mathbf{I} - \mathbf{P}_j \mathbf{P}'_j \right)$$

where $\mathbf{e}_i$ is the vector of errors from the principal component model for observation i and $\mathbf{x}_i$ is the ith observation. The contributions to the ith T^2 statistic are computed in the same way as the entries of the vector

$$\mathbf{T}_i^2 = \mathbf{x}_i \mathbf{P}_j \mathbf{L}^{-1} \mathbf{P}'_j$$

where $\mathbf{P}_j$ is the matrix of the first j eigenvectors and $\mathbf{L}$ is the diagonal matrix of the first j eigenvalues.

Paneled Contribution Plot Layouts

The CONTRIBUTIONPANEL statement produces paneled contribution plots. You can use the options NCOLS=c and NROWS=r to specify the number of columns and rows in the layout, respectively.

By default, c and r are determined by p, the number of contribution plots to be displayed. If $p \leq 16$, then $c = \lceil \sqrt{p} \rceil$ and $r = \lceil p/c \rceil$. Otherwise, one of the following three layouts is used to minimize the number of empty panels in the last page of the graph:

- $c = 4, r = 4$
- $c = 4, r = 3$
- $c = 3, r = 3$

If you specify only NCOLS=c, then $r = \lceil p/c \rceil$. If you specify only NROWS=r, then $c = \lceil p/r \rceil$.

Although $c \leq 4$ and $r \leq 4$ by default, you can specify values greater than 4 in the NCOLS= and NROWS= options.

Input Data Sets

The MVPDIAGNOSE procedure accepts a primary input data set that has one of the following two types:

- a DATA= data set that contains new process data to be analyzed by using an existing principal component model (Phase II analysis)
- a HISTORY= data set that contains process data and the accompanying scores, residuals, and statistics that are produced by using a principal component model. The process data can be the original data that were used to create the model (Phase I analysis) or subsequent data that were analyzed by using a previously created model (Phase II analysis)

These options are mutually exclusive. If you do not specify an option that identifies a primary input data set, PROC MVPDIAGNOSE uses the most recently created SAS data set as a DATA= data set.

When you specify a DATA= data set, you must also specify a LOADINGS= data set that contains principal component loadings and other information that describes the principal component model. When you specify a HISTORY= data set, you must also specify a LOADINGS= data set if you use a CONTRIBUTIONPANEL or CONTRIBUTIONPLOT statement and specify the TYPE=TSQUARE option.

DATA= Data Set

A DATA= data set provides the process measurement data for a Phase II analysis. In addition to containing the process variables, a DATA= data set can contain the following:

- BY variables
- ID variables
- a TIME variable

When you specify a DATA= data set, you must also specify a LOADINGS= data set that contains the loadings for the principal component model that describes the variation of the process. These loadings are used to score the new data from the DATA= data set. The process variables in the LOADINGS= data set must have the same names as those in the DATA= data set.

HISTORY= Data Set

A HISTORY= data set provides the input data set for a Phase I or Phase II analysis. In addition to containing the original process variables, a HISTORY= data set contains principal component scores, residuals, SPE and T^2 statistics, and a count of the observations that are used to construct the principal component model. These variables are summarized in Table 12.5.

Table 12.5 Variables in the HISTORY= Data Set

Variable	Description
Prin1–Prin$j$	Principal component scores
R_var1–R_$varp$	Residuals

Table 12.5 *(continued)*

Variable	Description
NOBS	Number of observations used in the analysis
SPE	Squared prediction error (SPE) statistic
TSQUARE	T^2 statistic computed from principal component scores

The score variables names must consist of a common prefix followed by the numbers 1, 2, . . . , j, where j is the number of principal components. By default, the common prefix is Prin. You can use the PREFIX= option to specify another prefix for score variables.

If the number of principal components is less than the total number of process variables, the HISTORY= data set should also contain residual variables. A residual variable name consists of a common prefix followed by the corresponding process variable name. The default residual variable prefix is R_. For example, if the process variables are A, B, and C, the default residual variable names are R_A, R_B, and R_C. You can use the RPREFIX= option to specify a different residual variable prefix.

NOTE: Usually you create a HISTORY= data set by specifying the OUT= option in the PROC MVPMODEL statement or the OUTHISTORY= option in the PROC MVPMONITOR statement. If the PREFIX= or RPREFIX= option is used when such an output data set is created, you must specify the same prefixes to identify the score and residual variables when you read it as a HISTORY= data set.

LOADINGS= Data Set

A LOADINGS= data set contains the following information about the principal component model:

- eigenvalues of the correlation or covariance matrix used to construct the model
- principal component loadings
- process variable means used to center the variable values
- process variable standard deviations used to scale the variable values

You can produce a LOADINGS= data set by using the OUTLOADINGS= option in the PROC MVPMODEL statement. Table 12.6 lists the variables that are required in a LOADINGS= data set.

Table 12.6 Variables in the LOADINGS= Data Set

Variable	Description
VALUE	The value contained in *process variables* for a given observation
NOBS	Number of observations used to build the principal component model
_PC_	The principal component number; 0 for the observation that contains eigenvalues
process variables	Values associated with the process variables

Valid values for the _VALUE_ variable are as follows:

EIGEN	eigenvalues from the principal component analysis
LOADING	principal component loadings
MEAN	process variable means
STD	process variable standard deviations

The LOADINGS= data set contains one EIGEN observation and *j* LOADING observations, where *j* is the number of principal components in the model. The presence of a MEAN observation indicates that the process variables were centered when the principal component model was built, and the presence of a STD observation indicates that the process variables were scaled when the principal component model was built. The means and standard deviations are used to center and scale new data in a Phase II analysis.

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The MVPDIAGNOSE procedure assigns a name to each graph that it creates. You can use these names to refer to the graphs when you use ODS Graphics. The ODS graph names are listed in Table 12.7.

Table 12.7 ODS Graphics Produced by PROC MVPDIAGNOSE

ODS Graph Name	Plot Description	Statement
ContributionPanel	Paneled contribution plots	CONTRIBUTIONPANEL
ContributionPlot	Contribution plots	CONTRIBUTIONPLOT
ScoreMatrix	Matrix of pairwise score plots	SCOREMATRIX
ScorePlot	Score plot	SCOREPLOT

Examples: MVPDIAGNOSE Procedure

Example 12.1: Phase II Analysis with MVPDIAGNOSE

The example in "Getting Started: MVPDIAGNOSE Procedure" on page 902 illustrates how you build a principal component model and apply the MVPMONITOR and MVPDIAGNOSE procedures to perform a Phase I analysis. In Phase I analysis you analyze the data that were used to build the principal component model. This example is a continuation of that example and illustrates how you can use PROC MVPDIAGNOSE to analyze process data that were not used to build the model. This is called a Phase II analysis. A Phase II analysis is usually performed on data that are collected after the data that are used to build the model.

In the original example the principal component model was built using flight delay data from February 1–16, 2007. The following statements create a data set named MWflightDelays2 that contains average delays for flights that originated in the midwestern United States on February 17–28, 2007:

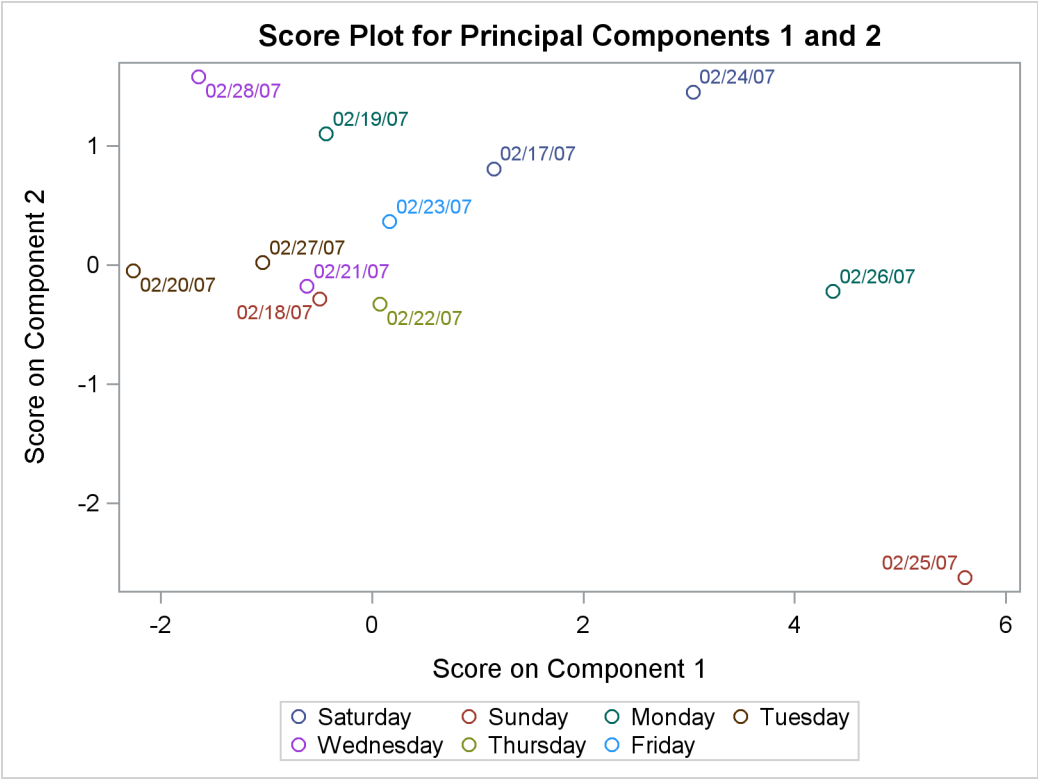
```
data MWflightDelays2;
   label flightDate='Date';
   format flightDate MMDDYY8.;
   input flightDate :MMDDYY8. AA CO DL F9 FL NW UA US WN;
   dayofweek = put(flightDate,downame.);
   datalines;
02/17/07 25.6  7.8 15.5 13.4 16.1 16.2 23.0 24.2  8.2
02/18/07  5.4 16.0  9.9  1.1 11.5 17.0 15.6 15.5  5.1
02/19/07 13.2 16.3 10.0 10.6  5.4 10.3  9.5 16.8  9.3
02/20/07  4.2  6.9  1.4  0.1  7.2  6.6  7.4 10.4  2.9
02/21/07  5.4 -0.1  7.4  8.7 16.3 24.3  9.4  6.0 10.2
02/22/07 19.6 30.2  6.8  2.7  8.9 16.4 14.3 12.6  8.2
02/23/07 14.9 18.9  9.9  9.1 12.0 16.5 17.4 12.8  6.0
02/24/07 21.4  5.5 11.1 46.1 10.6 55.3 22.9  8.8  3.4
02/25/07 42.6  7.7 14.6 14.4 32.0 50.7 46.1 49.4 39.1
02/26/07 43.2 25.1 18.1 18.2 28.8 31.1 38.6 29.6 18.6
02/27/07 11.3 17.1  5.3  4.1  4.8 13.9  9.8  9.7  7.1
02/28/07  8.1  3.7  2.7 17.1 -0.8  5.5 11.0 14.3  3.1
;
```

The dayofweek variable contains the day of the week for each date in the input data set. The following statements apply the model that is saved in the mvpairloadings data set to the flightDelays2 data and produce a score plot for the first two principal components:

```
proc mvpdiagnose data=MWflightDelays2 loadings=mvpairloadings;
   id flightDate;
   scoreplot / labels=on group=dayofweek;
   label dayofweek='';
run;
```

The ID statement labels the points in the score plot with flightDate variable values. The GROUP= option displays the observations grouped by day of the week. The LABEL statement suppresses the GROUP= legend label. Figure 12.1.1 shows the score plot.

Output 12.1.1 Score Plot with Observations Grouped by Day of the Week

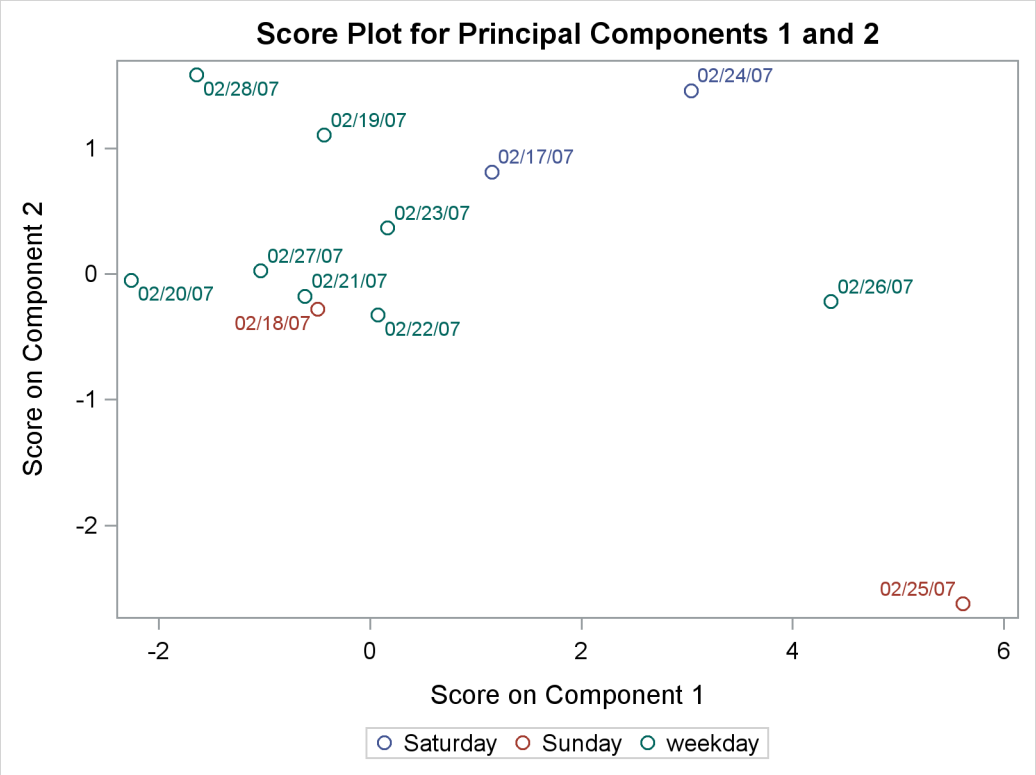


The Saturday and Sunday observations seem to be divided by scores for principal component 1. The following statements modify the dayofweek values to merge the observations for the other days into a "weekday" group:

```
data MWflightDelays2;
   set MWflightDelays2;
   weekday = put(flightDate,weekday.);
   if not ( weekday in (1 7) ) then dayofweek='weekday';
run;
```

Figure 12.1.2 shows the score plot that is produced by running PROC MVPDIAGNOSE with this new grouping. Merging the weekday observations into a single group emphasizes the Saturday and Sunday scores.

Output 12.1.2 Score Plot with Alternate Grouping



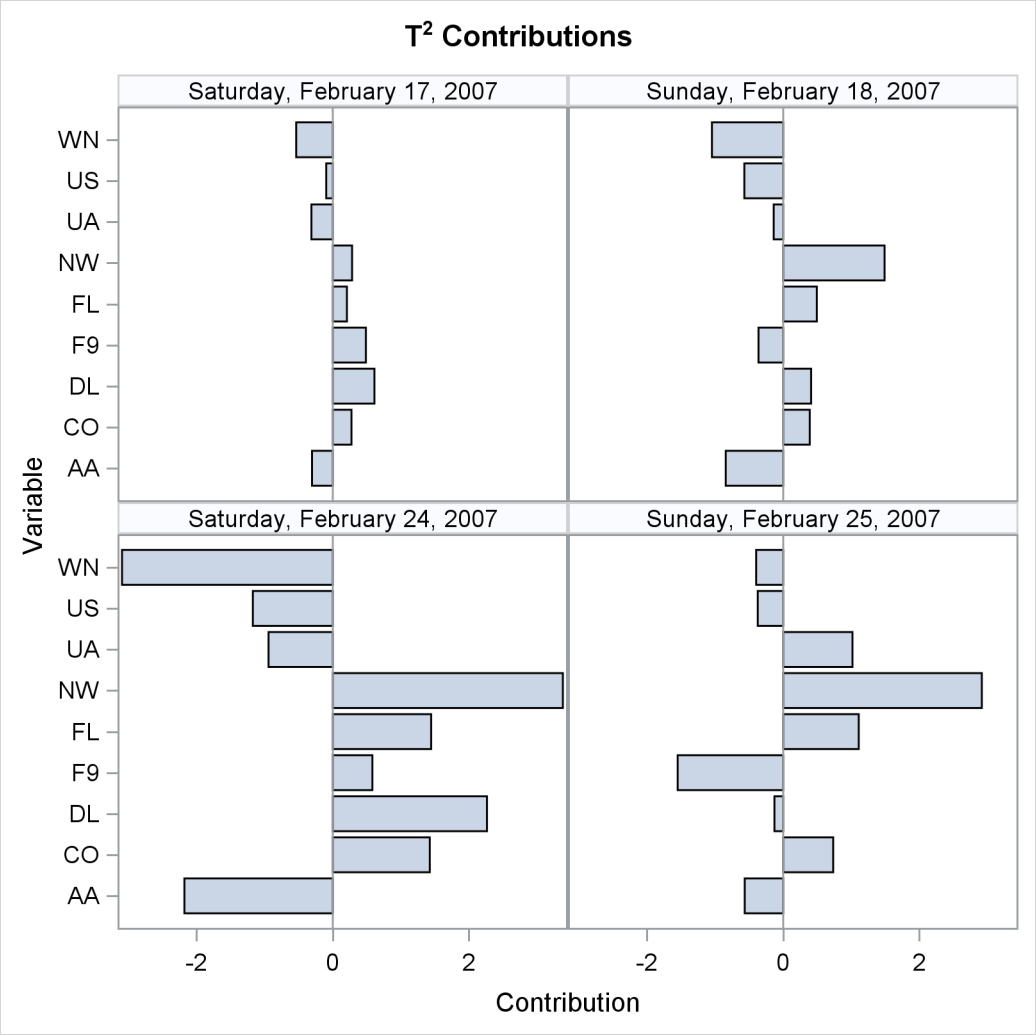
Because the score plot shows that something interesting might be happening on the weekends, you might want to examine contribution plots for those days. The following statements produce paneled T^2 and SPE contribution plots for the weekend observations:

```
proc mvpdiagnose data=MWflightDelays2 loadings=mvpairloadings;
   where dayofweek ne 'weekday';
   time flightDate;
   contributionpanel;
   contributionpanel / type=spe;
   format flightDate weekdate.;
run;
```

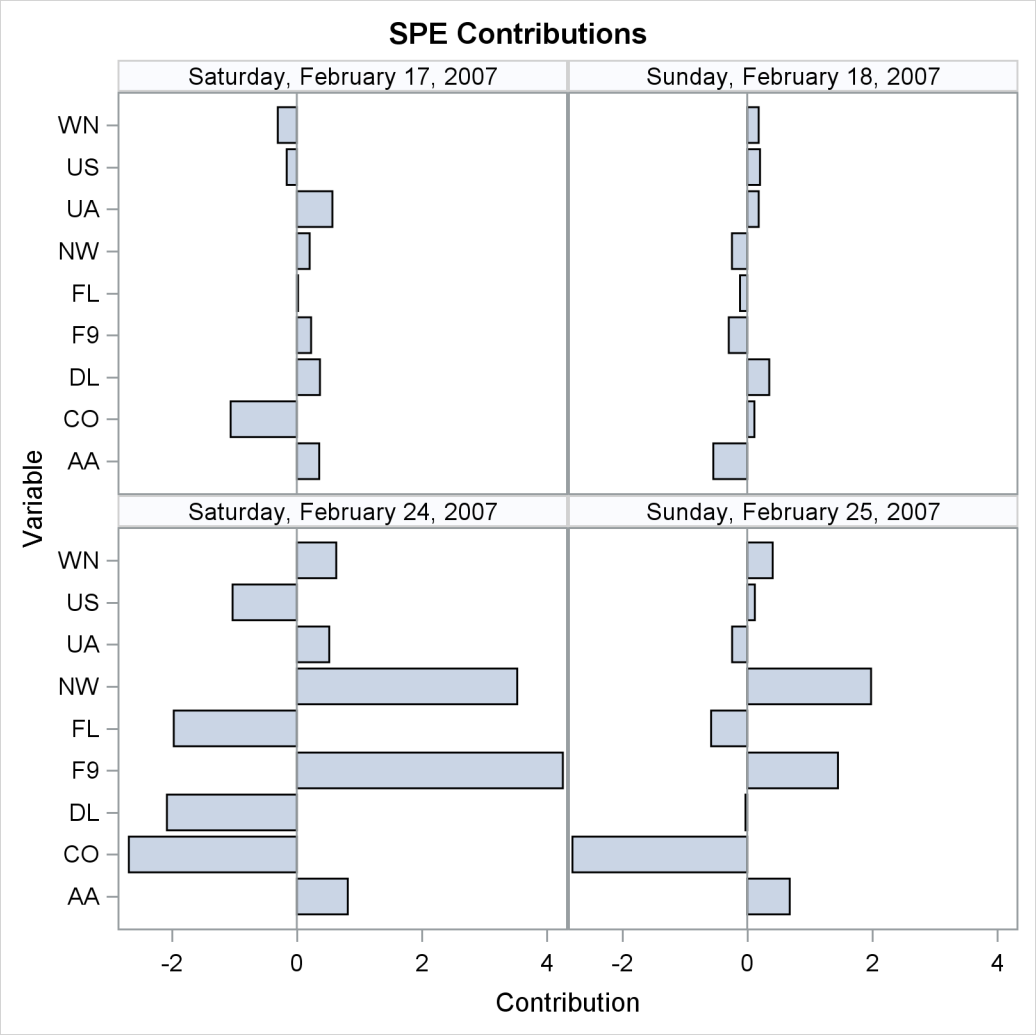
Because the CONTRIBUTIONPANEL statement produces contribution plots for a series of observations starting with the first observation in the input data set, it is convenient to use a WHERE statement to select observations of interest.

Figure 12.1.3 and Figure 12.1.4 show the paneled contribution plots.

Output 12.1.3 T^2 Contribution Plots for Weekends



Output 12.1.4 SPE Contribution Plots for Weekends



References

Alt, F. (1985). "Multivariate Quality Control." In *Encyclopedia of Statistical Sciences,* vol. 6, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons.

Cooley, W. W., and Lohnes, P. R. (1971). *Multivariate Data Analysis*. New York: John Wiley & Sons.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. New York: John Wiley & Sons.

Hotelling, H. (1933). "Analysis of a Complex of Statistical Variables into Principal Components." *Journal of Educational Psychology* 24:417–441, 498–520.

Jackson, J. E., and Mudholkar, G. S. (1979). "Control Procedures for Residuals Associated with Principal Component Analysis." *Technometrics* 21:341–349.

Jensen, D. R., and Solomon, H. (1972). "A Gaussian Approximation to the Distribution of a Definite Quadratic Form." *Journal of the American Statistical Association* 67:898–902.

Kourti, T., and MacGregor, J. F. (1995). "Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods." *Chemometrics and Intelligent Laboratory Systems* 28:3–21.

Kourti, T., and MacGregor, J. F. (1996). "Multivariate SPC Methods for Process and Product Monitoring." *Journal of Quality Technology* 28:409–428.

Kshirsagar, A. M. (1972). *Multivariate Analysis*. New York: Marcel Dekker.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. London: Academic Press.

Miller, P., Swanson, R. E., and Heckler, C. H. E. (1998). "Contribution Plots: A Missing Link in Multivariate Quality Control." *Applied Mathematics and Computer Science* 8:775–792.

Morrison, D. F. (1976). *Multivariate Statistical Methods*. 2nd ed. New York: McGraw-Hill.

Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space." *Philosophical Magazine* 6:559–572.

Rao, C. R. (1964). "The Use and Interpretation of Principal Component Analysis in Applied Research." *Sankhyā, Series A* 26:329–358.

Wilks, S. S. (1962). *Mathematical Statistics*. New York: John Wiley & Sons.

Chapter 13
The MVPMODEL Procedure

Contents

Overview: MVPMODEL Procedure **925**
- Using the MVP Procedures 926
- Functionality of the MVPMODEL Procedure 926

Getting Started: MVPMODEL Procedure **927**

Syntax: MVPMODEL Procedure **934**
- PROC MVPMODEL Statement 935
- BY Statement 941
- ID Statement 941
- VAR Statement 941

Details: MVPMODEL Procedure **942**
- Classical T^2 Charts 942
- Principal Component Analysis 942
- Relationship of Principal Components to Multivariate Control Charts 943
- Cross Validation (Experimental) 945
- Centering and Scaling 946
- Missing Values 946
- Input Data Set 946
- Output Data Sets 947
- ODS Table Names 948
- ODS Graphics 948

Examples: MVPMODEL Procedure **949**
- Example 13.1: Using Cross Validation to Select the Number of Principal Components 949
- Example 13.2: Computing the Classical T^2 Statistic 952

References **954**

Overview: MVPMODEL Procedure

The MVPMODEL procedure is used in conjunction with the MVPMONITOR and MVPDIAGNOSE procedures to monitor multivariate process variation over time, to determine whether the process is stable, and to detect and diagnose changes in a stable process. Collectively these three procedures are referred to as the *MVP procedures*. See Chapter 11, "Introduction to Multivariate Process Monitoring Procedures," for a description of how the MVP procedures work together, and Chapter 14, "The MVPMONITOR Procedure," and Chapter 12, "The MVPDIAGNOSE Procedure," for details about the other MVP procedures.

The MVPMODEL procedure provides computational and graphical tools for building a principal component model from multivariate process data in which the measured variables are continuous and correlated. This model then serves as input to the other MVP procedures, described in Chapter 12, "The MVPDIAGNOSE Procedure," and Chapter 14, "The MVPMONITOR Procedure." The MVPMONITOR procedure creates various multivariate control charts, including T^2 charts and SPE (squared prediction error) charts, which are used to detect and diagnose changes in the process. Multivariate control charts can detect unusual variation that would not be detected by individually monitoring the variables with univariate control charts, such as Shewhart charts.

The MVPMODEL procedure implements principal component analysis (PCA) techniques that evolved in the field of chemometrics for monitoring hundreds or even thousands of correlated process variables; see Kourti and MacGregor (1995, 1996) for an introduction. These techniques differ from the classical multivariate T^2 chart in which Hotelling's T^2 statistic is computed as a distance from the multivariate mean scaled by the covariance matrix of the variables; see Alt (1985). Instead, principal component methods compute T^2 based on a small number of principal components that model most of the variation in the data.

One advantage of PCA methods over the classical T^2 chart is that they avoid computational issues that arise when the process measurement variables are collinear and their covariance matrix is nearly singular. A second advantage is that they offer diagnostic tools for interpreting unusual values of T^2. A third advantage is that by projecting the data to a low-dimensional subspace, a principal component model more adequately describes the variation in a multivariate process, which is often driven by a small number of underlying factors that are not directly observable.

Using the MVP Procedures

There are two primary scenarios for using the MVP procedures:

1. To determine whether a process is stable, you can construct T^2 and SPE charts from an existing set of process measurements (this is referred to as a Phase I analysis). First, build a principal component model with the MVPMODEL procedure, saving the measurements and the computed observationwise statistics (including T^2 and SPE) in an OUT= data set. Then specify this data set as a HISTORY= input data set for the MVPMONITOR procedure to create T^2 and SPE charts. Contribution plots indicate which of the original variables are involved in unusual variation displayed by the T^2 and SPE charts. Follow-up action might be needed to adjust the process and eliminate unusual variation signaled by the charts.

2. To detect changes in a stable process, you can construct T^2 and SPE charts from newly acquired data by using the principal component model developed from previous data (this is referred to as a Phase II analysis). You can save information about the model in the OUTLOADINGS= data set created by the MVPMODEL procedure. Specify this data set as a LOADINGS= input data set and specify the new data as a DATA= input data set to create T^2 and SPE charts with the MVPMONITOR procedure.

Functionality of the MVPMODEL Procedure

The MVPMODEL procedure performs principal component analysis (PCA) on multivariate process measurement data that consist of p continuous variables that are assumed to be correlated. The input data set for

PROC MVPMODEL provides the values of the p variables that are to be analyzed.

The MVPMODEL procedure computes the following quantities:

- the loadings from the principal component analysis
- the eigenvalues from the principal component analysis, which are the variances of the principal component variables
- the scores from the principal component analysis
- the T^2 statistic for each observation
- the SPE (squared prediction error) statistic for each observation, also known as SSE, Q, or DModX

By default, principal components are computed from the correlation matrix of the variables. Optionally, they can be computed from their covariance matrix instead. The number of principal components in the model (denoted by j, where $j \leq p$) can be specified or determined by one of several cross validation methods.

By default, PROC MVPMODEL outputs the correlation matrix of the input variables and the eigenvalues of the correlation matrix. When ODS Graphics is enabled, the output can also include the following plots:

- a scree plot and a variance-explained plot of the principal components (these plots are created by default)
- when using cross validation, plots of W and root mean PRESS (predicted residual sum of squares) for each principal component
- pairwise score plots of principal component scores
- pairwise loading plots of principal component loadings

PROC MVPMODEL saves information about the principal component model in the following two output data sets, which can subsequently serve as inputs to the MVPMONITOR and MVPDIAGNOSE procedures:

- an output data set which contains all the variables and observations in the input data set together with observationwise statistics, such as scores, residuals, T^2, and SPE
- an output data set that contains the j loadings for each process variable and the eigenvalues associated with each of the principal components

Getting Started: MVPMODEL Procedure

This example illustrates the basic features of the MVPMODEL procedure by using airline flight delay data available from the U.S. Bureau of Transportation Statistics at `http://www.transtats.bts.gov`. The example applies multivariate process monitoring to flight delays.

Suppose you want to use a principal component model to create T^2 and SPE charts to monitor the variation in flight delays. These charts are appropriate because the data are multivariate and correlated.

The following statements create a SAS data set named MWflightDelays to contain the average flight delays for flights that originate in the midwestern United States by airline. The data set contains variables for nine airlines: AA (American Airlines), CO (Continental Airlines), DL (Delta Airlines), F9 (Frontier Airlines), FL (AirTran Airways), NW (Northwest Airlines), UA (United Airlines), US (US Airways), and WN (Southwest Airlines).

```
data MWflightDelays;
   format flightDate MMDDYY8.;
   label flightDate='Date';
   input flightDate :MMDDYY8. AA CO DL F9 FL NW UA US WN;
   datalines;
02/01/07 14.9  7.1  7.9  8.5 14.8  4.5  5.1 13.4  5.1
02/02/07 14.3  9.6 14.1  6.2 12.8  6.0  3.9 15.3 11.4
02/03/07 23.0  6.1  1.7  0.9 11.9 15.2  9.5 18.4  7.6
02/04/07  6.5  6.3  3.9 -0.2  8.4 18.8  6.2  8.8  8.0
02/05/07 12.0 14.1  3.3 -1.3 10.0 13.1 22.8 16.5 11.5
02/06/07 31.9  8.6  4.9  2.0 11.9 21.9 29.0 15.5 15.2
02/07/07 14.2  3.0  2.1 -0.9 -0.6  7.8 19.9  8.6  6.4
02/08/07  6.5  6.8  1.8  7.7  1.3  6.9  6.1  9.2  5.4
02/09/07 12.8  9.4  5.5  9.3 -0.2  4.6  7.6  7.8  7.5
02/10/07  9.4  3.5  1.5 -0.2  2.2  9.9  3.1 12.5  3.0
02/11/07 12.9  5.4  0.9  6.8  2.1  7.9  3.7 10.7  5.6
02/12/07 34.6 15.9  1.8  1.0  4.5 10.2 14.0 19.1  4.9
02/13/07 34.0 16.0  4.4  6.1 18.3  9.1 30.2 46.3 50.6
02/14/07 21.2 45.9 16.6 12.5 35.1 23.8 40.4 43.6 35.2
02/15/07 46.6 36.3 23.9 20.8 30.4 24.3 30.3 59.9 25.6
02/16/07 31.2 20.8 15.2 20.1  9.1 12.9 22.9 36.4 16.4
;
```

The observations for a given date are the average flight delays in minutes of flights that depart from the Midwest. For example, on February 2, 2007, F9 (Frontier Airlines) flights departed an average of 6.2 minutes late.

Preliminary Analysis

The following statements use the MVPMODEL procedure to conduct a preliminary principal component analysis:

```
ods graphics on;
proc mvpmodel data=MWflightDelays;
   var AA CO DL F9 FL NW UA US WN;
run;
```

The DATA= option specifies the input data set, which contains the process measurement variables. The VAR statement specifies the process measurement variables to be analyzed. The ODS GRAPHICS ON statement enables ODS Graphics, which is used to produce plots for interpreting the model.

The procedure first outputs a summary of the model and the data, as shown in Figure 13.1.

Figure 13.1 Summary of Model and Data Information

The MVPMODEL Procedure

Data Set	WORK.MWFLIGHTDELAYS
Number of Variables	9
Missing Value Handling	Exclude
Number of Observations Read	16
Number of Observations Used	16
Number of Principal Components	9

This output includes the number of principal components in the model and the number of variables. In this case the procedure produces a model with nine principal components by default, because there are nine process variables.

Next, the procedure outputs the correlation matrix shown in Figure 13.2.

Figure 13.2 Correlation Matrix

Correlation Matrix

	AA	CO	DL	F9	FL	NW	UA	US	WN
AA	1.0000	0.5640	0.5206	0.4874	0.5403	0.4860	0.6466	0.7856	0.5506
CO	0.5640	1.0000	0.7855	0.6580	0.8519	0.6421	0.7672	0.8415	0.6526
DL	0.5206	0.7855	1.0000	0.8231	0.7598	0.4782	0.4951	0.7463	0.4525
F9	0.4874	0.6580	0.8231	1.0000	0.5119	0.2279	0.3509	0.6832	0.3914
FL	0.5403	0.8519	0.7598	0.5119	1.0000	0.6807	0.6975	0.8207	0.7186
NW	0.4860	0.6421	0.4782	0.2279	0.6807	1.0000	0.6715	0.5598	0.3970
UA	0.6466	0.7672	0.4951	0.3509	0.6975	0.6715	1.0000	0.7540	0.7736
US	0.7856	0.8415	0.7463	0.6832	0.8207	0.5598	0.7540	1.0000	0.8152
WN	0.5506	0.6526	0.4525	0.3914	0.7186	0.3970	0.7736	0.8152	1.0000

There are strong correlations (greater than 0.8) between variable pairs F9 and DL, CO and FL, and US and WN. This is not surprising, because these pairs of airlines have closely located hubs or focus cities.

The procedure also outputs the eigenvalue and variance information shown in Figure 13.3.

Figure 13.3 Eigenvalue and Variance Information

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	6.09006397	5.02872938	0.6767	0.6767
2	1.06133459	0.36642409	0.1179	0.7946
3	0.69491050	0.16102099	0.0772	0.8718
4	0.53388951	0.28357563	0.0593	0.9311
5	0.25031387	0.09537517	0.0278	0.9589
6	0.15493870	0.03339131	0.0172	0.9762
7	0.12154739	0.06166364	0.0135	0.9897
8	0.05988375	0.02676604	0.0067	0.9963
9	0.03311771		0.0037	1.0000

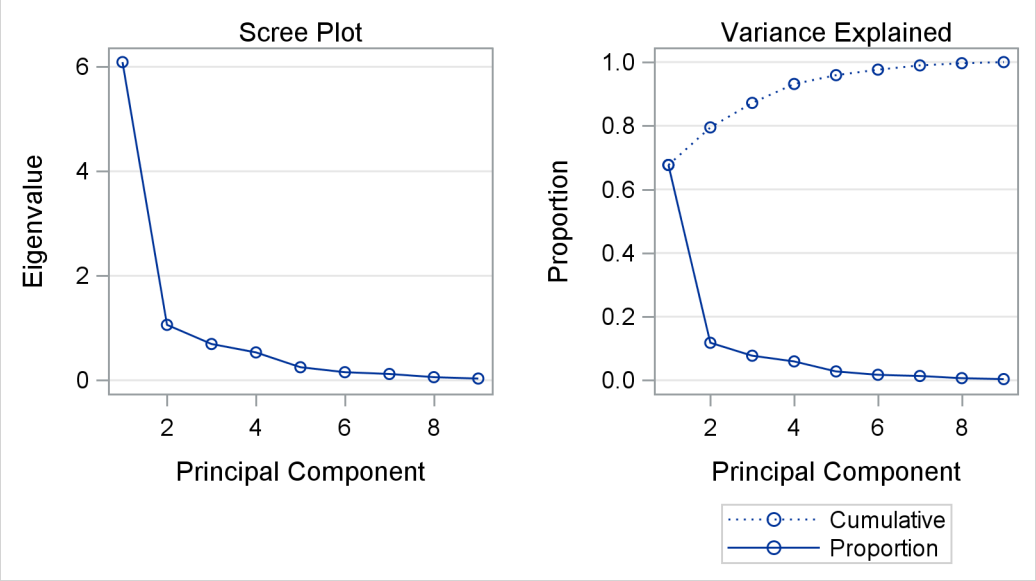
The eigenvalues are the variances of the principal components, and the proportions reflect the relative amount

of variance explained by each component. The eigenvalues and the proportions are ordered from largest to smallest. Recall that principal components are orthogonal linear combinations of the variables that maximize variance in orthogonal directions.

More than 85% of the variance is explained by the first three principal components, as shown in the cumulative variance column. This suggests that a model with three principal components is adequate; this is confirmed by the plots in Figure 13.4.

Figure 13.4 shows a paneled display, with a scree plot in the left panel and a variance-explained plot in the right panel.

Figure 13.4 Scree Plot and Variance-Explained Plot



The scree plot shows the eigenvalues for each principal component. Traditionally, the scree plot has been recommended as an aid in selecting the number of principal components for the model by examining the "knee" in the plot (Mardia, Kent, and Bibby 1979). The variance-explained plot shows both the proportion of variance and the cumulative variance explained by the principal components.

Building a Principal Component Model

To build a model that has only three principal components, you can use the NCOMP= option as shown in the following statements:

```
proc mvpmodel data=MWflightDelays ncomp=3 plots=(all score(labels=on))
              out=outDelays;
   var AA CO DL F9 FL NW UA US WN;
run;
```

The PLOTS=ALL option requests all possible plots, which include pairwise plots of the principal component scores and loadings in addition to the default scree plot and variance-explained plot. The OUT= option produces an output data set called outDelays that contains principal component scores, T^2 statistics, SPE statistics, residuals, and more, as described in the section "Output Data Sets" on page 947. Note that ODS Graphics is still enabled, so you do not need to specify the ODS GRAPHICS ON statement here.

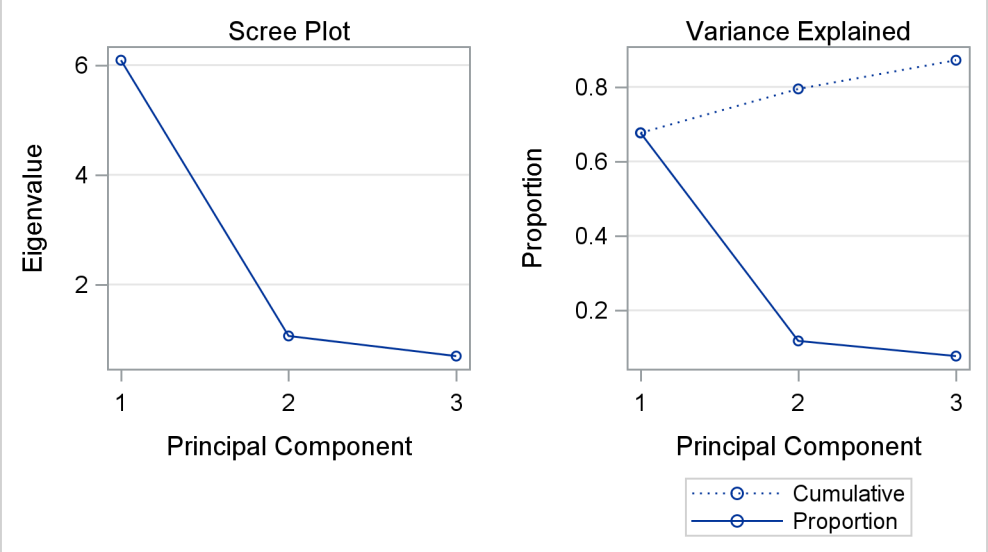
The correlation matrix is the same as in Figure 13.2. The eigenvalue information, scree plot, and variance-explained plot are similar to those in Figure 13.3 and Figure 13.4. However, the use of the NCOMP=3 option results in outputs that show information only for the three components in the model, as seen in Figure 13.5 and Figure 13.6.

Figure 13.5 Eigenvalue and Variance Information

The MVPMODEL Procedure

Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	6.09006397	5.02872938	0.6767	0.6767
2	1.06133459	0.36642409	0.1179	0.7946
3	0.69491050		0.0772	0.8718

Figure 13.6 Scree Plot and Variance-Explained Plot



Also, the model summary, shown in Figure 13.7, is different because there are now only three principal components in the model.

Figure 13.7 Summary of Model and Data Information

The MVPMODEL Procedure

Data Set	WORK.MWFLIGHTDELAYS
Number of Variables	9
Missing Value Handling	Exclude
Number of Observations Read	16
Number of Observations Used	16
Number of Principal Components	3

The outDelays output data set that is partially listed in Figure 13.8 contains T^2 and SPE statistics based on

the model that has three principal components, in addition to the original variables and other observationwise statistics.

Figure 13.8 Partial Listing of Output Data Set outDelays

flightDate	AA	CO	DL	F9	FL	NW	UA	US	WN	Prin1	Prin2	Prin3	_NOBS_	_TSQUARE_
02/01/07	14.9	7.1	7.9	8.5	14.8	4.5	5.1	13.4	5.1	-1.08708	1.20953	-0.03839	16	1.57457
02/02/07	14.3	9.6	14.1	6.2	12.8	6.0	3.9	15.3	11.4	-0.65786	1.26249	0.11447	16	1.59169
02/03/07	23.0	6.1	1.7	0.9	11.9	15.2	9.5	18.4	7.6	-0.86457	-0.73183	0.29270	16	0.75065
02/04/07	6.5	6.3	3.9	-0.2	8.4	18.8	6.2	8.8	8.0	-1.50578	-0.69718	1.32511	16	3.35709
02/05/07	12.0	14.1	3.3	-1.3	10.0	13.1	22.8	16.5	11.5	-0.63903	-1.11141	0.38617	16	1.44549

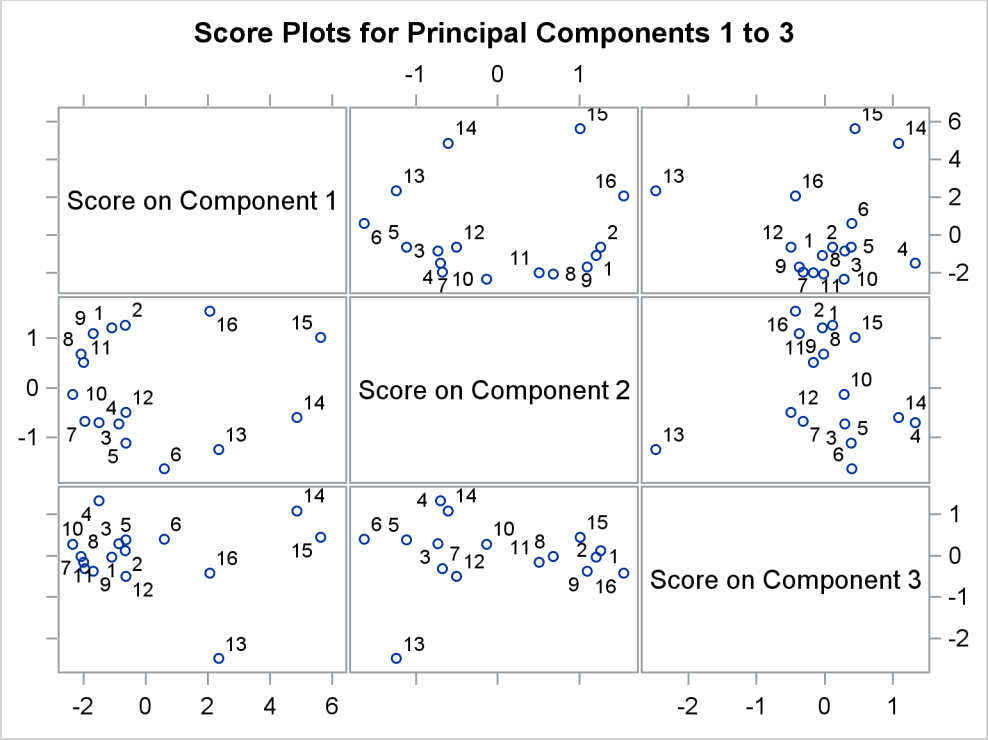
R_AA	R_CO	R_DL	R_F9	R_FL	R_NW	R_UA	R_US	R_WN	_SPE_
-0.05779	-0.18178	-0.01835	-0.15280	0.87457	-0.37864	-0.06037	-0.12896	-0.02300	0.98911
-0.17802	-0.16663	0.68047	-0.62682	0.49289	-0.35101	-0.27027	-0.14161	0.41169	1.54414
0.54274	-0.30297	-0.20552	-0.02408	0.31360	0.21270	-0.49772	0.24287	-0.23829	0.93626
-0.25729	-0.24974	0.05624	0.05279	-0.02427	0.29305	-0.44076	0.13493	0.50899	0.69253
-0.44128	0.28274	0.09998	-0.15050	0.00176	-0.36866	0.39124	0.07233	-0.06265	0.60545

The variables Prin1, Prin2, and Prin3 contain the principal component scores. Variables R_AA through R_WN are the residuals for the process variables. The contents of an OUT= data set are described in detail in the section "Output Data Sets" on page 947. See the section "Principal Component Analysis" on page 942 for computational details of the results saved in the output data set.

You can use an OUT= data set as an input to the MVPMONITOR and MVPDIAGNOSE procedures. The MVPMONITOR procedure produces control charts for the T^2 and SPE statistics. Control charts that are created from the outDelays data set are shown in Example 13.2 and in the MVPMONITOR procedure chapter.

The PLOTS=ALL option produces score plots for pairs of principal components in the model. By default, the score plots are displayed in a matrix. You can specify the PLOTS(SCORES(UNPACK)) option to display the score plots as separate graphs. The score plot matrix is shown in Figure 13.9.

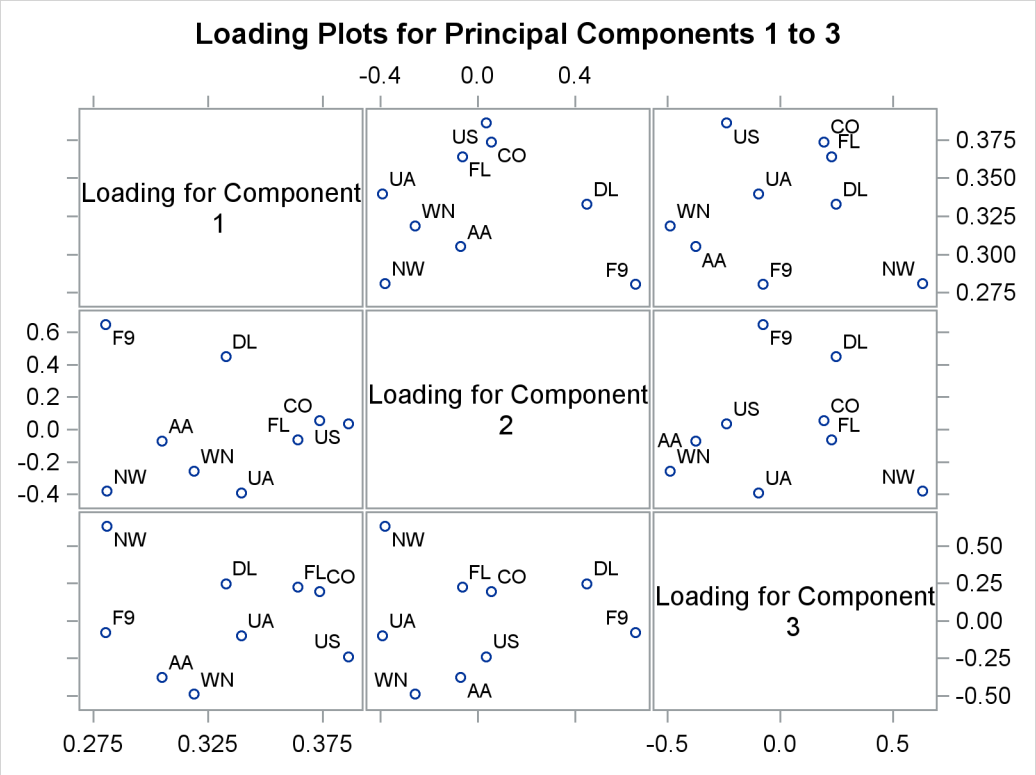
Figure 13.9 Score Plots for Principal Components 1–3



A score plot is a scatter plot of the scores for two principal components. The labels indicate the observation numbers of the points. By examining clusters and outliers in these plots, you can better understand the relationships among the observations and the variation in the process. For example, points 13 through 16 are extreme points in the direction of the first principal component. The directions of the principal components are not uniquely determined, so you need the loadings and external information to interpret them. These points represent flight delays between February 13, 2007, and February 16, 2007, when there was a major winter storm in the Midwest.

Figure 13.10 displays the loading plots that are produced. Loading plots are also displayed in a matrix by default, and they can be unpacked into separate graphs with the PLOT(LOADINGS(UNPACK)) option.

Figure 13.10 Loading Plot for Principal Components 1–3



A loading plot is a scatter plot of the variable loadings for a pair of principal components, and it helps you understand the relationships among the variables. Loadings are the variable coefficients in the eigenvectors (linear combinations of variables) that define the principal component. The loadings explain how variables contribute to the linear combination. Here, the loadings for the first principal component are all positive and all similar in value, which suggests that the first principal component describes the average delay. The second principal component appears to be a contrast between the delays of F9, DL, CO, and US and those of the remaining airlines. See the section "Principal Component Analysis" on page 942 for more information about interpreting principal component loadings and scores.

Syntax: MVPMODEL Procedure

The following statements are available in PROC MVPMODEL:

PROC MVPMODEL < *options* > **;**
BY *variables* **;**
ID *variables* **;**
VAR *variables* **;**

The following sections describe the PROC MVPMODEL statement and then describe the other statements in alphabetical order.

PROC MVPMODEL Statement

PROC MVPMODEL < *options* > **;**

The PROC MVPMODEL statement invokes the MVPMODEL procedure and optionally identifies input and output data sets, specifies details of the analyses performed, and controls displayed output. Table 13.1 summarizes the *options*.

Table 13.1 Summary of PROC MVPMODEL Statement Options

option	**Description**
COV	Computes the principal components from the covariance matrix
CV=	Performs cross validation to select the number of principal components
DATA=	Specifies the input data set
MISSING=	Specifies how observations with missing values are handled
NCOMP=	Specifies the number of principal components to extract
NOCENTER	Suppresses centering of process variables before fitting the model
NOCVSTDIZE	Suppresses re-centering and rescaling of process variables before each model is fit in the cross validation
NOPRINT	Suppresses the display of all output
NOSCALE	Suppresses scaling of process variables before fitting the model
OUT=	Specifies the output data set
OUTLOADINGS=	Specifies the output data set for loadings (eigenvectors)
PLOTS=	Requests and specifies details of plots
PREFIX=	Specifies the prefix for naming principal component score variables in the OUT= data set
RPREFIX=	Specifies the prefix for naming residual variables in the OUT= data set
STDSCORES	Standardizes the principal component scores

You can specify the following *options*.

COV

computes the principal components from the covariance matrix. By default, the correlation matrix is analyzed. The COV option causes variables with large variances to be more strongly associated with components that have large eigenvalues, and it causes variables with small variances to be more strongly associated with components that have small eigenvalues. You should not specify the COV option unless the units in which the variables are measured are comparable or the variables are standardized in some way.

NOTE: Specifying the COV option has the same effect as specifying the NOSCALE option.

CV=ONE

CV=BLOCK < **(***cv-block-options***)** >

CV=SPLIT < **(***cv-split-options***)** >

CV=RANDOM < **(***cv-random-options***)** >

specifies that cross validation be performed to determine the number of principal components and specifies the method to be used. If you do not specify the CV= option, no cross validation is performed.

In cross validation, the input data are repeatedly divided into a *training set*, which is used to compute a model, and a *test set*, which is used to test the model fit. The cross validation that is performed here is along both observations and variables, as described in Eastment and Krzanowski (1982), which is a more detailed version of the "alternative scheme" of Wold (1978). The observations and variables are separately divided into groups. Each test set is the intersection of one observation group and one variable group, so the number of test sets that are used is the product of the number of observation groups and the number of variable groups. See the section "Cross Validation (Experimental)" on page 945 for more information.

NOTE: The CV= option is experimental in this release.

CV=ONE requests *one-at-a-time* cross validation, in which each observation group contains one observation and each variable group contains one variable. This approach is very computationally intensive because it computes $n \times p$ separate principal component models for each potential number of principal components, where n is the number of observations in the input data set and p is the number of process variables.

CV=BLOCK requests *blocked* cross validation, in which observation groups consist of blocks of *nobs* consecutive observations and variable groups consist of blocks of *nvar* consecutive variables. You can specify the following *cv-block-options* in parentheses after the CV=BLOCK option:

NOBS=*nobs*
: specifies that observation groups consist of blocks of *nobs* consecutive observations from the input data. For example, if you specify NOBS=*8*, the first group contains observations 1 through 8, the second group contains observations 9 through 16, and so on. The default is 7.

NVAR=*nvar*
: specifies that variable groups consist of blocks of *nvar* consecutive variables from the input data. For example, if you specify NVAR=*3*, the first group contains variables 1 through 3, the second group contains variables 4 through 6, and so on. The default is 7.

CV=SPLIT requests *split-sample* cross validation, in which observation groups are formed by selecting every *nobs*th observation and variable groups are formed by selecting very *nvar*th variable. You can specify the following *cv-split-options* in parentheses after the CV=SPLIT option:

NOBS=*nobs*
: specifies that observation groups be created by selecting every *nobs*th observation from the input data. For example, if you specify NOBS=*8*, the first group contains observations {1, 9, 17, ... }, the second group contains observations {2, 10, 18, ... }, and so on. The default is 7.

NVAR=*nvar*
: specifies that variable groups be created by selecting every *nvar*th variable from the input data. For example, if you specify NVAR=*5*, the first group contains variables {1, 6, 11, ... }, the second group contains variables {2, 7, 12, ... }, and so on. The default is 7.

CV=RANDOM requests that observations and variables be assigned to groups randomly. You can specify the following *cv-random-options* in parentheses after the CV=RANDOM option:

NITEROBS=*nogrp*
: specifies the number of observation groups. The default is 10.

NITERVAR=*nvgrp*
specifies the number of variable groups. The default is 10.

NTESTOBS=*nobs*
specifies the number of observations in each observation group. The default is one-tenth the total number of observations.

NTESTVAR=*nvar*
specifies the number of variables in each variable group. The default is one-tenth the total number of variables.

SEED=*n*
specifies an integer used to start the pseudorandom number generator for selecting the random test set. If you do not specify a seed or if you specify a value less than or equal to zero, the seed is generated by default from reading the time of day from the computer's clock.

NOTE: You cannot specify the CV= option together with the NCOMP= option.

DATA=*SAS-data-set*
specifies the input SAS data set to be analyzed. If the DATA= option is omitted, the procedure uses the most recently created SAS data set.

MISSING=AVG | NONE
specifies how observations with missing values are to be handled in computing the fit. MISSING=AVG specifies that the fit be computed by replacing missing values of a process variable with the average of its nonmissing values. The default is MISSING=NONE, which excludes observations with missing values for any process variables from the analysis.

NCOMP=*n* **| ALL**
specifies the number of principal components to extract. The default is $\min\{15, p, N\}$, where p is the number of process variables and N is the number of observations (runs). You can specify NCOMP=ALL to override the limit of 15 principal components. You cannot specify the NCOMP= option together with the CV= option. If the number of nonzero eigenvalues of the correlation matrix is less than the number of components specified, p, then the p will be reset to the number of nonzero eigenvalues.

NOCENTER
suppresses centering of the process variables before fitting. This is useful if the variables are already centered and scaled. See the section "Centering and Scaling" on page 946 for more information.

NOCVSTDIZE
suppresses re-centering and rescaling of the process variables before each model is fit in the cross validation. See the section "Centering and Scaling" on page 946 for more information.

NOPRINT
suppresses the display of all results, both tabular and graphical. This is useful when you want to produce only output data sets.

NOSCALE
suppresses scaling of the process variables before fitting. This is useful if the variables are already centered and scaled.

NOTE: Specifying the NOSCALE option has the same effect as specifying the COV option.

OUT=*SAS-data-set*
creates an output data set that contains all the original data from the input data set, principal component scores, and multivariate summary statistics. See the section "Output Data Sets" on page 947 for details.

OUTLOADINGS=*SAS-data-set*
creates an output data set that contains the loadings for the principal components and the eigenvalues of the correlation (or covariance) matrix. See the section "Output Data Sets" on page 947 for details.

PLOTS <(*global-plot-options***) > <= ***plot-request* < *(options)* > **>**

PLOTS <(*global-plot-options***) > <= (***plot-request* < *(options)* > **<...** *plot-request* < *(options)* > **>) >**
controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. For example:

```
plots=none
plots=score
plots=loadings
```

ODS Graphics must be enabled before you request plots. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

You can specify the following *global-plot-options*:

FLIP
interchanges the X-axis and Y-axis dimensions for all score and loading plots.

NCOMP=*n*
specifies that pairwise score and loading plots be produced for the first *n* principal components. The default is 5 or the total number of components $j(\geq 2)$, whichever is smaller. If $n > j$, then the default is NCOMP=*j*. Be aware that the number of score or loading plots produced ($\frac{n\times(n-1)}{2}$) grows quadratically as *n* increases.

ONLY
suppresses the default plots. Only plots specifically requested are displayed. The default plots are the CV plot, when you specify the CV= option, and the scree and variation-explained plots otherwise.

You can specify the following *plot-requests*:

ALL
produces all appropriate plots.

CVPLOT
produces a plot that displays the results of the cross validation and R-square analysis. This plot requires that the CV= option be specified and in that case is displayed by default.

LOADINGS < (*loading-options*) >
produces a matrix of pairwise scatter plots of the principal component loadings. Use NCOMP=*n* to specify the number of principal components for which plots are produced, and use the FLIP option to interchange the default X-axis and Y-axis dimensions.

You can specify the following *loading-options*:

FLIP
flips or interchanges the X-axis and Y-axis dimensions of the loading plots. Specify PLOTS=LOADING(FLIP) to flip the X-axis and Y-axis dimensions.

NCOMP=*n*
specifies that pairwise loading plots be produced for the first *n* principal components. The default is the value specified by the NCOMP= *global-plot-option*. If $n > j$, then the default is NCOMP=*j*. Be aware that the number of loading plots produced ($\frac{n\times(n-1)}{2}$) grows quadratically as *n* increases.

UNPACKPANEL
UNPACK
suppresses paneling of loading plots. By default, all the loading plots appear in a single output panel. Specify UNPACKPANEL to display each loading plot in a separate panel.

NONE
suppresses the display of all plots.

SCORES < (*score-options*) >
produces pairwise scatter plots of the principal component scores. You can use the NCOMP= option to control the number of plots that are displayed.

You can specify the following *score-options*:

ALPHA=*value*
specifies the probability used to compute a prediction ellipse that is overlaid on the score plot. The default is 0.05. If you specify the ALPHA= option, you do not need to specify the ELLIPSE option.

ELLIPSE
requests that a prediction ellipse be overlaid on the principal component score plots. The probability that a new observation falls outside the prediction ellipse is specified by the ALPHA= option.

FLIP
flips or interchanges the X-axis and Y-axis dimensions of the score plots. Specify PLOTS=SCORES(FLIP) to flip the X-axis and Y-axis dimensions.

GROUP=*variable*
specifies a variable in the input data set used to group the points on the score plots. Points with different GROUP= variable values are plotted using different markers and colors to distinguish the groups.

LABELS=ON | OFF | OUTSIDE
specifies which points in the score plots to label. Specify LABELS=ON to label all points and LABELS=OFF to label none of the points. Points are labeled with the values of the first variable listed in the ID statement, or the observation number if no ID statement is specified.

If you specify the ELLIPSE and UNPACKPANEL options, you can specify LABELS=OUTSIDE to label only the points outside the confidence ellipse.

The default is ON if you specify UNPACKPANEL and OFF otherwise.

NCOMP=*n*
specifies that pairwise score plots be produced for the first *n* principal components. The default is the value specified by the NCOMP= *global-plot-option*. If $n > j$, then the default is NCOMP=*j*. Be aware that the number of loading plots produced ($\frac{n \times (n-1)}{2}$) grows quadratically as *n* increases.

UNPACKPANEL
suppresses paneling of score plots. By default, all the score plots appear in a single output panel. Specify UNPACKPANEL to display each score plot in a separate panel.

SCREE < UNPACK >
EIGEN
EIGENVALUE
produces a scree plot of eigenvalues and a variance-explained plot. By default, both plots are produced in a panel. Specify PLOTS= SCREE(UNPACKPANEL) to display each plot in a separate panel. This plot is produced by default unless you specify the CV= option.

PREFIX=*name*
specifies a prefix for naming the principal component scores in the OUT= data set. By default, the names are Prin1, Prin2, . . . , Prin*j*. If you specify PREFIX=ABC, the components are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits in *j* should not exceed the current name length defined by the VALIDVARNAME= system option.

RPREFIX=*name*
specifies a prefix for naming the residual variables in the OUT= data set. The default is R_. Residual variable names are formed by appending process variable names to the prefix.

If the length of the resulting residual variable exceeds the maximum name length defined by the VALIDVARNAME= system option, characters are removed from the middle of the process variable name before it is appended to the residual prefix. For example, if you specify RPREFIX=*Residual_*, the maximum variable name length is 32, and there is a process variable named PrimaryThermometerReading, then the corresponding residual variable name is Residual_PrimaryThermeterReading.

STDSCORES
standardizes the principal component scores in the OUT= data set to unit variance. If you omit the STDSCORES option, the variances of the scores are equal to the corresponding eigenvalues. STDSCORES has no effect on the eigenvalues themselves.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MVPMODEL to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the MVPMODEL procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The first variable that is specified in the ID statement is used to label observations in score plots for principal components. If you do not specify an ID statement, then score plot points are labeled with their observation numbers.

The values of all ID variables are displayed in tooltips when you create HTML output and specify the IMAGEMAP option in the ODS GRAPHICS statement. See Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*), for details.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the process variables and their order in the results. By default, if you omit the VAR statement, the MVPMODEL procedure analyzes all numeric variables that are not listed in the BY or ID statement.

Details: MVPMODEL Procedure

Classical T^2 Charts

Classical T^2 charts are defined as follows. Assume that there are n observations for p variables, denoted by $\mathbf{X}_1, \ldots, \mathbf{X}_n$, where $\mathbf{X}_i$ is a p-dimensional vector. The T^2 statistic for observation i is

$$T_i^2 = \left(\mathbf{X}_i - \bar{\mathbf{X}}_n\right)' \mathbf{S}^{-1} \left(\mathbf{X}_i - \bar{\mathbf{X}}_n\right)$$

where

$$\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij} \quad , \quad \mathbf{X}_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{ip} \end{bmatrix}, \quad \bar{\mathbf{X}}_n = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_p \end{bmatrix}$$

and

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n} \left(\mathbf{X}_i - \bar{\mathbf{X}}_n\right) \left(\mathbf{X}_i - \bar{\mathbf{X}}_n\right)'$$

For purposes of deriving control limits for the T^2 chart, it is assumed that $\mathbf{X}_i$ has a p-dimensional multivariate normal distribution with mean vector $\boldsymbol{\mu} = \left(\mu_1, \mu_2, \cdots, \mu_p\right)'$ and covariance matrix $\boldsymbol{\Sigma}$ for $i = 1, 2, \ldots, n$. The classical formulation of the T^2 chart does not involve a principal component model for the data, and it bases the computation of T^2 on the sample covariance matrix $\mathbf{S}$. See Alt (1985) for theoretical details and the section "Multivariate Control Charts" on page 2177 for an example.

A classical T^2 chart is equivalent to a T^2 chart based on a full principal component model (with p components), as discussed in the section "Relationship of Principal Components to Multivariate Control Charts" on page 943. See Example 13.2 for more information.

Principal Component Analysis

Principal component analysis was originated by Pearson (1901) and later developed by Hotelling (1933). The application of principal components is discussed by Rao (1964), Cooley and Lohnes (1971), Gnanadesikan (1977), and Jackson (1991). Excellent statistical treatments of principal components are found in Kshirsagar (1972), Morrison (1976), and Mardia, Kent, and Bibby (1979).

Principal component modeling focuses on the number of components used. The analysis begins with an eigenvalue decomposition of the sample covariance matrix, $\mathbf{S}$,

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n} \left(\mathbf{X}_i - \bar{\mathbf{X}}_n\right) \left(\mathbf{X}_i - \bar{\mathbf{X}}_n\right)'$$

as

$$\begin{aligned} \mathbf{S} &= \mathbf{PLP}' \\ \mathbf{P}'\mathbf{SP} &= \mathbf{L} \end{aligned}$$

where $\mathbf{L}$ is a diagonal matrix and $\mathbf{P}$ is an orthogonal matrix (Jackson 1991; Mardia, Kent, and Bibby 1979). The columns of $\mathbf{P}$ are the eigenvectors, and the diagonal elements of $\mathbf{L}$ are the eigenvalues. The eigenvectors are customarily scaled so that they have unit length.

A principal component, $\mathbf{t}_i$, is a linear combination of the original variables. The coefficients are the eigenvectors of the covariance matrix. The principal component scores for the ith observation are computed as

$$\mathbf{t}_i = \mathbf{P}' (\mathbf{x}_i - \bar{\mathbf{x}})$$

The principal components are sorted by descending order of the eigenvalues, which are equal to the variances of the components.

The eigenvectors are the principal component loadings. The eigenvectors are orthogonal, so the principal components represent jointly perpendicular directions through the space of the original variables. The scores on the first j principal components have the highest possible generalized variance of any set of j unit-length linear combinations of the original variables.

The first j principal components provide a least squares solution to the model

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E}$$

where $\mathbf{X}$ is an $n \times p$ matrix of the centered observed variables, $\mathbf{T}$ is the $n \times j$ matrix of scores on the first j principal components, $\mathbf{P}'$ is the $j \times p$ matrix of eigenvectors, and $\mathbf{E}$ is an $n \times p$ matrix of residuals. The first j principal components are the vectors (rows of $\mathbf{P}'$) that minimize trace($\mathbf{E}'\mathbf{E}$), the sum of all the squared elements in $\mathbf{E}$.

The first j principal components are the best linear predictors of the process variables among all possible sets of j variables, although any nonsingular linear transformation of the first j principal components provides equally good prediction. The same result is obtained by minimizing the determinant or the Euclidean norm of $\mathbf{E}'\mathbf{E}$ rather than the trace.

Relationship of Principal Components to Multivariate Control Charts

Multivariate control charts typically plot the T^2 statistic, which is a summary of multivariate variation. The classical T^2 statistic is defined in "Classical T^2 Charts" on page 942. When there is high correlation among the process variables, the correlation matrix is nearly singular. The subspace in which the process varies can be adequately explained by fewer variables than the original p variables. Thus, the principal component approach to multivariate control charts is to project the original p variables into a lower-dimensional subspace by using a model based on j principal components, where $j < p$.

The key to the relationship between principal components and multivariate control charts is the decomposition of the sample covariance matrix, $\mathbf{S}$, into the form $\mathbf{S} = \mathbf{PLP}'$, where $\mathbf{L}$ is a diagonal matrix (Jackson 1991; Mardia, Kent, and Bibby 1979). This is also the eigenvalue decomposition of $\mathbf{S}$, where the columns of $\mathbf{P}$ are the eigenvectors and the diagonal elements of $\mathbf{L}$ are the eigenvalues.

Equivalence of T^2 Statistics

The T^2 statistic that is produced by the full principal component model is equivalent to the classical T^2 statistic. This is seen in the matrix representation of the T^2 statistic computed from a principal component model that uses all p components,

$$T_i^2 = \left(\mathbf{t}_i - \bar{\mathbf{t}}_n\right)' \mathbf{L}_n^{-1} \left(\mathbf{t}_i - \bar{\mathbf{t}}_n\right)$$

Because $\bar{\mathbf{t}}_n$ is the zero matrix by construction, then

$$T_i^2 = \mathbf{t}_i' \mathbf{L}_n^{-1} \mathbf{t}_i$$

Because $\mathbf{t}_i = \mathbf{P}' \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)$, then

$$\begin{aligned} T_i^2 &= \mathbf{t}_i' \mathbf{L}_n^{-1} \mathbf{t}_i \\ &= \left(\mathbf{P}' \left(\mathbf{x_i} - \bar{\mathbf{x}}\right)\right)' \mathbf{L}_n^{-1} \left(\mathbf{P}' \left(\mathbf{x_i} - \bar{\mathbf{x}}\right)\right) \\ &= \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)' \mathbf{P} \mathbf{L}_n^{-1} \mathbf{P}' \left(\mathbf{x}_i - \bar{\mathbf{x}}\right) \\ &= \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)' \mathbf{S}^{-1} \left(\mathbf{x}_i - \bar{\mathbf{x}}\right) \end{aligned}$$

which is the classical form. Consequently the classical T^2 statistic can be expressed as a sum of squares,

$$T_i^2 = \frac{t_{i1}^2}{l_1^2} + \ldots + \frac{t_{ip}^2}{l_p^2}$$

where l_k^2 is the variance of the kth principal component.

Computing the T^2 and SPE Statistics

Creating a T^2 chart that is based on a principal component model begins with choosing the number (j) of principal components. Effectively, this involves selecting a subspace in $j < p$ dimensions and then creating a T^2 statistic based on that j-component model.

The T^2 statistic is meant to monitor variation in the model space. However, if variation appears in the $p - j$ subspace orthogonal to model space, then the model assumptions and physical process should be reexamined. Variation outside the model space can be detected with an SPE chart.

In a model with j principal components, the T^2 statistic is calculated as

$$T_i^2 = \frac{t_{i1}^2}{l_1^2} + \ldots + \frac{t_{ij}^2}{l_j^2}$$

where t_{ik} is the principal component score for the kth principal component of the ith observation and l_k is the standard deviation of t_{ik}.

The information in the remaining $p - j$ principal components is monitored with charts for the SPE statistic, which is calculated as

$$\begin{aligned} \mathrm{SPE}_i &= \sum_{k=j+1}^{p} e_{ik}^2 \\ &= \sum_{k=j+1}^{p} \left(x_{ik} - \hat{x}_{ik}\right)^2 \end{aligned}$$

Cross Validation (Experimental)

NOTE: The CV= option is experimental in this release.

You can use cross validation to choose the number of principal components in the model to avoid overfitting.

One method of choosing the number of principal components is to fit the model to only part of the available data (the *training set*) and to measure how well models with different numbers of extracted components fit the other part of the data (the *test set*). This is called *test set validation*. However, it is rare that you have enough make both parts large enough for pure test set validation to be useful. Alternatively, you can make several different divisions of the observed data into a training set and a test set. This is called *cross validation*. The MVPMODEL procedure supports four types of cross validation. In *one-at-a-time* cross validation, the first observation is held out as a single-element test set, with all other observations as the training set; next, the second observation is held out, then the third, and so on. Another method is to hold out successive blocks of observations as test sets—for example, observations 1 through 7, then observations 8 through 14, and so on; this is known as *blocked* validation. A similar method is *split-sample* cross validation, in which successive groups of widely separated observations are held out as the test set—for example, observations {1, 11, 21, ... }, then observations {2, 12, 22, ... }, and so on. Finally, test sets can be selected from the observed data randomly; this is known as *random-sample* cross validation.

Which cross validation method you should use depends on your data. The most common method is one-at-a-time validation (CV=ONE), but it is not appropriate when the observed data are serially correlated. In that case either blocked (CV=BLOCK) or split-sample (CV=SPLIT) validation might be more appropriate; you can select the number of test sets in blocked or split-sample validation by specifying options in parentheses after the CV= option. The numbers in parentheses are the number of test sets over the rows and columns. For more information, see the section "An Alternative Scheme" in Wold (1978), as well as Eastment and Krzanowski (1982), both of which describe the cross validation approach used here in more detail.

CV=ONE is the most computationally intensive of the cross validation methods, because it requires you to recompute the principal component model for every input observation. Using random subset selection with CV=RANDOM might lead different researchers to produce different principal component models from the same data (unless the same seed is used).

Whichever validation method you use, the number of principal components that are chosen is usually the one that optimizes some criterion or selection rule. Choices of a criterion include the ratio described by Wold (1978), the W statistic described by Eastment and Krzanowski (1982), and the predicted residual sum of squares (PRESS). The W statistic is used by the MVPMODEL procedure.

The method of choosing the number of principal components in the MVPMODEL procedure is described in Eastment and Krzanowski (1982). This method is a heuristic based on the ratio of the mean PRESS (MPRESS) to the degrees of freedom for the principal component model. First, the MPRESS is computed for models with 0 to $maxcomp$ principal components. The maximum number of components is $\min(15, nvar, nobs) - 1$ and can be further reduced to the number of nonzero eigenvalues in the covariance matrix. Second, for each of the i possible number of components, the W_i statistic is computed as

$$W_i = \frac{MPRESS(i-1) - MPRESS(i)}{D_i} \div \frac{MPRESS(i)}{D_R}$$

where $MPRESS = \frac{1}{np} PRESS$, D_i is the number of degrees of freedom used to fit the model with i principal components, and D_R is the remaining number of degrees of freedom.

Extracting too many components can lead to an overfit model, one that matches the training data too well, sacrificing predictive ability. Thus, if you specify the number of principal components in the model, you should not use cross validation to select the appropriate number of components for the final model, or you should consider the analysis to be preliminary and examine the results to determine the appropriate number of components for a subsequent analysis.

Centering and Scaling

By default, the variables are centered and scaled to have mean 0 and standard deviation 1. Without centering, both the mean variable value and the variation around that mean are involved in selecting principal component loadings. Scaling serves to place all process variables on an equal footing relative to their variation in the data. For example, if Time and Temp are two of the process variables, then scaling says that a change of std (Time) in Time is roughly equivalent to a change of std (Temp) in Temp.

The formulas that are used to compute the variation in the different centering and scaling cases are defined in the section "Definitional Formulas" in Chapter A, "Special SAS Data Sets" (*SAS/STAT User's Guide*). The definitional formula that is used when either the NOSCALE or COV option is specified is the COV formula. The definitional formula that is used when the NOCENTER option is specified is the UCORR formula. The definitional formula that is used when both the NOCENTER and NOSCALE options are specified is the UCOV formula. The default definitional formula, when no centering or scaling options are specified, is the CORR formula.

Missing Values

By default, observations that have missing process variables are simply excluded from the analysis. If you specify MISSING=AVG in the PROC MVPMODEL statement, then all observations in the input data set contribute to both the analysis and the OUT= data set. With MISSING=AVG, the fit is computed by replacing missing values of a process variable with the average of its nonmissing values.

Input Data Set

The input data set provides the set of process variables that are analyzed. You can specify the input data set by using the DATA= option in the PROC MVPMODEL statement. If you do not specify the DATA= option, the procedure uses the last data set created as its input data set.

The MVPMODEL procedure treats each observation in the DATA= data set as an individual multivariate observation. The observations do not need to be identified or sorted by time because the sequence of the data is not used to build the principal component model. If you provide a time variable in the input data set, it is preserved in the OUT= data set and can be used subsequently by the MVPMONITOR procedure to create control charts.

In basic applications of the MVPMODEL procedure, the observations in the DATA= data set represent measurements from a single process. You can build different principal component models for two or more processes by grouping their measurements in the DATA= data and processing them as BY groups.

In some applications, it is desirable to combine the data from two or more processes and build a common principal component model. This might be the case with processes that are peers in the sense that they are believed to share the same pattern of common cause variation. When you provide the MVPMONITOR procedure with a common model for a set of peer processes, it uses the model to construct identical control limits for each process. This enables you to decide whether a particular process exhibits unusual variation relative to the behavior of its peers.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the input data set plus new variables that contain the principal component scores, residuals, and other computed values listed in Table 13.2.

The names of the score variables are formed by concatenating the value given by the PREFIX= option (or the default Prin, if PREFIX= is not specified) and the numbers 1, 2, ..., j, where j is the number of principal components in the model.

The names of the residual variables are formed by concatenating the value given by the RPREFIX= option (or the default R_, if RPREFIX= is not specified) and the names of the process variables used in the analysis. Residual variables are created only when the number of principal components in the model is less than the number of process measurement variables in the input data set.

Table 13.2 Computed Variables in the OUT= Data Set

Variable	Description
Prin1–Prin$j$	Principal component scores
R_$var1$–R_$varp$	Residuals
NOBS	Number of observations used in the analysis
SPE	Squared prediction error (SPE)
TSQUARE	T^2 statistic computed from principal component scores

OUTLOADINGS= Data Set

The OUTLOADINGS= data set contains the eigenvalues of the correlation (or covariance) matrix, the loadings computed for the process variables, and other information about the principal component model. The variables that are saved in the OUTLOADINGS= data set are listed in Table 13.3.

Table 13.3 Variables in the OUTLOADINGS= Data Set

Variable	Description
VALUE	Character variable identifying the type of values in an observation
_PC_	Principal component number
NOBS	Number of observations used in the analysis
process variables	Eigenvalues, means, standard deviations, and loadings for *process variables*

Valid values for the _VALUE_ variable are as follows:

EIGEN	eigenvalues from the principal component analysis
LOADING	principal component loadings
MEAN	process variable means
STD	process variable standard deviations

For an observation where _VALUE_ is equal to LOADING, the _PC_ variable identifies the principal component whose loadings are recorded in that observation.

The process variable means and standard deviations are used by the other MVP procedures to center and scale new data in a Phase II analysis. If you specify the NOCENTER option, the OUTLOADINGS= data set does not contain a MEAN observation. If you specify the NOSCALE option, the OUTLOADINGS= data set does not contain a STD observation.

ODS Table Names

PROC MVPMODEL assigns a name to each table that it creates. You can use these names to refer to the tables when you use the Output Delivery System (ODS) to select tables and create output data sets. The ODS table names are listed in Table 13.4.

Table 13.4 ODS Tables Produced with the PROC MVPMODEL Statement

ODS Table Name	Description	Option
Corr	Correlation matrix	Default
Cov	Covariance matrix	COV or NOSCALE
CVResults	Results of cross validation	CV=
Eigenvalues	Eigenvalues of the correlation or covariance matrix	Default
ModelInfo	Model information	Default
ResidualSummary	Residual summary from cross validation	CV=

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The MVPMODEL procedure assigns a name to each graph that it creates using ODS Graphics. You can use these names to refer to the graphs when you use ODS. The ODS graph names are listed in Table 13.5.

Table 13.5 ODS Graphics Produced by PROC MVPMODEL

ODS Graph Name	Plot Description	Statement
CVPlot	Cross validation and R^2 analysis	CV=
LoadingMatrix	Scatter plot matrix of variable loadings	PLOTS=LOADINGS
LoadingPlot	Scatter plot of variable loadings	PLOTS=LOADINGS(UNPACK)
ScoreMatrix	Scatter plot of scores	PLOTS=SCORE
ScorePlot	Scatter plot of scores	PLOTS=SCORE(UNPACK)
ScreePlot	Scree and variance-explained plots	Default
VariancePlot	Variance-explained plot	PLOTS=SCREE(UNPACK)

Examples: MVPMODEL Procedure

Example 13.1: Using Cross Validation to Select the Number of Principal Components

This example uses cross validation to select the number of principal components in a model. It uses the chromatography data from McReynolds (1970), which is also used in Wold (1978) and Eastment and Krzanowski (1982). The following statements create the chromatography data set:

```
data mcreynolds;
   input x1 - x10;
   datalines;
 653    590    627    652    699    690    818    841    654    1006
 654    591    628    654    701    691    818    842    655    1006
 665    592    624    653    710    690    828    843    659    1014
 662    595    629    658    710    692    827    843    660    1012
 663    595    630    659    712    693    829    843    663    1013
 664    596    629    659    712    692    830    843    663    1015
 667    604    635    669    720    700    833    846    668    1016
 684    612    642    682    739    702    850    851    682    1035
 685    612    642    684    741    703    853    852    685    1039

   ... more lines ...

1247   1447   1386   1683   1616   1370   1327   1220   1508    1275
1300   1509   1424   1695   1675   1403   1362   1229   1571    1305
1343   1581   1480   1762   1699   1463   1375   1212   1618    1285
;
```

The observations are liquid phases, and the variables are compounds. The (i, j) value is the retention index for liquid phase i in compound j. The retention index values in the original article had the value of squalane subtracted from them. In this data set, the values have been corrected by adding the retention indices for squalane to all observations.

The following statements use the MVPMODEL procedure to select the number of principal components by using one-at-a-time cross validation:

```
proc mvpmodel data=mcreynolds plots=(scree cvplot) noscale cv=one;
run;
```

The CV= option specifies which method of cross validation to use to produce model diagnostics; in this case one-at-a-time cross validation is used. The PLOTS= option produces only the combination scree plot and variance-explained plot in addition to the cross validation plots.

Output 13.1.1 shows the model and data set information.

Output 13.1.1 Summary of Model and Data Set Information

The MVPMODEL Procedure

Data Set	WORK.MCREYNOLDS
Number of Variables	10
Missing Value Handling	Exclude
Number of Observations Read	226
Number of Observations Used	225
Maximum Number of Principal Components	9
Validation Method	Leave-one-out Cross Validation

Output 13.1.1 shows that one observation, liquid phase 69 (Triton X-400), was omitted because of a missing value. Also, notice that the maximum number of principal components is $\min(15, nvar, nobs) - 1 = 9$, which is less than the number of variables; this is described in detail in Eastment and Krzanowski (1982).

The root mean PRESS values and the W statistic are shown in Output 13.1.2.

Output 13.1.2 Residual Summary

Cross Validation for the Number of Components		
Number of Components	**Root Mean PRESS**	**W**
0	974.3136	.
1	30.77631	9586.179
2	26.85973	2.707278
3	26.49878	0.211824
4	22.94873	2.261922
5	21.50501	0.810642
6	20.91568	0.279385
7	20.53967	0.14514
8	20.25766	0.082967
9	20.03932	0.04342

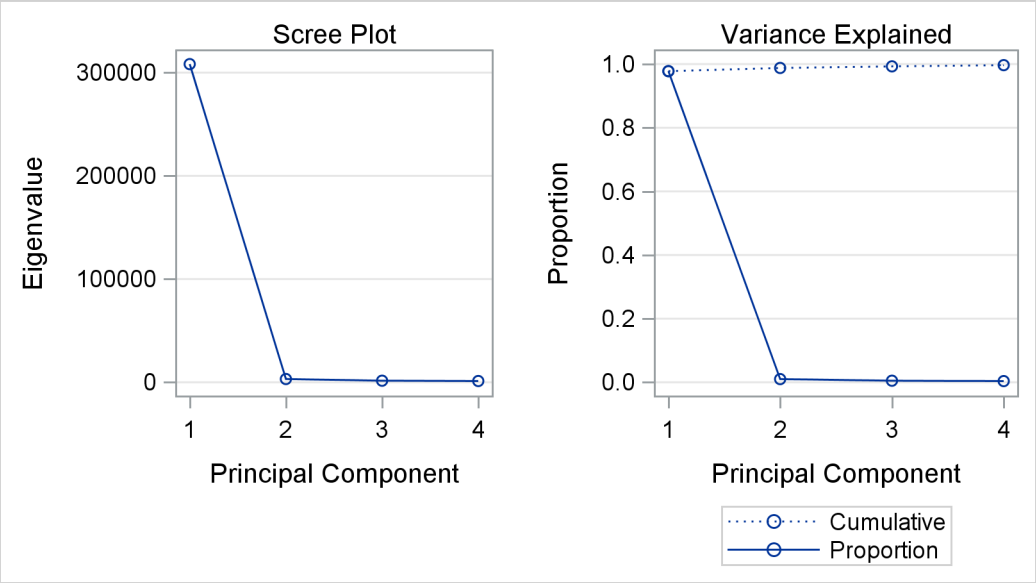
In this case the index of the last W statistics greater than one is $W[4]$, suggesting a model with four components as shown in Output 13.1.3.

Output 13.1.3 Cross Validation Results

Number of Components Suggested by W Statistic	4

You can also use scree and variance-explained plots to select the number of principal components, as shown in Output 13.1.4.

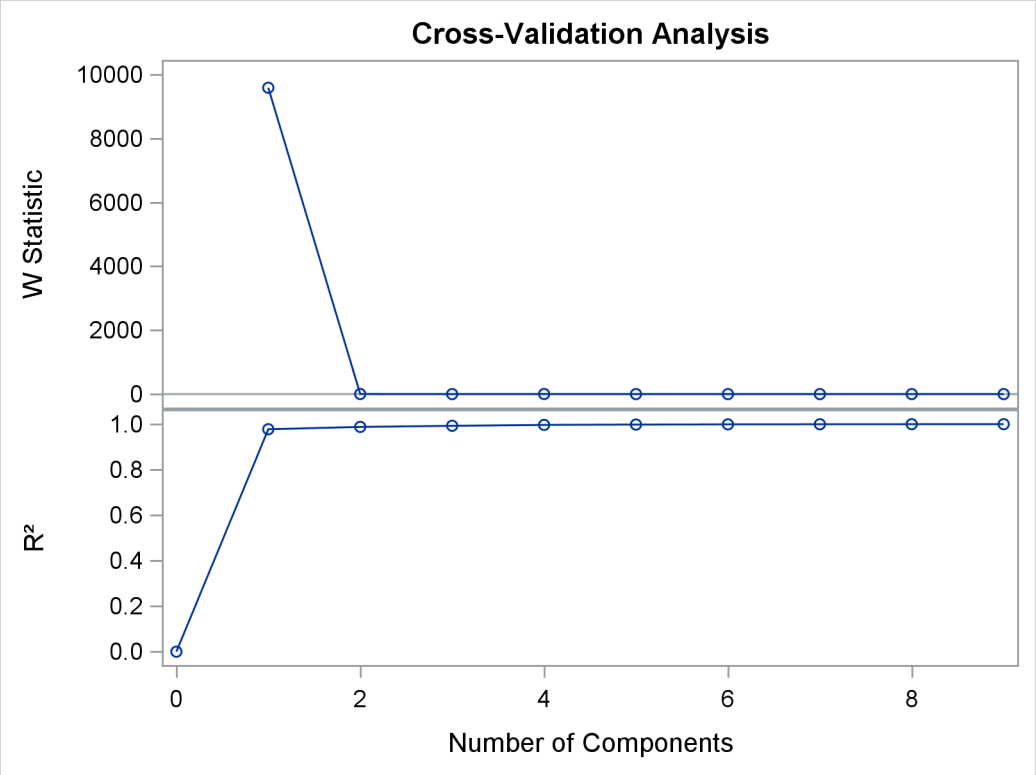
Output 13.1.4 Scree and Variance-Explained Plots



The plots in Output 13.1.4 indicate that one or two principal components explain almost all the variation.

The W statistic and R^2 plots are shown in Output 13.1.5.

Output 13.1.5 Cross Validation Analysis



The cross validation plot is produced only when you specify both the CV= option and PLOTS=ALL or PLOTS=CVPLOT.

It is interesting that the cross validation methods of Wold (1978) and Eastment and Krzanowski (1982) choose five and four components, respectively, for this model, whereas a visual examination of the knee in the scree plot might suggest using only one or two components.

Example 13.2: Computing the Classical T^2 Statistic

NOTE: The CV= option is experimental in this release.

This example uses the MVPMODEL procedure to produce a classical T^2 statistic and then compares it to the T^2 statistic produced by the principal component model with the NCOMP=ALL option. The two statistics are discussed in the section "Details: MVPMODEL Procedure" on page 942, and this example demonstrates that when the data set is centered and scaled correctly, the statistics are equal. The classical T^2 statistic is computed using the common quadratic form, which is implemented in SAS/IML. This example highlights the standardization that occurs by default in the MVPMODEL procedure. The example uses more of the airline delay data set that is first described in the section "Getting Started: MVPMODEL Procedure" on page 927. This data set covers the New England region of the continental United States. As before, the variables are airlines and the observations are mean daily delays during February 2007. The following statements create a SAS data set that contains these airline flight delays:

```
data flightDelaysNE;
   input  AA CO DL F9 FL NW UA US WN;
   datalines;
15.7  7.1  8.6  6.3 14.6  6.2  7.0 11.0  6.4
16.0 19.4 10.7  6.4 19.0  6.1  8.3 14.4 14.2
14.5  1.5  5.4 13.3 13.6  9.7 16.6  7.5  9.9
12.4 14.3  5.8  0.7 11.8 20.1 11.2  8.6  8.1
19.8 27.6  7.3 16.1 13.3 14.8 39.9 16.4  9.7
20.5 12.2  0.2 -4.8  3.7 14.2 41.7  4.9  9.2
 8.3  4.1  3.4  4.2 -2.3  6.3 24.9  8.7  4.4
 4.7 14.1  1.8 18.1 -1.9 10.2  5.4  5.8  3.7
16.7 15.0  3.5 11.8  0.8  7.3 11.1  7.2  5.1
 6.2  0.6  2.6  9.3  3.0  4.0  4.0  6.9  1.9
 6.9  8.4  0.3  1.7 -1.1 10.4  8.7  9.4  4.6
16.5  7.7  2.5  8.1  4.2 11.0 18.4  6.2  2.4
21.2 10.2  5.6  1.1 18.7  9.2 35.0 49.7 35.9
22.5 30.0 26.1 14.2 41.5 46.2 43.6 75.5 34.1
62.7 60.4 39.5 27.6 44.9 27.9 51.5 64.7 38.2
31.3 41.4 23.1 40.2 19.3 19.7 28.3 40.4 17.3
```

The following statements use the MVPMODEL procedure to create classical T^2 statistics:

```
proc mvpmodel data=flightDelaysNE ncomp=all plots=none out=mvpout;
   var AA CO DL F9 FL NW UA US WN;
run;
```

Specifying NCOMP=ALL sets the number of principal components to be used in the model equal to the number of process variables. Therefore, as discussed in the section "Details: MVPMODEL Procedure" on page 942, the mvpout data set contains the classical T^2 statistic for each observation, $T_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})$.

The following SAS/IML statements generate the Hotelling T^2 statistic for the data set by using the traditional quadratic form. However, the data must first be standardized as done by the MVPMODEL procedure.

NOTE: If you do not want PROC MVPMODEL to center or scale the data, specify the NOCENTER or NOSCALE option, respectively.

```
proc iml;
   use flightDelaysNE;
   read all into x;
   n = nrow(x);
   p = ncol(x);
   xc = x-x[:,]; /* Create a centered data set*/
   ss = xc[##,];  /* Compute sum of squares */
   std=sqrt(ss/(n-1));  /* Compute standard deviations */
   std_x = xc/std;  /* Create a standardized data set*/
   S= cov(std_x);  /* Compute covariance of standardized data*/
   tsq = J(n,1,.);
   do i = 1 to n;
   /* Compute the classical T2 statistic using quadratic form */
      tsq[i] = std_x[i,]*inv(S)*std_x[i,]`;
   end;
   varnames = "tsq";
   create classicTsq from tsq [colname = varnames];
   append from tsq;
quit;
```

To compare the output from the MVPMODEL procedure with the output from SAS/IML, a new data set, mvpTsq, which contains the T^2 statistics computed by using the quadratic form in SAS/IML, is created:

```
data mvpTsq;
   set mvpOut(rename=(_TSQUARE_=tsq));
   keep tsq;
run;
```

Finally, you can verify that the two statistics are equivalent within machine precision by using the COMPARE procedure:

```
proc compare base=classicTsq compare=mvpTsq
             method=relative briefsummary;
run;
```

Output 13.2.1 Comparison of T^2 Statistics

```
                    The COMPARE Procedure
                Comparison of WORK.CLASSICTSQ with WORK.MVPTSQ
                     (Method=RELATIVE, Criterion=0.00001)

      NOTE: All values compared are within the equality criterion used. However, 16
            of the values compared are not exactly equal.
```

References

Alt, F. (1985). "Multivariate Quality Control." In *Encyclopedia of Statistical Sciences,* vol. 6, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons.

Cooley, W. W., and Lohnes, P. R. (1971). *Multivariate Data Analysis*. New York: John Wiley & Sons.

Eastment, H. T., and Krzanowski, W. J. (1982). "Cross-Validatory Choice of the Number of Components from a Principal Component Analysis." *Technometrics* 24:73–77.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. New York: John Wiley & Sons.

Hotelling, H. (1933). "Analysis of a Complex of Statistical Variables into Principal Components." *Journal of Educational Psychology* 24:417–441, 498–520.

Jackson, J. E. (1991). *A User's Guide to Principal Components*. New York: John Wiley & Sons.

Kourti, T., and MacGregor, J. F. (1995). "Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods." *Chemometrics and Intelligent Laboratory Systems* 28:3–21.

Kourti, T., and MacGregor, J. F. (1996). "Multivariate SPC Methods for Process and Product Monitoring." *Journal of Quality Technology* 28:409–428.

Kshirsagar, A. M. (1972). *Multivariate Analysis*. New York: Marcel Dekker.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. London: Academic Press.

McReynolds, W. O. (1970). "Characterization of Some Liquid Phases." *Journal of Chromatographic Science* 8:685–691.

Miller, P., Swanson, R. E., and Heckler, C. H. E. (1998). "Contribution Plots: A Missing Link in Multivariate Quality Control." *Applied Mathematics and Computer Science* 8:775–792.

Morrison, D. F. (1976). *Multivariate Statistical Methods*. 2nd ed. New York: McGraw-Hill.

Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space." *Philosophical Magazine* 6:559–572.

Rao, C. R. (1964). "The Use and Interpretation of Principal Component Analysis in Applied Research." *Sankhyā, Series A* 26:329–358.

Wold, S. (1978). "Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models." *Technometrics* 20:397–405.

Chapter 14
The MVPMONITOR Procedure

Contents

Overview: MVPMONITOR Procedure **957**
Getting Started: MVPMONITOR Procedure **958**
Syntax: MVPMONITOR Procedure **962**
PROC MVPMONITOR Statement 963
BY Statement 964
ID Statement 965
SERIES Statement 965
SCORECHART Statement 965
SPECHART Statement 967
TIME Statement 967
TSQUARECHART Statement 968
Common Chart Statement Options 969
Details: MVPMONITOR Procedure **972**
Computing T^2 Control Limits 972
Computing SPE Control Limits 973
Contribution Plots 974
Input Data Sets 975
Output Data Sets 978
ODS Graphics 980
Examples: MVPMONITOR Procedure **980**
Example 14.1: Combining Data from Peer Processes 981
Example 14.2: Creating Multivariate Control Charts for Phase II 986
Example 14.3: Comparison of Univariate and Multivariate Control Charts 988
Example 14.4: Creating a Classical T^2 Chart 993
References **995**

Overview: MVPMONITOR Procedure

The MVPMONITOR procedure is used in conjunction with the MVPMODEL and MVPDIAGNOSE procedures to monitor multivariate process variation over time, to determine whether the process is stable, and to detect and diagnose changes in a stable process. Collectively, these three procedures are referred to as the *MVP procedures*. See Chapter 11, "Introduction to Multivariate Process Monitoring Procedures," for a description of how the MVP procedures work together, and Chapter 12, "The MVPDIAGNOSE Procedure," and Chapter 13, "The MVPMODEL Procedure," for detailed descriptions of the other MVP procedures.

The MVPMONITOR procedure produces control charts for multivariate process data. It reads data sets that contain statistics and principal component model information such as those created by the MVPMODEL procedure. The MVPMONITOR procedure creates two types of multivariate control chart: T^2 charts and squared prediction error (SPE) charts. It can also produce contribution plots for out-of-control points in the multivariate control charts and univariate control charts of principal component scores.

Multivariate control charts detect unusual variation that would not be uncovered by individually monitoring the process variables with univariate control charts, such as Shewhart charts. A major impetus in the development of multivariate control charts is the inadequacy of individual univariate control charts in handling correlated measurement variables. A multivariate control chart can detect changes in the linear relationships of the variables in addition to their marginal means and variances.

The multivariate control charts that the MVPMONITOR procedure produces are based on principal component models that reduce the dimensionality of the data by projecting the process measurements to a low-dimensional subspace that is defined by a small number of principal components. This subspace is also known as the *model hyperplane*. T^2 charts are used to monitor variation within the model hyperplane, whereas SPE charts show variation from the model hyperplane.

The principal component approach offers several advantages over the construction of the classical T^2 chart:

- It avoids computational issues that arise when the process variables are collinear and their covariance matrix is nearly singular.
- It offers diagnostic tools for interpreting unusual values of T^2.
- By projecting the data to a low-dimensional subspace, it more adequately describes the variation in a multivariate process, which is often driven by a small number of underlying factors that are not directly observable.

Getting Started: MVPMONITOR Procedure

This example illustrates the basic features of the MVPMONITOR procedure by using airline flight delay data available from the U.S. Bureau of Transportation Statistics at `http://www.transtats.bts.gov`. The example applies multivariate process monitoring to flight delays, and it is a continuation of the example in the section "Getting Started: MVPMODEL Procedure" on page 927.

Suppose you want to use a principal component model to create T^2 and SPE charts to monitor the variation in flight delays. These charts are appropriate because the data are multivariate and correlated.

The following statements create a SAS data set named MWflightDelays, which provides daily average delays by airline for flights that originated in the midwestern United States. The data set contains variables for nine airlines: AA (American Airlines), CO (Continental Airlines), DL (Delta Airlines), F9 (Frontier Airlines), FL (AirTran Airways), NW (Northwest Airlines), UA (United Airlines), US (US Airways), and WN (Southwest Airlines).

```
data MWflightDelays;
   format flightDate MMDDYY8.;
   label flightDate='Date';
   input flightDate :MMDDYY8. AA CO DL F9 FL NW UA US WN;
```

```
   datalines;
02/01/07 14.9  7.1  7.9  8.5 14.8  4.5  5.1 13.4  5.1
02/02/07 14.3  9.6 14.1  6.2 12.8  6.0  3.9 15.3 11.4
02/03/07 23.0  6.1  1.7  0.9 11.9 15.2  9.5 18.4  7.6
02/04/07  6.5  6.3  3.9 -0.2  8.4 18.8  6.2  8.8  8.0
02/05/07 12.0 14.1  3.3 -1.3 10.0 13.1 22.8 16.5 11.5
02/06/07 31.9  8.6  4.9  2.0 11.9 21.9 29.0 15.5 15.2
02/07/07 14.2  3.0  2.1 -0.9 -0.6  7.8 19.9  8.6  6.4
02/08/07  6.5  6.8  1.8  7.7  1.3  6.9  6.1  9.2  5.4
02/09/07 12.8  9.4  5.5  9.3 -0.2  4.6  7.6  7.8  7.5
02/10/07  9.4  3.5  1.5 -0.2  2.2  9.9  3.1 12.5  3.0
02/11/07 12.9  5.4  0.9  6.8  2.1  7.9  3.7 10.7  5.6
02/12/07 34.6 15.9  1.8  1.0  4.5 10.2 14.0 19.1  4.9
02/13/07 34.0 16.0  4.4  6.1 18.3  9.1 30.2 46.3 50.6
02/14/07 21.2 45.9 16.6 12.5 35.1 23.8 40.4 43.6 35.2
02/15/07 46.6 36.3 23.9 20.8 30.4 24.3 30.3 59.9 25.6
02/16/07 31.2 20.8 15.2 20.1  9.1 12.9 22.9 36.4 16.4
;
```

The observations for a given date are the average flight delays in minutes for flights that departed from the midwestern United States. For example, on February 2, 2007, F9 (Frontier Airlines) flights departed 6.2 minutes late on average.

Creating a Multivariate Control Chart in a Phase I Situation

In a Phase I analysis you first perform a principal component analysis (PCA) of the data. Then you can use control charts to determine whether the data that you use to build the principal component model indicate a stable multivariate process. The MVPMONITOR procedure creates multivariate control charts from T^2 and SPE statistics computed from a principal component model that the MVPMODEL procedure produced. This example uses the model built in the section "Building a Principal Component Model" on page 930 in Chapter 13, "The MVPMODEL Procedure."

The following statements fit the model:

```
proc mvpmodel data=MWflightDelays ncomp=3 noprint
              out=mvpair outloadings=mvpairloadings;
   var AA CO DL F9 FL NW UA US WN;
run;
```

The NCOMP= option requests a principal component model that contains three principal components. The OUT= option creates a data set that contains the original data, the principal component scores, and the T^2 and SPE statistics. The OUTLOADINGS= data set contains the variances and loadings for the principal components.

The following statements produce the multivariate control charts:

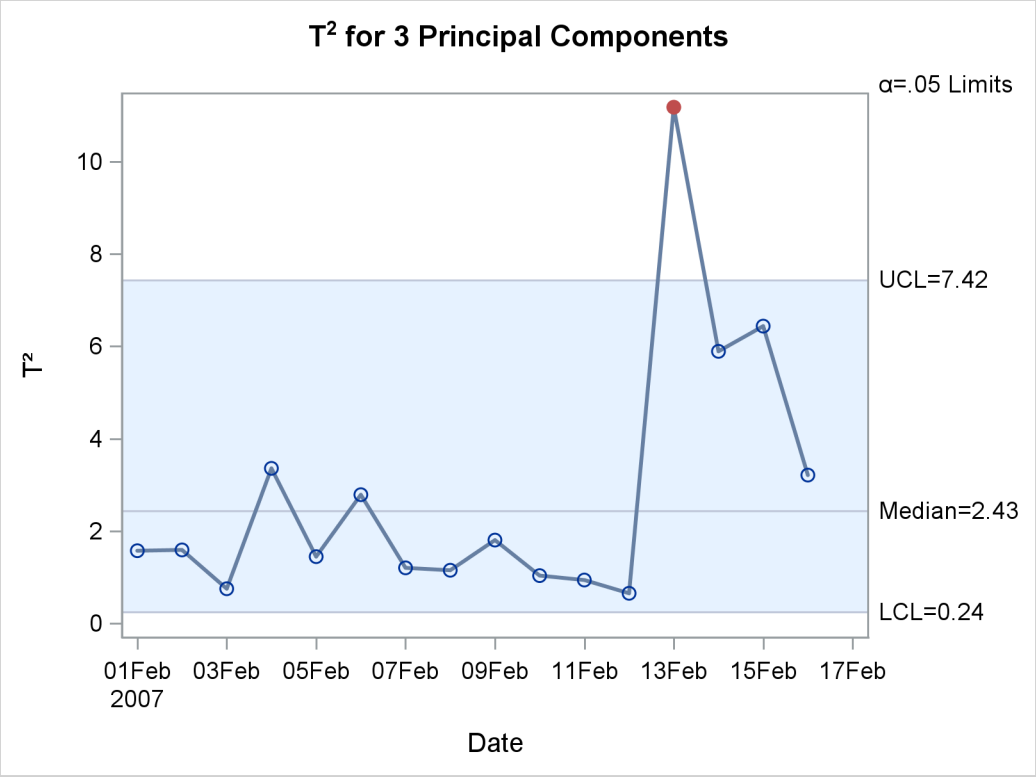
```
ods graphics on;
proc mvpmonitor history=mvpair loadings=mvpairloadings;
   time flightDate;
   tsquarechart / contributions;
   specchart / contributions;
run;
```

The HISTORY= option specifies the input data set. The LOADINGS= option specifies the data set that contains the principal component model information. The TSQUARECHART statement requests a T^2 chart,

and the SPECHART statement requests an SPE chart. The CONTRIBUTIONS options that are specified in the TSQUARECHART and SPECHART statements request contribution plots for all out-of-control points in the charts. The TIME statement specifies that the variable flightDate provide the chronological ordering of the observations.

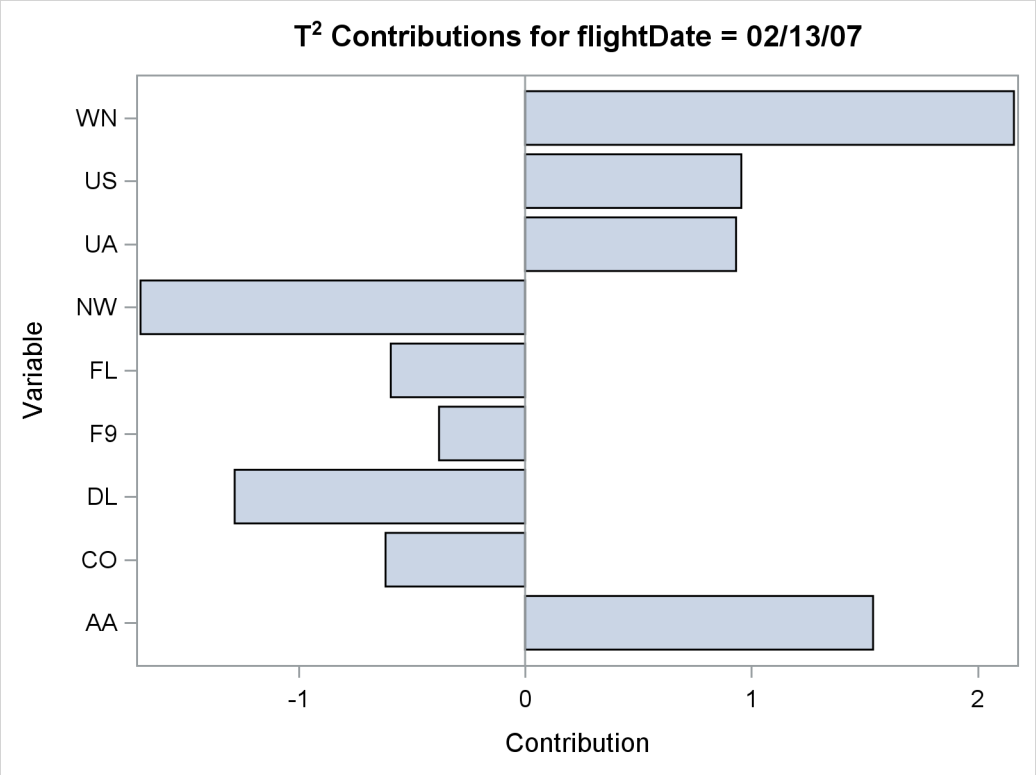
Figure 14.1 shows the T^2 chart.

Figure 14.1 Multivariate Control Chart for T^2 Statistics



The T^2 chart shows an out-of-control point on February 13, 2007. On this day, a strong winter storm battered the midwestern United States. To see which variables contributed to this statistic, you can use the contribution plot shown in Figure 14.2.

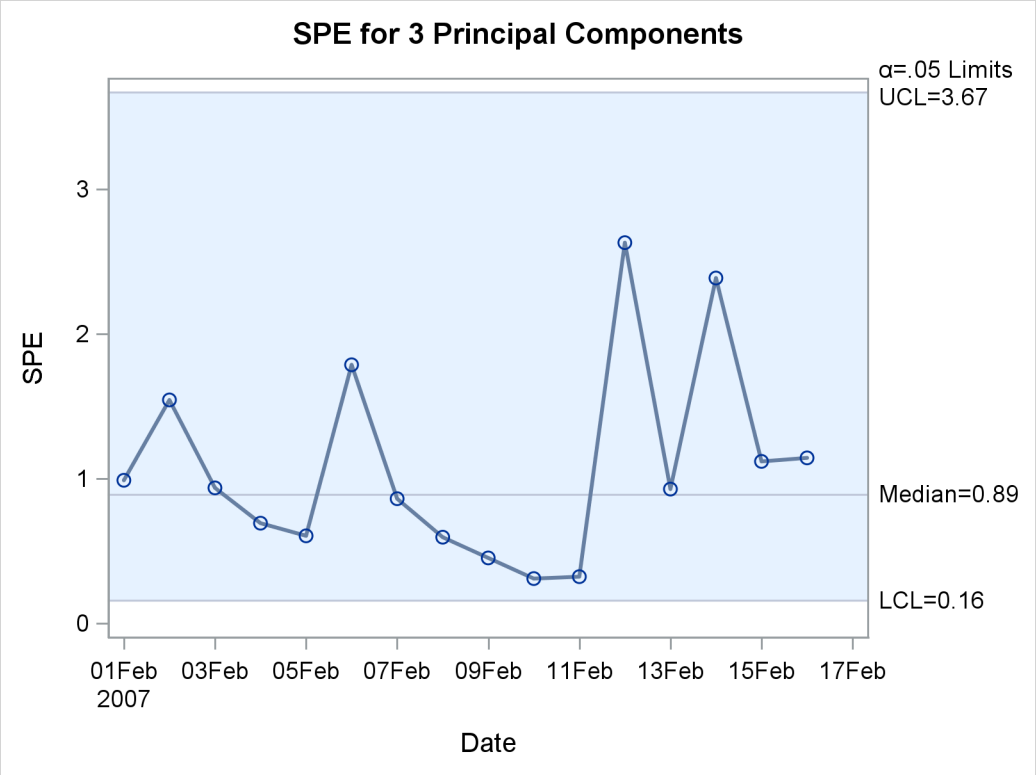
Figure 14.2 Contribution Plot



The contribution plot shows that the variables WN, AA, NW, and DL are the major contributors to the out-of-control point.

Figure 14.3 shows the SPE chart.

Figure 14.3 Multivariate Control Chart for SPE Statistics



There are no out-of-control points in the SPE chart. This indicates that the unusual point displayed in the T^2 chart represents a departure from the variation described by the principal component model that lies within the model hyperplane.

Syntax: MVPMONITOR Procedure

PROC MVPMONITOR < *options* > ;
BY *variables* ;
ID *variable* ;
SCORECHART < / *options* > ;
SERIES *variable* ;
SPECHART < / *options* > ;
TIME *variable* ;
TSQUARECHART < / *options* > ;

The following sections describe the PROC MVPMONITOR statement and then describe the other statements in alphabetical order.

PROC MVPMONITOR Statement

PROC MVPMONITOR < *options* > **;**

The PROC MVPMONITOR statement invokes the MVPMONITOR procedure and specifies input and output data sets. You can specify the following *options*:

DATA=*SAS-data-set*
specifies an input SAS data set that contains process measurement data for a Phase II analysis. If you specify a DATA= data set, you must also specify a LOADINGS= data set. You cannot specify the HISTORY= or TABLE= option together with the DATA= option. See the section "DATA= Data Set" on page 975 for details about DATA= data sets.

HISTORY=*SAS-data-set*
specifies an input SAS data set that contains process variable data augmented with principal component scores, multivariate summary statistics, and other calculated values. This data set is used for a Phase I analysis. Usually, a HISTORY= data set is created as an OUT= data set from the MVPMODEL procedure. You cannot specify the DATA= or TABLE= option with the HISTORY= option. See the section "HISTORY= Data Set" on page 975 for details about HISTORY= data sets.

LOADINGS=*SAS-data-set*
specifies an input SAS data set that contains eigenvalues, principal component loadings, and process variable means and standard deviations that are used to compute principal component scores and multivariate summary statistics for a Phase II analysis. Usually, the LOADINGS= data set is produced by the MVPMODEL procedure as an OUTLOADINGS= data set. See the section "LOADINGS= Data Set" on page 976 for details about LOADINGS= data sets.

MISSING=AVG | NONE
specifies how to handle observations that have missing process variable values in the DATA= data set. The option MISSING=AVG specifies that missing values for a given variable be replaced by the average of the nonmissing values for that variable. The default is MISSING=NONE, which excludes observations that have missing values for any of the process variables from the analysis.

OUTHISTORY=*SAS-data-set*
OUT=*SAS-data-set*
creates an output data set that contains all the original data from the input data set, principal component scores, and multivariate summary statistics. See the section "OUTHISTORY= Data Set" on page 978 for details. You can produce an OUTHISTORY= data set only when you specify a DATA= input data set.

PREFIX=*name*
specifies the prefix to use to identify variables that contain principal component scores in the HISTORY= data set. For example, if you specify PREFIX=ABC, PROC MVPMONITOR attempts to read the score variables ABC1, ABC2, ABC3, and so on. The default PREFIX= value is Prin, which is the default score variable prefix the that PROC MVPMODEL uses when it creates an OUT= data set. If you use an OUT= data set from MVPMODEL as a HISTORY= data set, the PREFIX= value must match the PROC MVPMODEL PREFIX= value that is specified when the OUT= data set is created.

NOTE: The number of characters in the prefix plus the number of digits that are required to enumerate the principal components must not exceed the maximum name length defined by the VALIDVARNAME= system option.

RPREFIX=*name*

specifies the prefix to use to identify variables that contain residuals in the HISTORY= data set. Residual variable names are formed by appending process variable names to the prefix. The default RPREFIX= value is R_, which is the default residual variable prefix that PROC MVPMODEL uses when it creates an OUT= data set. If you use an OUT= data set from PROC MVPMODEL as a HISTORY= data set, the RPREFIX= value must match the PROC MVPMODEL RPREFIX= value that is specified when the OUT= data set is created.

If the combined length of the residual prefix and a process variable name exceeds the maximum name length defined by the VALIDVARNAME= system option, characters are removed from the middle of the process variable name before it is appended to the residual prefix. For example, if you specify RPREFIX=*Residual_* (nine characters), the maximum variable name length is 32, and there is a process variable named PrimaryThermometerReading (25 characters), then two characters are dropped from the middle of the process variable name. The resulting residual variable name is Residual_PrimaryThermeterReading.

TABLE=*SAS-data-set*

specifies an input SAS data set that contains summary information from a score chart, SPE chart, or T^2 chart. You can produce a TABLE= data set by specifying the OUTTABLE= option in a SCORECHART, SPECHART, or TSQUARECHART statement. You can use a TABLE= input data set to display a previously computed control chart. You cannot specify the DATA= or HISTORY= option together with the TABLE= option. See the section "TABLE= Data Set" on page 977 for details.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MVPMONITOR to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the MVPMONITOR procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The values of the ID *variables* are displayed in tooltips associated with points on a T^2 or SPE chart when you create HTML output and specify the IMAGEMAP option in the ODS GRAPHICS statement. See Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*), for details.

SERIES Statement

SERIES *variable* **;**

The SERIES *variable* uniquely identifies a series of observations in the input data set to be plotted in a control chart. You must specify a SERIES statement when the input data set contains more than one observation that has the same TIME variable value. See Example 14.1 for an illustration of how to use the SERIES statement.

SCORECHART Statement

SCORECHART < */ options* > **;**

The SCORECHART statement produces univariate control charts of principal component scores. In a Phase I analysis, the scores are computed by PROC MVPMODEL and read from a HISTORY= data set. In a Phase II analysis, PROC MVPMONITOR computes the scores from process data contained in a DATA= data set and information about the principal component model that is specified in the LOADINGS= data set.

Table 14.1 summarizes the *options* available in the SCORECHART statement.

Table 14.1 SCORECHART Statement Options

Option	Description
COMP=	Specifies the principal components whose scores are plotted
EXCHART	Specifies that a score chart be displayed only if there are points lying outside the control limits
INTERVAL=	Specifies the time interval between consecutive positions on a score chart
NOCHART	Suppresses creation of score charts
NOHLABEL	Suppresses the horizontal axis label of a score chart
NOVLABEL	Suppresses the vertical axis label of a score chart
NPANELPOS=	Specifies the number of horizontal axis plotting positions per panel
ODSFOOTNOTE=	Adds a footnote to a score chart
ODSFOOTNOTE2=	Adds a secondary footnote to a score chart
ODSTITLE=	Specifies a title for a score chart
ODSTITLE2=	Specifies a secondary title for a score chart
OUTTABLE=	Creates a SAS data set that contains a summary of a score chart
OVERLAY=	Overlays separate series of observations or scores for multiple principal components in a single score chart

Table 14.1 *(continued)*

Option	Description
SERIESVALUE=	Specifies SERIES variable values that select sequences to be plotted
SIGMAS=	Specifies the multiples of a score's standard deviation used to compute score chart limits
TOTPANELS=	Specifies the number of panels used to display a score chart

The following options are available only in the SCORECHART statement. For detailed descriptions of options common to the SCORECHART, SPECHART, and TSQUARECHART statements, see the section "Common Chart Statement Options" on page 969.

COMP=*value-list* **| ALL**
specifies the principal components whose scores are plotted. You can specify the following values:

value-list plots the principal components whose numbers (1, 2, . . .) appear in *value-list*.

ALL plots all the principal components whose scores are in the input data set.

If you do not specify this option, scores for only the first principal component are plotted.

OVERLAY=COMP | SERIES
requests that multiple sequences of principal component scores be overlaid on a single control chart. You can specify the following values:

COMP overlays the scores for different principal components on the chart. The COMP= option determines which principal components are included.

SERIES overlays sequences of scores defined by the SERIES variable and selected by the SERIESVALUE= option on the chart. This value is applicable only when a TIME variable is specified and the input data set contains multiple scores for each principal component that have the same time value.

The value you specify in this option determines the number of separate control charts that are produced. For example, if you select p principal components and n SERIESVALUE= values, PROC MVPMONITOR produces

- n charts if you specify OVERLAY=COMP,
- p charts if you specify OVERLAY=SERIES, and
- $p \times n$ charts if you do not specify the OVERLAY= option.

SIGMAS=*k*
specifies the width of the control limits in terms of the multiple k of a score's standard deviation. By default, k=3.

SPECHART Statement

SPECHART < */ options* > **;**

The SPECHART statement produces a control chart of the squared prediction error (SPE) statistics based on a principal component model that the MVPMODEL procedure computes. In a Phase I analysis, the SPE statistics are computed by PROC MVPMODEL and read from the HISTORY= data set. In a Phase II analysis, the SPE statistics are computed by the MVPMONITOR procedure from information about the principal component model that is specified in the LOADINGS= data set.

Table 14.2 summarizes the *options* available in the SPECHART statement. For detailed descriptions, see the section "Common Chart Statement Options" on page 969.

Table 14.2 SPECHART Statement Options

Option	Description
ALPHA=	Specifies the probability used to compute SPE chart limits
CONTRIBUTIONS	Creates contribution plots for points outside control limits
EXCHART	Specifies that an SPE chart be displayed only if there are points lying outside control limits
INTERVAL=	Specifies the time interval between consecutive positions on an SPE chart
NOCHART	Suppresses creation of an SPE chart
NOHLABEL	Suppresses the horizontal axis label of an SPE chart
NOVLABEL	Suppresses the vertical axis label of an SPE chart
NPANELPOS=	Specifies the number of horizontal axis plotting positions per panel
ODSFOOTNOTE=	Adds a footnote to an SPE chart
ODSFOOTNOTE2=	Adds a secondary footnote to an SPE chart
ODSTITLE=	Specifies a title for an SPE chart
ODSTITLE2=	Specifies a secondary title for an SPE chart
OUTTABLE=	Creates a SAS data set that contains a summary of an SPE chart
OVERLAY	Specifies that separate sequences of observations be plotted in a single SPE chart
SERIESVALUE=	Specifies SERIES variable values selecting sequences to be plotted
TOTPANELS=	Specifies the number of panels used to display an SPE chart

TIME Statement

TIME *variable* **;**

The TIME *variable* is a numeric variable that provides the chronological order or time values for measurements in a DATA=, HISTORY=, or TABLE= input data set. The values of the TIME *variable* are displayed on the horizontal axis of control charts. If no TIME statement is specified, the observation numbers from the input data set are displayed on the horizontal axis.

When the input data set contains more than one observation with the same TIME value, you must identify the sequences of points to be plotted on the control charts by specifying a SERIES variable.

TSQUARECHART Statement

TSQUARECHART < */ options* > **;**

The TSQUARECHART statement produces a T^2 chart based on a principal component model that the MVPMODEL procedure computes. In a Phase I analysis, the T^2 statistics are computed by PROC MVPMODEL and read from the HISTORY= data set. In a Phase II analysis, the T^2 statistics are computed by the MVPMONITOR procedure from information about the principal component model specified in the LOADINGS= data set.

Table 14.3 summarizes the *options* available in the TSQUARECHART statement. For detailed descriptions, see the section "Common Chart Statement Options" on page 969.

Table 14.3 TSQUARECHART Statement Options

Option	Description
ALPHA=	Specifies the probability used to compute T^2 chart limits
CONTRIBUTIONS	Creates contribution plots for points outside the control limits
EXCHART	Specifies that a T^2 chart be displayed only if there are points lying outside the control limits
INTERVAL=	Specifies the time interval between consecutive positions on a T^2 chart
LIMITDIST=	Specifies the distribution that is used to compute the control limits
NOCHART	Suppresses creation of a T^2 chart
NOHLABEL	Suppresses the horizontal axis label of a T^2 chart
NOVLABEL	Suppresses the vertical axis label of a T^2 chart
NPANELPOS=	Specifies the number of horizontal axis plotting positions per panel
ODSFOOTNOTE=	Adds a footnote to a T^2 chart
ODSFOOTNOTE2=	Adds a secondary footnote to a T^2 chart
ODSTITLE=	Specifies a title for a T^2 chart
ODSTITLE2=	Specifies a secondary title for a T^2 chart
OUTTABLE=	Creates a SAS data set that contains a summary of a T^2 chart
OVERLAY	Specifies that separate sequences of observations be plotted on a single T^2 chart
SERIESVALUE=	Specifies SERIES variable values selecting sequences to be plotted
TOTPANELS=	Specifies the number of panels used to display a T^2 chart

LIMITDIST=BETA | CHISQ | F

specifies the distribution that is used to compute control limits for the T^2 chart. You can specify the following values:

BETA selects a beta distribution

CHISQ selects a χ^2 distribution

F selects an F distribution

See the section "Computing T^2 Control Limits" on page 972 for a description of how T^2 chart control limits are computed.

Common Chart Statement Options

You can specify the following *options* after a slash (/) in a SCORECHART, SPECHART, or TSQUARECHART statement (unless noted otherwise).

ALPHA=*value*

specifies the probability used to compute control limits for an SPE or T^2 chart. If you specify ALPHA=α, the control limits are computed so that the probability is α that the statistic exceeds its control limits. The *value* of α can range from 0 to 1. By default, α is 0.05. This option is not available in the SCORECHART statement.

CONTRIBUTIONS < (*contribution-options***) >**

creates a contribution plot for each point that falls outside the control limits of the chart. You can specify the following *contribution-options* in parentheses:

MAXNPLOTS=*n*

specifies the maximum number, *n*, of contribution plots to produce. When *n* is less than the number of points outside the control limits, contribution plots are produced for the first *n* out-of-control points.

MAXNVARS=*n*

specifies that only the *n* contributions with the greatest magnitudes be displayed in the contribution plots. For each out-of-control point, the *n* variables that contribute the most to that value of the statistic are displayed. By default, all variable contributions are displayed.

This option is not available in the SCORECHART statement.

EXCHART

specifies that a control chart be displayed only when one or more points lie outside the control limits.

INTERVAL=DAY | DTDAY | HOUR | MINUTE | MONTH | QTR | SECOND

specifies the natural time interval between consecutive TIME variable positions when a time, date, or datetime format is associated with the TIME variable. By default, the INTERVAL= option uses the number of positions per panel that you specify with the NPANELPOS= option. The default time interval keywords for various time formats are shown in Table 14.4.

Table 14.4 Default Time Intervals

Time Format	Default Interval
DATE	DAY
DATETIME	DTDAY
DDMMYY	DAY
HHMM	HOUR
HOUR	HOUR
MMDDYY	DAY
MMSS	MINUTE
MONYY	MONTH
TIME	SECOND
TOD	SECOND
WEEKDATE	DAY
WORDDATE	DAY
YYMMDD	DAY
YYQ	QTR

You can use the INTERVAL= option to modify the effect of the NPANELPOS= option, which specifies the number of positions per panel. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the TIME variable without having to associate a different format with the TIME variable.

For example, suppose your formatted time values span 100 days and a datetime format is associated with the TIME variable. Since the default interval for the datetime format is dtday and because NPANELPOS=50 by default, the chart is displayed with two panels.

Now suppose your data span 100 hours and a datetime format is associated with the TIME variable. The chart for these data is created in a single panel, but the data occupy only a small fraction of the chart because the scale of the data (hours) does not match that of the horizontal axis (days). If you specify INTERVAL=HOUR, the horizontal axis is scaled for 50 hours, matching the scale of the data, and the chart is displayed with two panels.

NOCHART

suppresses display of the control chart. You can use the NOCHART option with the CONTRIBUTIONS option to produce contribution plots for out-of-control points without displaying the control chart. You can use the NOCHART option with the OUTTABLE= option to save a summary of the control chart to an output data set without displaying the chart.

NOHLABEL

suppresses the horizontal axis label in the control chart.

NOVLABEL

suppresses the vertical axis label in the control chart.

NPANELPOS=*n*

NPANEL=*n*

specifies the number of horizontal axis plotting positions per panel in the chart. You typically specify the NPANELPOS= option to display more points in a panel than the default number, which is 50.

You can specify a positive or negative value for *n*. The absolute value of *n* must be at least 5. If *n* is positive, the number of positions is adjusted so that it is approximately equal to *n* and so that all panels display approximately the same number of positions. If *n* is negative, then no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string***'**

adds a footnote to the chart. If you specify the FOOTNOTE (or FOOTNOTE1) keyword, the value of the SAS FOOTNOTE statement is used as the chart footnote. If you specify a quoted string, that string is used as the footnote. The quoted string can contain the following escaped characters, which are replaced with the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the TIME variable has no label).

ODSFOOTNOTE2=FOOTNOTE2 | '*string***'**

adds a secondary footnote to the chart. If you specify the FOOTNOTE2 keyword, the value of the SAS FOOTNOTE2 statement is used as the secondary chart footnote. If you specify a quoted string, that string is used as the secondary footnote. The quoted string can contain the following escaped characters, which are replaced with the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the TIME variable has no label).

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | '*string***'**

specifies a title for the chart. You can specify the following values:

TITLE (or TITLE1)	uses the value of the SAS TITLE statement as the chart title.
NONE	suppresses all titles from the chart.
DEFAULT	uses the default title.

If you specify a quoted string, that string is used as the graph title. The quoted string can contain the following escaped characters, which are replaced with the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the analysis variable has no label).

ODSTITLE2=TITLE2 | '*string***'**

specifies a secondary title for the chart. If you specify the TITLE2 keyword, the value of the SAS TITLE2 statement is used as the secondary chart title. If you specify a quoted string, that string is used as the secondary title. The quoted string can contain the following escaped characters, which are replaced with the values indicated:

\n	is replaced by the TIME variable name.
\l	is replaced by the TIME variable label (or name if the analysis variable has no label).

OUTTABLE=*SAS-data-set*
creates an output SAS data set that contains the information plotted in the control chart, including the statistic values and control limits. See the section "OUTTABLE= Data Set" on page 979 for a description of the OUTTABLE= data set.

OVERLAY
specifies that the separate sequences of observations defined by the SERIES variable and selected by the SERIESVALUE= option be plotted in a single control chart. By default, each sequence is plotted in a separate chart. The OVERLAY option is applicable only when a TIME variable is specified and the input data set contains multiple observations that have the same time value.

This option is not available in the SCORECHART statement. You can use the OVERLAY= option in the SCORECHART statement to overlay control charts of principal component scores.

SERIESVALUE=*value-list*
specifies a list of values of the SERIES variable that define one or more sequences of observations to be plotted. If the SERIES variable is a character variable, the *value-list* must be a list of quoted strings. By default, a series is plotted for each unique value of the SERIES variable. The SERIESVALUE= option is applicable only when a TIME variable is specified and the input data set contains multiple observations that have the same time value.

TOTPANELS=*n*
specifies the number of panels to use to display the chart. By default, the number of panels is determined by the value that you specify in the NPANELPOS= option. If you specify both the TOTPANELS= and NPANELPOS= options, the TOTPANELS= value takes precedence.

Details: MVPMONITOR Procedure

Computing T^2 Control Limits

The control limits for the T^2 chart are the same for all the T^2 statistics on the chart. The control limits are computed based on one of the following distributions:

- a beta distribution

$$T_i^2 \sim \frac{(n-1)^2}{n} B\left(\frac{j}{2}, \frac{n-j-1}{2}\right) \qquad j \geq 2, n \geq j+1$$

- a χ^2 distribution

$$T_i^2 \sim \chi^2(j) \qquad j \geq 2, n \geq j+1$$

- an F distribution

$$T_i^2 \sim \frac{j(n+1)(n-1)}{n(n-j)} F(j, n-j) \qquad j \geq 2,\ n \geq j+1$$

where i is the observation, j is the number of principal components in the model, and n is the number of observations used to build the principal component model.

The upper control limit is computed as the $(1-\frac{\alpha}{2})$ quantile of the distribution, and the lower control limit is computed as the $\frac{\alpha}{2}$ quantile. You can specify the ALPHA= option in the TSQUARECHART statement to specify α.

You can specify the LIMITDIST= option in the TSQUARECHART statement to select the distribution that is used to compute the control limits. A beta distribution is used by default. See Tracy, Young, and Mason (1992) for a discussion of the conditions under which each distribution is applicable.

See the section "Computing the T^2 and SPE Statistics" on page 944 for details of computing the T^2 statistic based on a principal component model.

Computing SPE Control Limits

The SPE chart plots the sum of squares of the residuals from the principal component model. If either $j = p$ or the data matrix has rank less than p, then the SPE statistic is not defined and an SPE chart is not produced. The SPE statistic for observation i is denoted as

$$Q_i = \sum_{k=1}^{p} e_{ik}^2$$

where p is the number of variables and e_{ik} is the ith observation for the kth variable in the error matrix, $\mathbf{E}$, in the principal component model

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E}$$

The distribution of Q_i has been approximated in the literature under different conditions. Two methods of computing control limits for Q_i are implemented by the MVPMONITOR procedure. One method is used when the data that are used to build the principal component model consist of a single measurement per time point. The other method is used when there are multiple measurements per time point (Jensen and Solomon 1972; Nomikos and MacGregor 1995).

One Observation per Time Point

When there is a single observation at each time point, the data matrix $\mathbf{X}$ is $n \times p$, with exactly one observation at each time point in the input data set. The derivation of the control limits uses the central limit theorem approach of Jensen and Solomon (1972). They begin by defining $\theta_i = \sum_{k=j+1}^{p} \lambda_k^i$, $i = 1, 2, 3$, where λ_k is the kth eigenvalue from the principal component model.

Then the quantity

$$z = \frac{\theta_1 \left[\left(\frac{Q}{\theta_1} \right)^{h_0} - 1 - \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]}{\sqrt{2\theta_2 h_0^2}}$$

is distributed $N(0,1)$ as $n \to \infty$, where h_0 is defined as $1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$. The upper control limit for all Q_i is then computed by

$$Q_{i,1-\frac{\alpha}{2}} = \theta_1 \left[1 + \frac{z_{(1-\alpha/2)}\sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}}$$

where $z_{(1-\alpha/2)}$ is the $(1 - \frac{\alpha}{2})$ percentile of the standard normal distribution. The lower control limit is obtained similarly by using $\frac{\alpha}{2}$. You can specify α by using the ALPHA= option in the SPECHART statement.

Multiple Observations per Time Point

When there are multiple observations at a time value in an input data set, a different approximation of the SPE distribution is used to compute control limits. The approximate distribution at time i is the scaled chi-square distribution,

$$\frac{s_i^2}{2\bar{x}_i} \chi_{\frac{2\bar{x}_i^2}{s_i^2}}$$

where $\bar{x}_i$ and s_i^2 are the mean and variance, respectively, of the SPE statistics at time i. The upper control limit for all observations at time point i is computed as the $(1 - \frac{\alpha}{2})$ percentile of the scaled chi-square distribution:

$$SPE_{i,1-\frac{\alpha}{2}} = \frac{s_i^2}{2\bar{x}_i} \chi_{\frac{2\bar{x}_i^2}{s_i^2},1-\frac{\alpha}{2}}$$

Similarly the lower control limit is computed from the $\frac{\alpha}{2}$ percentile. You can specify α by using the ALPHA= option in the SPECHART statement.

For more information about the distribution approximations, see Nomikos and MacGregor (1995) and Jackson and Mudholkar (1979).

Contribution Plots

One way to diagnose the behavior of out-of-control points in multivariate charts is to use contribution plots (Miller, Swanson, and Heckler 1998). These plots tell you which variables contribute to the distance between the points in an SPE or T^2 chart and the sample mean of the data.

A contribution plot is a bar chart of the contributions of the process variables to the statistic. For the ith SPE statistic, the contribution of the kth variable is the kth entry of the vector $\mathbf{e}_i$, which is computed as

$$\mathbf{e}_i = \mathbf{x}_i \left(\mathbf{I} - \mathbf{P}_j \mathbf{P}'_j \right)$$

where $\mathbf{e}_i$ is the vector of errors from the principal component model for observation i and $\mathbf{x}_i$ is the ith observation. The contributions to the ith T^2 statistic are computed in the same way as the entries of the vector

$$\mathbf{T}_i^2 = \mathbf{x}_i \mathbf{P}_j \mathbf{L}^{-1} \mathbf{P}'_j$$

where $\mathbf{P}_j$ is the matrix of the first j eigenvectors and $\mathbf{L}$ is the diagonal matrix of the first j eigenvalues.

Input Data Sets

The MVPMONITOR procedure accepts a single primary input data set of one of three types.

- A DATA= data set contains new process data to be analyzed by using an existing PCA model (Phase II analysis).
- A HISTORY= data set contains process data and the accompanying scores, residuals, and statistics produced by applying a PCA model. The process data can be the original data that was used to create the model (Phase I analysis) or subsequent data that was analyzed by using a previously created model (Phase II analysis).
- A TABLE= data set contains a summary of score charts, SPE charts, or T^2 charts, which consists of the statistics, control limits, and other information.

These options are mutually exclusive. If you do not specify an option identifying a primary input data set, PROC MVPMONITOR uses the most recently created SAS data set as a DATA= data set.

When you specify a DATA= data set, you must also specify a LOADINGS= data set that contains loadings and other information describing the PCA model. When you specify a HISTORY= data set, you must also specify a LOADINGS= data set if you specify the CONTRIBUTIONS option in a TSQUARECHART statement.

DATA= Data Set

A DATA= data set provides the process measurement data for a Phase II analysis. In addition to the process variables, a DATA= data set can include the following:

- BY variables
- ID variables
- a SERIES variable
- a TIME variable

When you specify a DATA= data set, you must also specify a LOADINGS= data set that contains the loadings for the principal component model that describes the variation of the process. These loadings are used to score the new data from the DATA= data set. The process variables in the LOADINGS= data set must have the same names as those in the DATA= data set.

HISTORY= Data Set

A HISTORY= data set provides the input data set for a Phase I or Phase II analysis. In addition to the original process variables, it contains principal component scores, residuals, SPE and T^2 statistics, and a count of the observations that are used to construct the principal component model, as summarized in Table 14.5.

Table 14.5 Variables in the HISTORY= Data Set

Variable	Description
Prin1–Prin*j*	Principal component scores
R_*var*1–R_*varp*	Residuals
NOBS	Number of observations used to build the principal component model
SPE	Squared prediction error (SPE)
TSQUARE	T^2 statistic computed from principal component scores

A HISTORY= data set must include variables that contain principal component scores. The score variables names must consist of a common prefix followed by the numbers 1, 2, ..., *j*, where *j* is the number of principal components. By default, the common prefix is Prin. You can use the PREFIX= option to specify another prefix for score variables.

If the number of principal components is less than the total number of process variables, the HISTORY= data set should also contain residual variables. A residual variable name consists of a common prefix followed by the corresponding process variable name. The default residual variable prefix is R_. For example, if the process variables are A, B, and C, the default residual variable names are R_A, R_B, and R_C. You can use the RPREFIX= option to specify a different residual variable prefix.

NOTE: Usually you create a HISTORY= data set by specifying the PROC MVPMODEL OUT= option or the PROC MVPMONITOR OUTHISTORY= option. If the PREFIX= or RPREFIX= option is used when such an output data set is created, you must specify the same prefixes to identify the score and residual variables when you read it as a HISTORY= data set.

LOADINGS= Data Set

The LOADINGS= data set contains the following information about the principal component model:

- eigenvalues of the correlation or covariance matrix used to construct the model
- principal component loadings
- process variable means used to center the variable values
- process variable standard deviations used to scale the variable values

You can produce a LOADINGS= data set by using the PROC MVPMODEL OUTLOADINGS= option. Table 14.6 lists the variables that are required in a LOADINGS= data set.

Table 14.6 Variables in the LOADINGS= Data Set

Variable	Description
VALUE	The value contained in *process variables* for a given observation
NOBS	Number of observations used to build the principal component model
_PC_	Principal component number; 0 for the observation that contains eigenvalues
process variables	Values associated with the process variables

Valid values for the _VALUE_ variable are as follows:

EIGEN	eigenvalues from the principal component analysis
LOADING	principal component loadings
MEAN	process variable means
STD	process variable standard deviations

The LOADINGS= data set contains one EIGEN observation and *j* LOADING observations, where *j* is the number of principal components in the model. The presence of a MEAN observation indicates that the process variables were centered when the principal component model was constructed, and the presence of a STD observation indicates that the process variables were scaled when the principal component model was constructed. The means and standard deviations are used to center and scale new data in a Phase II analysis.

TABLE= Data Set

A TABLE= data set contains a summary of one or more score charts, SPE charts, or T^2 control charts. Usually, you create a TABLE= data set by specifying the OUTTABLE= option in a SCORECHART, SPECHART, or TSQUARECHART statement. Each type of TABLE= data set contains different variables, and when you specify a TABLE= data set you can only specify chart statements of the corresponding type. For example, if you use a TABLE= data set that contains SPE chart summary data, you cannot specify a SCORECHART or TSQUARECHART statement.

You can use a TABLE= data set to display previously created control charts or to specify custom control limits by computing your own _LCL_ and _UCL_ values.

Table 14.7, Table 14.8, and Table 14.9 list the variables that are contained in the three types of TABLE= data set.

NOTE:

1. SPE chart and T^2 chart TABLE= data sets contain one observation per time value. Score chart TABLE= data sets contain one observation for each principal component per time value.
2. SPE chart and T^2 chart TABLE= data sets contain residual variables corresponding to the process variables. Each residual variable has the same name as the corresponding process variable

Table 14.7 Score Chart TABLE= Data Set Variables

Variable	Description
COMP	Principal component number
EXLIM	Flag that indicates control limit was exceeded
LCL	Lower control limit
MEAN	Center line
SCORE	Principal component score
series	Optional SERIES variable
SIGMAS	Multiple of score standard deviation used to compute control limits
time	Optional TIME variable
UCL	Upper control limit

Table 14.8 SPE Chart TABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Flag to indicate control limit was exceeded
LCL	Lower control limit
MEDIAN	Center line
residuals	Residual variables
series	Optional SERIES variable
SPE	Squared prediction error (SPE) statistic
time	Optional TIME variable
UCL	Upper control limit

Table 14.9 T^2 Chart TABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Flag to indicate control limit was exceeded
LCL	Lower control limit
MEDIAN	Center line
residuals	Residual variables
series	Optional SERIES variable
time	Optional TIME variable
TSQUARE	T^2 statistic (TSQUARECHART statement only)
UCL	Upper control limit

Output Data Sets

OUTHISTORY= Data Set

The OUTHISTORY= data set contains all the variables in the input data set plus new variables that contain the principal component scores, residuals, and other computed values listed in Table 14.10.

Table 14.10 Computed Variables in the OUTHISTORY= Data Set

Variable	Description
Prin1–Prin$j$	Principal component scores
R_var1–R_$varp$	Residuals
NOBS	Number of observations used in the analysis
SPE	Squared prediction error (SPE)
TSQUARE	T^2 statistic computed from principal component scores

The names of the score variables are formed by concatenating the value given by the PREFIX= option (or the default Prin, if PREFIX= is not specified) and the numbers 1, 2, ..., j, where j is the number of principal components in the model.

The names of the residual variables are formed by concatenating the value given by the RPREFIX= option (or the default R_, if RPREFIX= is not specified) and the names of the process variables used in the analysis. Residual variables are created only when the number of principal components in the model is less than the number of process measurement variables in the input data set.

OUTTABLE= Data Set

You can save control chart statistics, control limits, and related information by specifying the OUTTABLE= option in a SCORECHART, SPECHART, or TSQUARECHART statement. Each chart statement produces OUTTABLE= data sets containing different variables. Table 14.11, Table 14.12, and Table 14.13 list the variables that are contained in the three types of OUTTABLE= data set.

Table 14.11 Score Chart OUTTABLE= Data Set Variables

Variable	Description
COMP	Principal component number
EXLIM	Flag to indicate control limit was exceeded
LCL	Lower control limit
MEAN	Center line
SCORE	Principal component score
series	SERIES variable, if specified
SIGMAS	Multiple of score standard deviation used to compute control limits
time	TIME variable, if specified
UCL	Upper control limit

Table 14.12 SPE Chart OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Flag to indicate control limit was exceeded
LCL	Lower control limit
MEDIAN	Center line
NCOMP	Number of principal components in the model
residuals	Residual variables
series	SERIES variable, if specified
SPE	Squared prediction error (SPE) statistic
time	TIME variable, if specified
UCL	Upper control limit

Table 14.13 T^2 Chart OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Flag to indicate control limit was exceeded
LCL	Lower control limit
MEDIAN	Center line
NCOMP	Number of principal components in the model
residuals	Residual variables
series	SERIES variable, if specified
time	TIME variable, if specified
TSQUARE	T^2 statistic
UCL	Upper control limit

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The MVPMONITOR procedure assigns a name to each graph that it creates using ODS Graphics. You can use these names to refer to the graphs when you use ODS. The graph names are listed in Table 14.14.

Table 14.14 ODS Graphics Produced by PROC MVPMONITOR

ODS Graph Name	Plot Description	Statement or Option
ContributionPlot	Contribution plot	CONTRIBUTIONS option
ScoreChart	control chart of principal component scores	SCORECHART statement
SPEChart	Squared prediction error chart	SPECHART statement
TSquareChart	T^2 chart	TSQUARECHART statement

Examples: MVPMONITOR Procedure

The following examples use an airline flight delay data set similar to the one described in the section "Getting Started: MVPMONITOR Procedure" on page 958. The following statements create a data set named flightDelays, which contains average flight delays for each region of the continental United States:

```
data flightDelays;
   label flightDate='Date';
   format flightDate MMDDYY8.;
   input flightDate :MMDDYY8. region $ AA CO DL F9 FL NW UA US WN;
   datalines;
02/01/07 MW 14.9  7.1  7.9  8.5 14.8  4.5  5.1 13.4  5.1
02/01/07 NE 15.7  7.1  8.6  6.3 14.6  6.2  7.0 11.0  6.4
02/01/07 NW 17.8  2.6  6.1 28.8 11.6  6.1 11.6 27.3  3.7
02/01/07 SC 19.9  8.3 13.9  4.9 25.8 15.3  9.0 15.1 12.8
02/01/07 SE 16.1  1.9  8.7  8.7 15.1 18.3  4.0 10.4  6.5
02/01/07 SW 19.3  7.8  4.8 11.5 34.7  7.4  7.3 12.0  5.6
02/02/07 MW 14.3  9.6 14.1  6.2 12.8  6.0  3.9 15.3 11.4

   ... more lines ...

02/16/07 SE 29.4 13.5 16.8 19.7 11.4 10.4 27.7 34.2 20.8
02/16/07 SW 28.1 18.4 17.1 25.5  8.0 15.2 30.4 26.6 20.8
;
```

Example 14.1: Combining Data from Peer Processes

In some situations you might want to build a common principal component model by combining data from multiple peer processes that have similar patterns of stable variation. This enables you to borrow strength from the data. A common set of control limits is then computed for each peer process.

This example uses observations from all regions in the continental United States at each time value to construct a common principal component model. It then applies the model to flight data for one region.

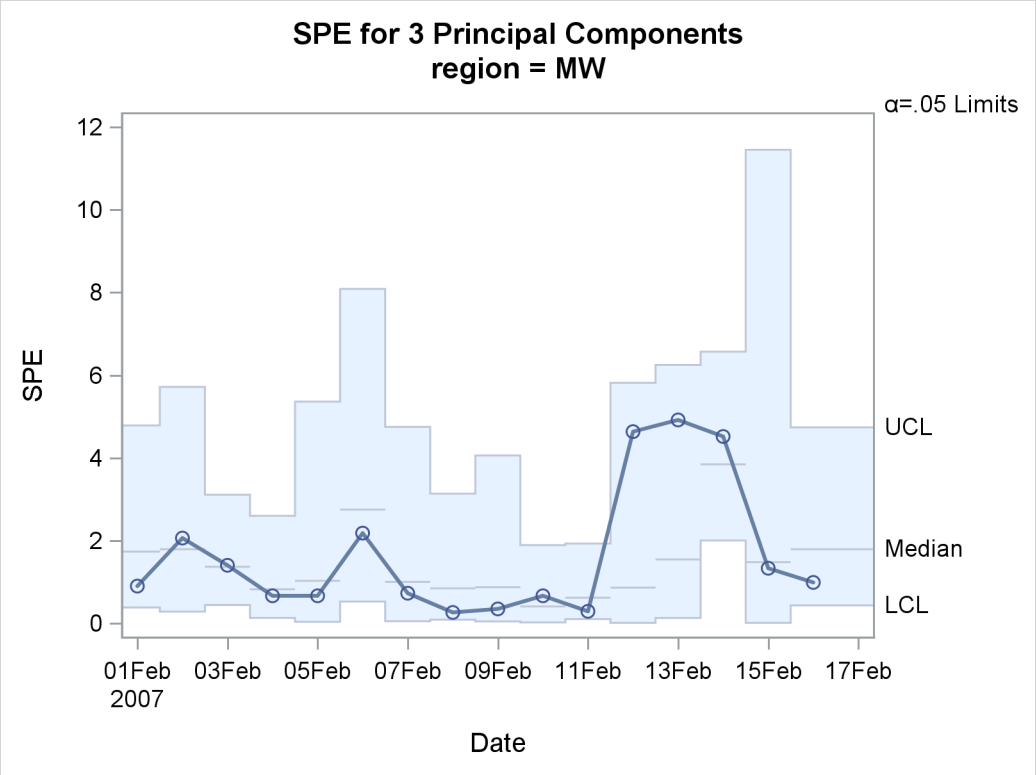
The following statements create a principal component model that contains three principal components from the flightDelays data set and apply the model to data for the Midwest region:

```
proc mvpmodel data=flightDelays ncomp=3 noprint
              out=mvpair outloadings=mvpairloadings;
   var AA CO DL F9 FL NW UA US WN;
run;
proc mvpmonitor history=mvpair loadings=mvpairloadings;
   time flightDate;
   series region;
   spechart / seriesvalue='MW';
   tsquarechart / seriesvalue='MW';
run;
```

The flightDelays data set contains observations from all continental United States regions, with multiple observations (one for each region) at each time point as defined by the flightDate variable. The OUTLOADINGS= data set that is produced by PROC MVPMODEL contains the model information. The SERIES statement specifies region as the variable that identifies sequences of related observations. The SERIESVALUE= option selects the Midwest region statistics to be plotted.

The resulting SPE chart is shown in Output 14.1.1.

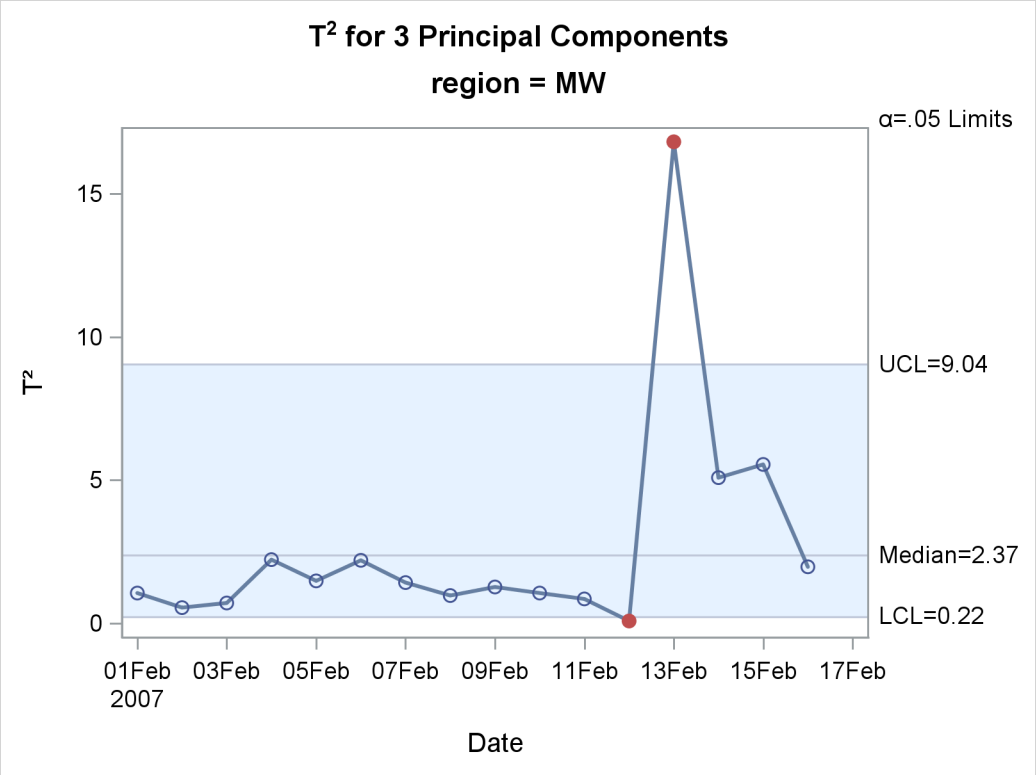
Output 14.1.1 Multivariate Control Chart for SPE Statistics



The control limits for the SPE chart are computed differently from a case with a single observation per time value, such as the chart shown in Figure 14.3. The control limits are based on different reference distributions for the SPE statistics in addition to different approximations to the reference distribution. See the section "Computing SPE Control Limits" on page 973 for more information.

The T^2 chart is shown in Output 14.1.2.

Output 14.1.2 Multivariate Control Chart for T^2 Statistics



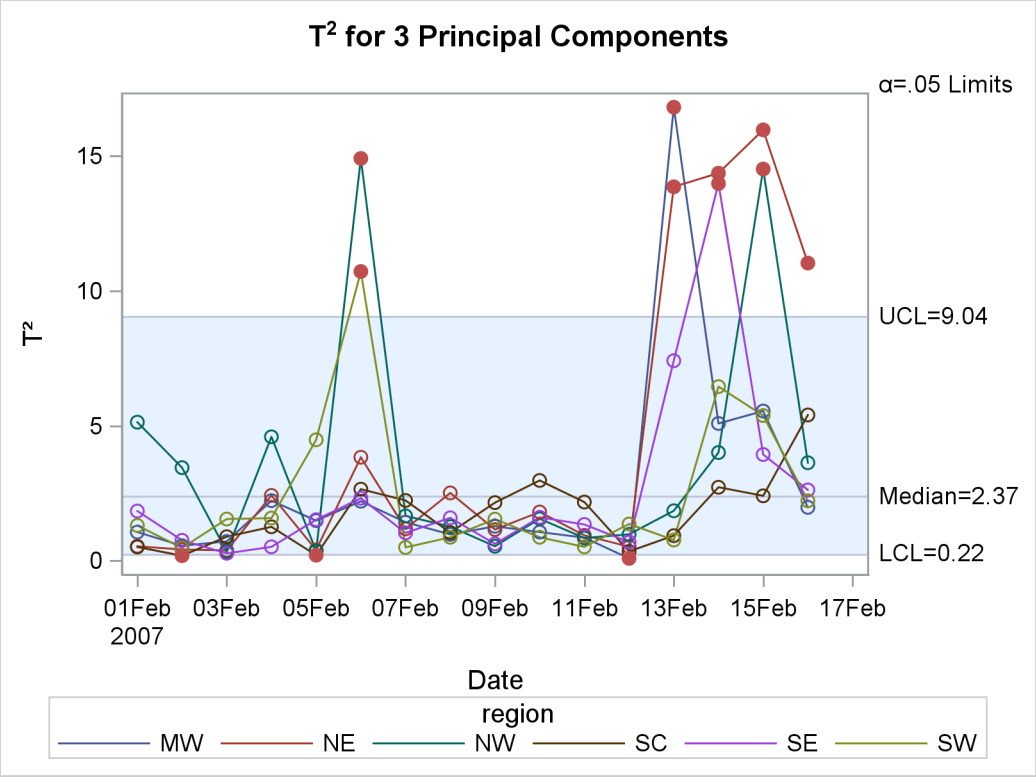
Compare the T^2 chart in Output 14.1.2 to the one in Figure 14.1. Both charts display T^2 statistics for the same flight delays from the Midwest region, but the charts are different because in this example the principal component model was constructed with data from all regions of the continental United States.

You can produce control charts for all the peer processes (regions in this example) by omitting the SERIES-VALUE= option. The following statements illustrate this approach:

```
proc mvpmonitor history=mvpair;
   time flightDate;
   series region;
   specchart;
   tsquarechart / overlay;
run;
```

By default, a separate control chart is created for each distinct value of the SERIES variable. The separate SPE charts for each region are not shown. The OVERLAY option in the TSQUARECHART statement specifies that the sequences for each region be plotted on a single T^2 chart, which is shown in Output 14.1.3.

Output 14.1.3 Overlaid T^2 Charts by Region

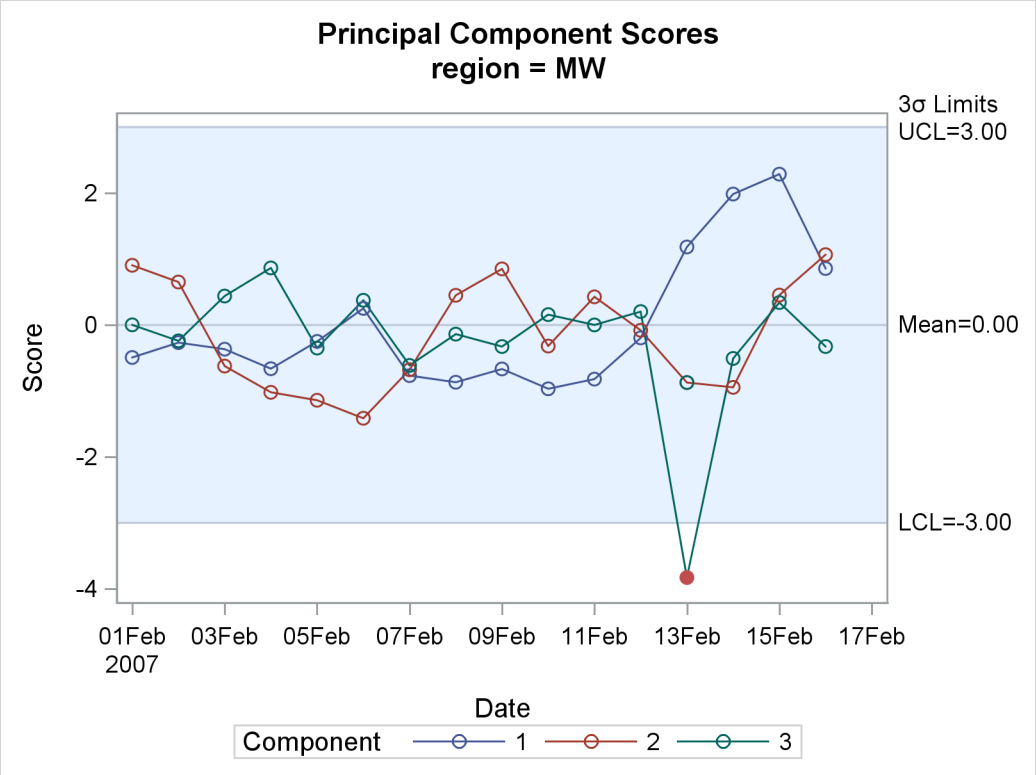


You can produce univariate control charts of standardized principal component scores by using the SCORECHART statement. The following statements produce control charts for the three principal components in the model and for each region:

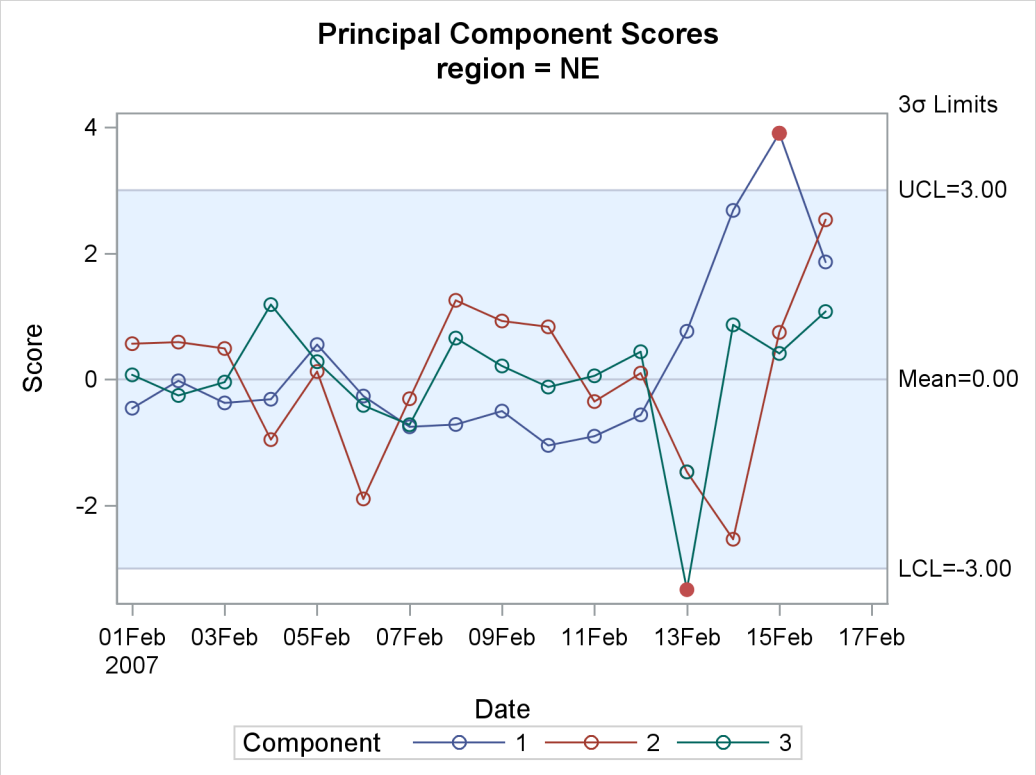
```
proc mvpmonitor history=mvpair loadings=mvpairloadings;
   time flightDate;
   series region;
   scorechart / comp=all overlay=comp;
run;
```

The COMP=ALL option requests score charts for all the principal components in the model. The OVERLAY=COMP option overlays the scores for each component in a single control chart. A separate chart is produced for each region. Output 14.1.4 and Output 14.1.5 show the score charts for the Midwest and Northeast regions, respectively.

Output 14.1.4 Score Charts for the Midwest Region



Output 14.1.5 Score Charts for the Northeast Region

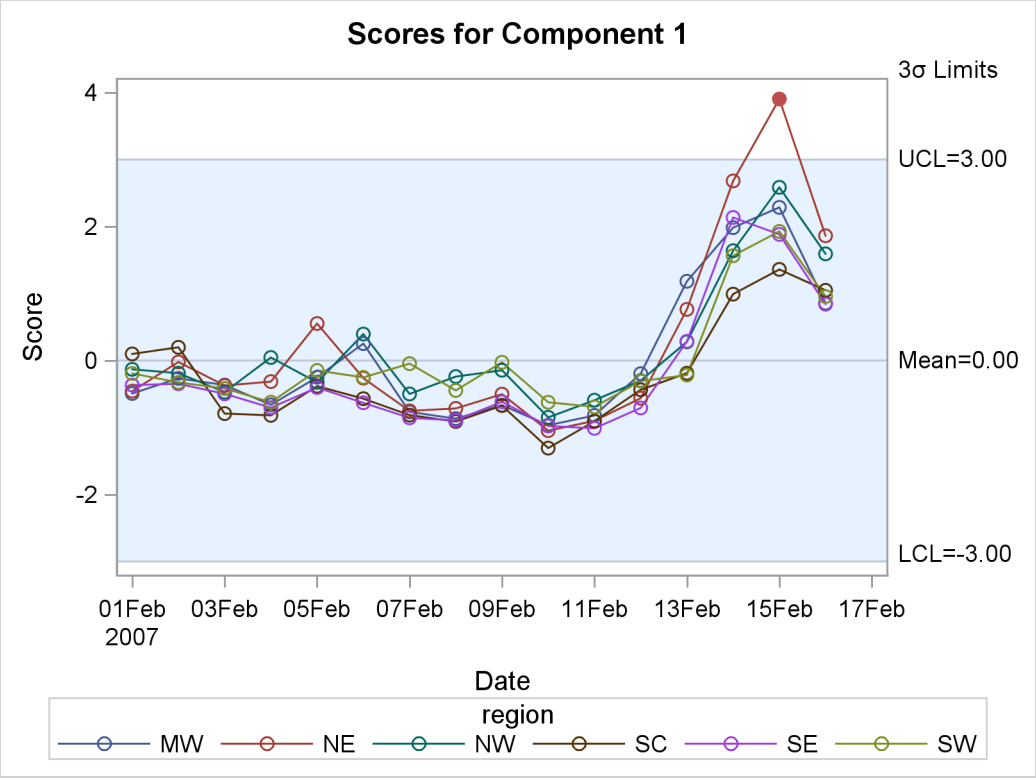


You can also overlay scores for different SERIES values in a single chart. The following statements produce a score chart for principal component 1 for each region:

```
proc mvpmonitor history=mvpair loadings=mvpairloadings;
   time flightDate;
   series region;
   scorechart / overlay=series;
run;
```

The resulting chart is shown in Output 14.1.6.

Output 14.1.6 Score Charts for Principal Component 1 and All Regions



Example 14.2: Creating Multivariate Control Charts for Phase II

This example is a continuation of the example in "Getting Started: MVPMONITOR Procedure" on page 958. The following statements create a data set named flightDelays2, which provides flight delays for the date range February 17–28, 2007, for the northeastern United States:

```
data flightDelays2;
   label flightDate='Date';
   format flightDate MMDDYY8.;
   input flightDate :MMDDYY8. AA CO DL F9 FL NW UA US WN;
   datalines;
02/17/07 25.6  7.8 15.5 13.4 16.1 16.2 23.0 24.2  8.2
02/18/07  5.4 16.0  9.9  1.1 11.5 17.0 15.6 15.5  5.1
02/19/07 13.2 16.3 10.0 10.6  5.4 10.3  9.5 16.8  9.3
```

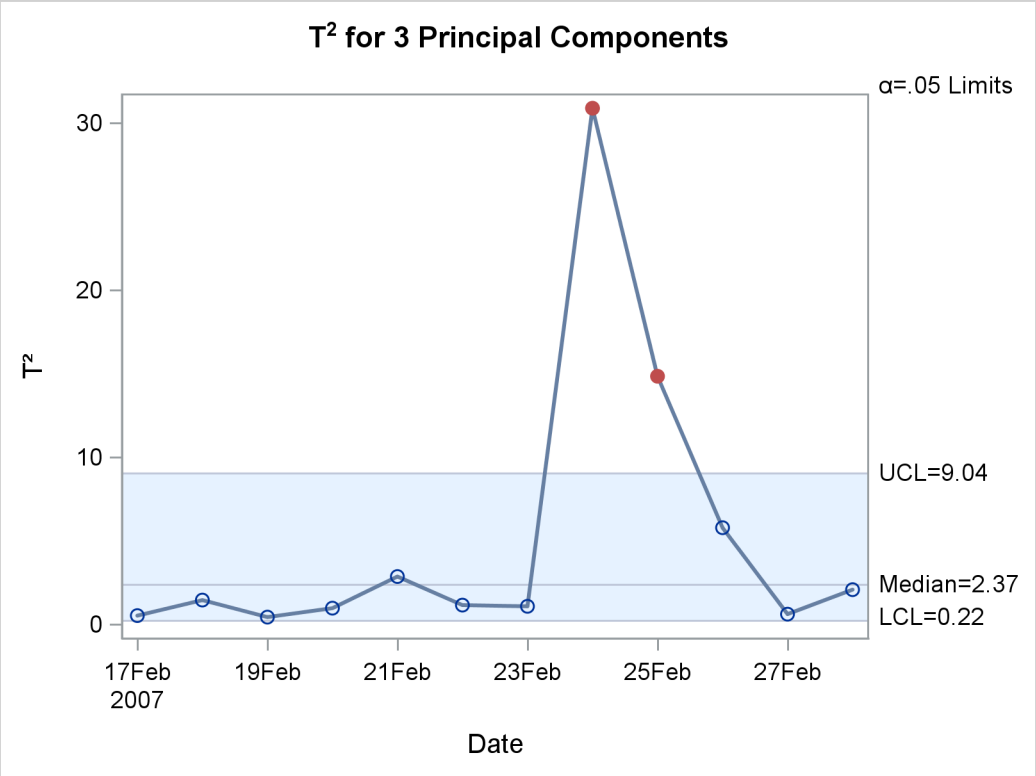
```
02/20/07  4.2  6.9  1.4  0.1  7.2  6.6  7.4 10.4  2.9
02/21/07  5.4 -0.1  7.4  8.7 16.3 24.3  9.4  6.0 10.2
02/22/07 19.6 30.2  6.8  2.7  8.9 16.4 14.3 12.6  8.2
02/23/07 14.9 18.9  9.9  9.1 12.0 16.5 17.4 12.8  6.0
02/24/07 21.4  5.5 11.1 46.1 10.6 55.3 22.9  8.8  3.4
02/25/07 42.6  7.7 14.6 14.4 32.0 50.7 46.1 49.4 39.1
02/26/07 43.2 25.1 18.1 18.2 28.8 31.1 38.6 29.6 18.6
02/27/07 11.3 17.1  5.3  4.1  4.8 13.9  9.8  9.7  7.1
02/28/07  8.1  3.7  2.7 17.1 -0.8  5.5 11.0 14.3  3.1
;
```

To use PROC MVPMONITOR in a Phase II analysis, you need a principal component model based on a process data from a stable process. The model that is produced by the MVPMODEL procedure in "Example 14.1: Combining Data from Peer Processes" on page 981 is used here. The model was generated from data for the continental United States during February 1–16, 2007. The model information is contained in the principal component loadings, which come from the mvpairloadings data set. The following statements apply the model to the new data in flightDelays2:

```
proc mvpmonitor data=flightDelays2 loadings=mvpairloadings;
   time flightDate;
   id flightDate;
   tsquarechart;
   specchart;
run;
```

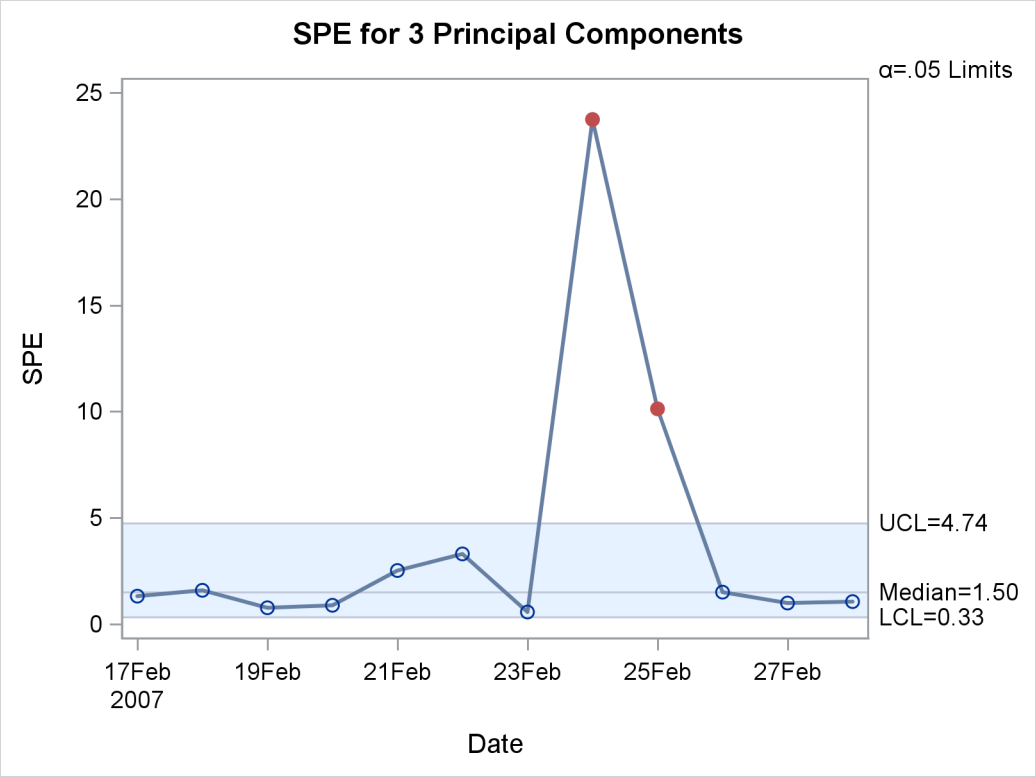
The T^2 chart is shown in Output 14.2.1.

Output 14.2.1 Multivariate Control Chart for T^2 Statistics



The SPE chart is shown in Output 14.2.2.

Output 14.2.2 Multivariate Control Chart for SPE Statistics



The SPE chart has out-of-control points on February 22, 24, and 25. This indicates that the variation is not along the model hyperplane, which suggests that the model is not appropriate for these new data.

Both the SPE chart and the T^2 chart have an out-of-control point on February 25. This point has very unusual variation. On that date, a major winter storm with high winds and blizzard conditions battered the Midwest while snow, sleet, and freezing rain hit the Northeast. These conditions contributed to delays that are not seen in the data set that the model was built with.

Example 14.3: Comparison of Univariate and Multivariate Control Charts

This example shows the effect of a change in correlation of the process variables on the SPE chart. The following statements create a data set called mvpStable, which consists of 30 samples from a trivariate normal distribution with strong positive correlation between all three variables:

```
proc iml;
   Mean = {0,0,0};
   Cor = {1.0 0.8 0.8,
          0.8 1.0 0.8,
          0.8 0.8 1.0};
   StdDevs = {2 2 2};
   D = diag(StdDevs);
   Cov = D*Cor*D; /* covariance matrix */
```

```
   NumSamples = 30;
   call randseed(123321);  /* set seed for the RandNormal module */
   X = RandNormal(NumSamples, Mean, Cov);
   varnames = { x1 x2 x3};
   create mvpStable from X [colname = varnames];
   append from X;
   quit;
run;
data mvpStable;
   set mvpStable;
   hour=_n_;
run;
```

The next statements create a data set called mvpOOC, which has five observations in which the correlations are negative:

```
proc iml;
   Mean = {0,0,0};
   Cor = { 1.0 -0.8  0.8,
          -0.8  1.0 -0.8,
           0.8 -0.8  1.0};
   StdDevs = {2 2 2};
   D = diag(StdDevs);
   Cov = D*Cor*D; /* covariance matrix */
   NumSamples = 5;
   call randseed(123321);  /* set seed for the RandNormal module */
   X = RandNormal(NumSamples, Mean, Cov);
   varnames = { x1 x2 x3};
   create mvpOOC from X [colname = varnames];
   append from X;
   quit;
run;
data mvpOOC;
   set mvpStable mvpOOC;
   hour=_n_;
run;
```

The following statements produce a principal component model for the data in mvpStable:

```
proc mvpmodel data=mvpStable ncomp=1 plots=none out=scores
              outloadings=loadings;
   var x1 x2 x3;
run;
```

The model hyperplane that is defined by specifying NCOMP= 1 is a line. The loadings in the principal component model, which are used to project the data to the model hyperplane, are defined by the correlation structure present in the DATA= data set.

The model explains about 90% of the variance in the data, as shown in Output 14.3.1.

Output 14.3.1 Eigenvalue and Variance Information

The MVPMODEL Procedure

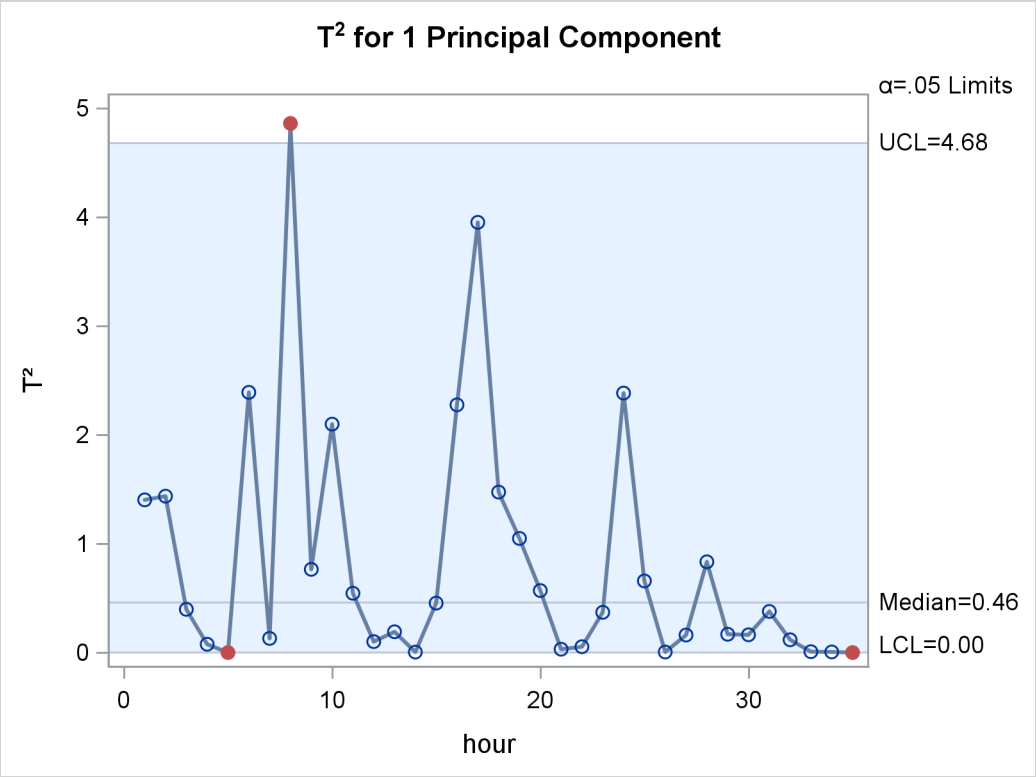
Eigenvalues of the Correlation Matrix				
	Eigenvalue	Difference	Proportion	Cumulative
1	2.67725690		0.8924	0.8924

The loadings from the model are then applied to the data in mvpOOC, which includes observations that have a different correlation structure, which vary in direction orthogonal to the model line. The following statements apply the loadings to these new data to produce T^2 and SPE charts:

```
proc mvpmonitor data=mvpOOC loadings=loadings;
   time hour;
   tsquarechart;
   specchart;
run;
```

The MVPMONITOR procedure generates a T^2 chart, shown in Output 14.3.2.

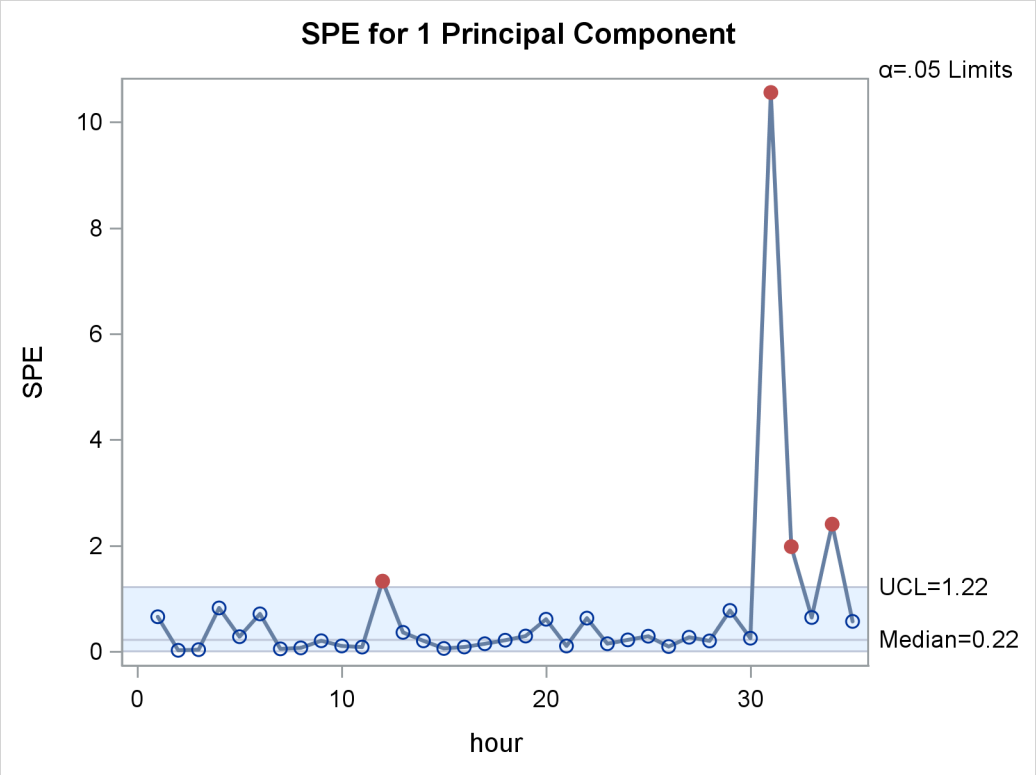
Output 14.3.2 T^2 Chart



The projection of the last five points to the model line results in small amounts of variation, and thus small T^2 statistics, for two reasons: the last five points are orthogonal to the model line, and they share the same mean. However, the orthogonality means that they are out-of-control points in the SPE chart.

The MVPMONITOR procedure also produces an SPE chart, shown in Output 14.3.3.

Output 14.3.3 SPE Chart

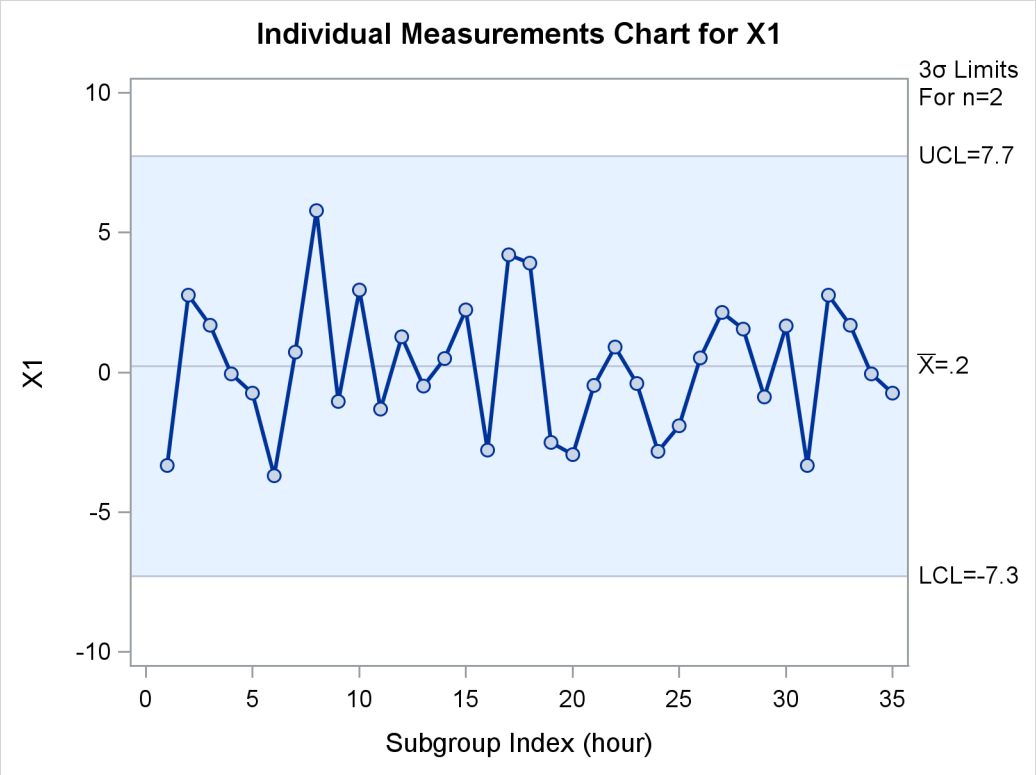


Because the last five points come from a correlation structure that is not seen in the data from which the model was built, these points can lie far from the model line, resulting in large values in the SPE statistics.

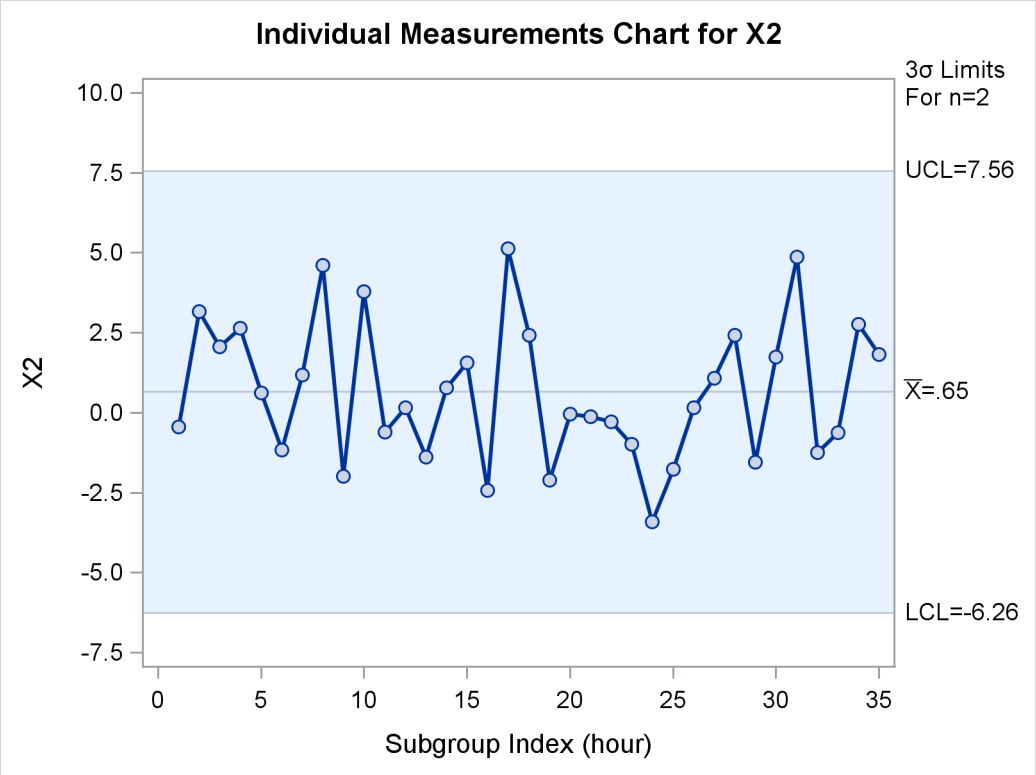
Because the marginal distributions are the same in both the original 30 points and the additional five points, the univariate control charts in Output 14.3.4, Output 14.3.5, and Output 14.3.6 fail to signal the multivariate change at hour 31. The following statements produce univariate control charts for each of the variables by using the SHEWHART procedure:

```
proc shewhart data=mvpOOC;
   irchart (x1 x2 x3) * hour / markers nochart2;
run;
```

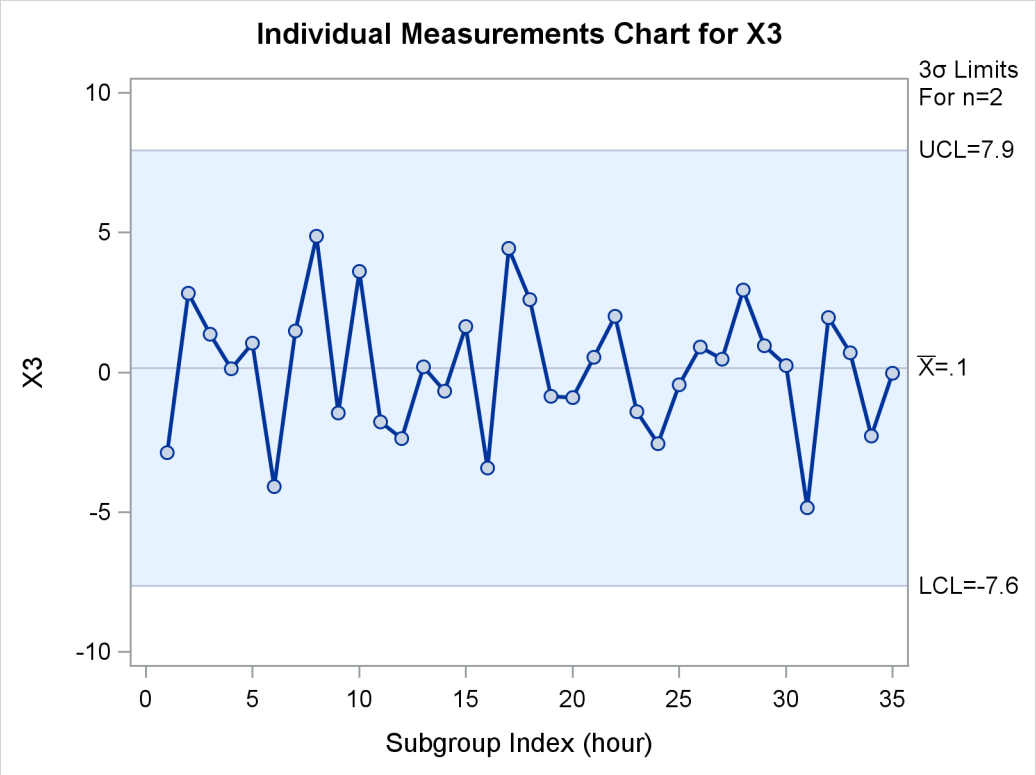
Output 14.3.4 Univariate Chart for x_1



Output 14.3.5 Univariate Chart for x_2



Output 14.3.6 Univariate Chart for x_3



Example 14.4: Creating a Classical T^2 Chart

The following statements use PROC MVPMODEL to create a model from which classical T^2 charts can be produced:

```
proc mvpmodel data=flightDelays ncomp=all noprint out=mvpout;
   var AA CO DL F9 FL NW UA US WN;
run;
```

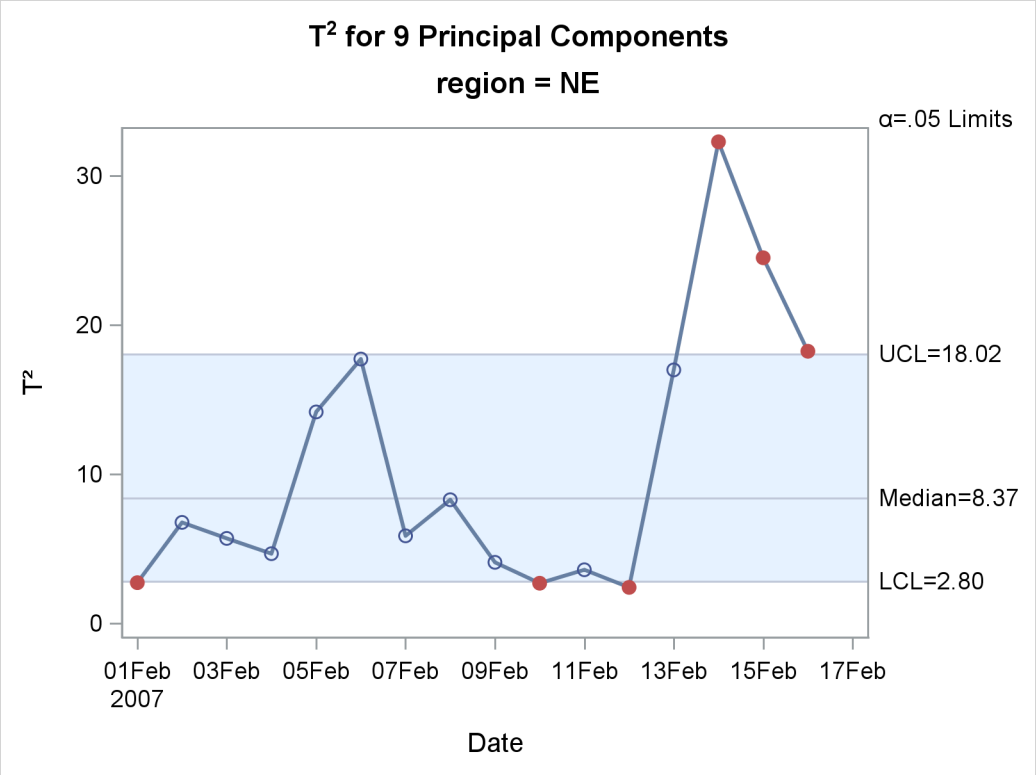
The NCOMP=ALL option specifies that the number of principal components equal the number of process variables, so the mvpout data set contains the classical T^2 statistic for each observation. The mvpout data set contains six observations per time point—one for each region.

The following statements create the classical T^2 chart:

```
proc mvpmonitor history=mvpout;
   time flightDate;
   series region;
   tsquarechart / seriesvalue='NE';
run;
```

The SERIES statement specifies region as the variable that identifies sequences of T^2 statistics, and the SERIESVALUE= option selects the region to be plotted, the Northeast. The classical T^2 chart is shown in Output 14.4.1.

Output 14.4.1 Classical T^2 Chart



In this case, the classical T^2 chart finds out-of-control observations above the upper control limit during February 14–16 and below the lower control limit on February 1, 10, and 12.

Output 14.4.2 shows a partial listing of the mvpout data set. It contains T^2 statistics based on the model that has nine principal components, in addition to the original variables and other observationwise statistics.

Output 14.4.2 Partial Listing of Output Data Set mvpout

flightDate	region	AA	CO	DL	F9	FL	NW	UA	US	WN	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6
02/01/07	MW	14.9	7.1	7.9	8.5	14.8	4.5	5.1	13.4	5.1	-1.16440	0.89425	0.00118	-0.84496	0.07443	-0.05231
02/01/07	NE	15.7	7.1	8.6	6.3	14.6	6.2	7.0	11.0	6.4	-1.08197	0.55936	0.05859	-0.84487	-0.02092	-0.16851
02/01/07	NW	17.8	2.6	6.1	28.8	11.6	6.1	11.6	27.3	3.7	-0.31223	2.23593	-0.03824	0.58893	0.21628	0.67149
02/01/07	SC	19.9	8.3	13.9	4.9	25.8	15.3	9.0	15.1	12.8	0.22974	-0.16976	0.56118	-1.30139	0.22990	-0.34406
02/01/07	SE	16.1	1.9	8.7	8.7	15.1	18.3	4.0	10.4	6.5	-0.87056	0.01044	1.06975	-0.50179	0.38041	-0.33789

Prin7	Prin8	Prin9	_NOBS_	_TSQUARE_	R_AA	R_CO	R_DL	R_F9	R_FL	R_NW	R_UA	R_US	R_WN	_SPE_
-0.27403	0.31880	-0.06726	96	3.2122	0	0	0	0	0	0	0	0	0	.
-0.30596	0.25090	0.12804	96	2.7349	0	0	0	0	0	0	0	0	0	.
0.85708	1.07348	-0.49530	96	18.5555	0	0	0	0	0	0	0	0	0	.
0.11471	0.30201	0.46481	96	6.3335	0	0	0	0	0	0	0	0	0	.
0.63114	0.45028	0.21437	96	6.1537	0	0	0	0	0	0	0	0	0	.

Notice that no SPE statistics are produced when the number of principal components equals the number of process variables.

References

Alt, F. (1985). "Multivariate Quality Control." In *Encyclopedia of Statistical Sciences,* vol. 6, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons.

Cooley, W. W., and Lohnes, P. R. (1971). *Multivariate Data Analysis*. New York: John Wiley & Sons.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. New York: John Wiley & Sons.

Hotelling, H. (1933). "Analysis of a Complex of Statistical Variables into Principal Components." *Journal of Educational Psychology* 24:417–441, 498–520.

Jackson, J. E., and Mudholkar, G. S. (1979). "Control Procedures for Residuals Associated with Principal Component Analysis." *Technometrics* 21:341–349.

Jensen, D. R., and Solomon, H. (1972). "A Gaussian Approximation to the Distribution of a Definite Quadratic Form." *Journal of the American Statistical Association* 67:898–902.

Kourti, T., and MacGregor, J. F. (1995). "Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods." *Chemometrics and Intelligent Laboratory Systems* 28:3–21.

Kourti, T., and MacGregor, J. F. (1996). "Multivariate SPC Methods for Process and Product Monitoring." *Journal of Quality Technology* 28:409–428.

Kshirsagar, A. M. (1972). *Multivariate Analysis*. New York: Marcel Dekker.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. London: Academic Press.

Miller, P., Swanson, R. E., and Heckler, C. H. E. (1998). "Contribution Plots: A Missing Link in Multivariate Quality Control." *Applied Mathematics and Computer Science* 8:775–792.

Morrison, D. F. (1976). *Multivariate Statistical Methods*. 2nd ed. New York: McGraw-Hill.

Nomikos, P., and MacGregor, J. F. (1995). "Multivariate SPC Charts for Monitoring Batch Processes." *Technometrics* 37:41–59.

Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space." *Philosophical Magazine* 6:559–572.

Rao, C. R. (1964). "The Use and Interpretation of Principal Component Analysis in Applied Research." *Sankhyā, Series A* 26:329–358.

Tracy, N. D., Young, J. C., and Mason, R. L. (1992). "Multivariate Control Charts for Individual Observations." *Journal of Quality Technology* 24:88–95.

Wilks, S. S. (1962). *Mathematical Statistics*. New York: John Wiley & Sons.

Chapter 15
The OPTEX Procedure

Contents

Overview: OPTEX Procedure **999**
Features 999
Learning about the OPTEX Procedure 1000
Getting Started: OPTEX Procedure **1001**
Constructing a Nonstandard Design 1001
Creating the Candidate Set 1001
Generating the Design 1002
Customizing the Number of Runs 1004
Including Specific Runs 1004
Using an Alternative Search Technique 1006
Optimal Design Scenarios 1006
Constructing a Saturated Second-Order Design 1007
Augmenting a Resolution 4 Design 1007
Handling Many Variables 1008
Constructing an Incomplete Block Design 1008
Constructing a Mixture-Process Design 1009
Syntax: OPTEX Procedure **1010**
Statement Ordering for Covariate Designs 1010
Summary of Functions 1010
PROC OPTEX Statement 1012
BLOCKS Statement 1014
CLASS Statement 1015
EXAMINE Statement 1019
GENERATE Statement 1020
ID Statement 1024
MODEL Statement 1024
OUTPUT Statement 1025
Details: OPTEX Procedure **1026**
Input Data Sets 1026
DATA= Data Set 1027
AUGMENT= Data Set 1027
INITDESIGN= Data Set 1027
BLOCKS DESIGN= Data Set 1028
BLOCKS COVAR= Data Set 1028
Output Data Sets 1028
Specifying Effects in MODEL Statements 1029

Types of Effects . . . 1029
Bar and @ Operators . . . 1030
Examples of Models . . . 1031
Design Efficiency Measures . . . 1031
Design Coding . . . 1032
Static Coding . . . 1032
Orthogonal Coding . . . 1033
Example of Coding . . . 1033
General Recommendations . . . 1033
Optimality Criteria . . . 1034
Types of Criteria . . . 1034
D-Optimality . . . 1034
A-Optimality . . . 1035
G- and I-Optimality . . . 1035
Distance-Based Criteria . . . 1036
Memory and Run-Time Considerations . . . 1036
Search Methods . . . 1037
Useful Matrix Formulas . . . 1037
Sequential Search Algorithm . . . 1038
Exchange Algorithm . . . 1038
DETMAX Algorithm . . . 1039
Fedorov and Modified Fedorov Algorithms . . . 1039
Optimal Blocking . . . 1039
Search Strategies . . . 1040
General Recommendations . . . 1040
Set of Candidate Points . . . 1040
Initial Design . . . 1040
Output . . . 1041
ODS Tables . . . 1042
Examples: OPTEX Procedure . . . 1042
Example 15.1: Nonstandard Linear Model . . . 1042
Example 15.2: Comparing the Fedorov Algorithm to the Sequential Algorithm . . . 1045
Example 15.3: Using an Initial Design to Search an Optimal Design . . . 1047
Example 15.4: Optimal Design Using an Augmented Best Design . . . 1049
Example 15.5: Optimal Design Using a Small Candidate Set . . . 1050
Example 15.6: Bayesian Optimal Design . . . 1052
Example 15.7: Balanced Incomplete Block Design . . . 1054
Example 15.8: Optimal Design with Fixed Covariates . . . 1057
Example 15.9: Optimal Design in the Presence of Covariance . . . 1060
Example 15.10: Adding Space-Filling Points to a Design . . . 1062
References . . . 1065

Overview: OPTEX Procedure

The OPTEX procedure searches for optimal experimental designs. You specify a set of candidate design points and a linear model, and the procedure chooses points so that the terms in the model can be estimated as efficiently as possible.

Most experimental situations call for standard designs, such as fractional factorials, orthogonal arrays, central composite designs, or Box-Behnken designs. Standard designs have assured degrees of precision and orthogonality that are important for the exploratory nature of experimentation. However, standard designs are not available in some situations, such as the following:

- Not all combinations of the factor levels are feasible.
- The region of experimentation is irregularly shaped.
- Resource limitations restrict the number of experiments that can be performed.
- There is a nonstandard linear or a nonlinear model.

The OPTEX procedure can generate an efficient experimental design for any of these situations.

NOTE: Instead of using PROC OPTEX directly, a more appropriate tool for you might be the ADX Interface. The ADX Interface is designed primarily for engineers and researchers who require a point-and-click solution for the entire experimental process, from building the designs through determining significant effects to optimization and reporting. In addition to offering the standard designs, ADX makes it easy to use PROC OPTEX to find optimal designs for nonstandard factorial, response surface, and mixture experiments, with and without blocking. For more information about the ADX Interface, see *Getting Started with the SAS ADX Interface for Design of Experiments*.

Features

This section summarizes key features of the OPTEX procedure.

The OPTEX procedure offers various criteria for searching a design; these criteria are summarized in Table 15.1 and Table 15.2. In the formulas for these criteria, $\mathbf{X}$ denotes the design matrix, $\mathcal{C}$ the set of candidate points, and $\mathcal{D}$ the set of design points. The default criterion is D-optimality. You can also use the OPTEX procedure to generate G- and I-efficient designs.

The OPTEX procedure also offers a variety of search algorithms, ranging from a simple sequential search (Dykstra 1971) to the computer-intensive Fedorov algorithm (Fedorov 1972; Cook and Nachtsheim 1980). You can customize many aspects of the search, such as the initialization method and the number of iterations.

You can use the full general linear modeling facilities of the GLM procedure to specify a model for your design, allowing for general polynomial effects in addition to classification or ANOVA effects. Optionally, you can specify the following:

- design points to be optimally augmented
- fixed covariates (for example, blocks) for the design
- prior precisions for Bayesian optimal design

The OPTEX procedure is an interactive procedure. After specifying an initial design, you can submit additional statements without reinvoking the OPTEX procedure. Once you have found a design, you can do the following:

- examine the design
- output the design to a data set
- change the model and find another design
- change the characteristics of the search and find another design

Table 15.1 Information-Based Optimality Criteria

Criterion	Goal	Formula
D-optimality	Maximize determinant of the information matrix	$\max \lvert\mathbf{X}'\mathbf{X}\rvert$
A-optimality	Minimize sum of the variances of estimated coefficients	$\min \ \mathrm{trace}(\mathbf{X}'\mathbf{X})^{-1}$

Table 15.2 Distance-Based Optimality Criteria

Criterion	Goal	Formula
U-optimality	Minimize distance from design to candidates	$\min \sum_{\mathbf{x}\in\mathcal{C}} d(\mathbf{x}, \mathcal{D})$
S-optimality	Maximize distance between design points	$\min \sum_{\mathbf{y}\in\mathcal{D}} d(\mathbf{y}, \mathcal{D}-\mathbf{y})$

Learning about the OPTEX Procedure

To learn the basic syntax of the OPTEX procedure, read the introductory example in the next section, which covers a typical application of optimal designs. Other applications are illustrated in the section "Optimal Design Scenarios" on page 1006. The summary tables in the section "Summary of Functions" on page 1010 provides an overview of the syntax. The section "Examples: OPTEX Procedure" on page 1042 illustrates construction of complex designs.

Getting Started: OPTEX Procedure

The examples in this section illustrate basic features of the OPTEX procedure. In addition, the examples show how a variety of SAS software tools can be used to construct candidate sets. If you are working through these examples on your own computer, note that the randomness in the OPTEX procedure's search algorithm will cause your results to be slightly different from those shown.

For illustrations of complex features, see the section "Examples: OPTEX Procedure" on page 1042.

Constructing a Nonstandard Design

NOTE: See *Constructing a Nonstandard Design* in the SAS/QC Sample Library.

This example shows how you can use the OPTEX procedure to construct a design for a complicated experiment for which no standard design is available.

A chemical company is designing a new reaction process. The engineers have isolated the following five factors that might affect the total yield:

Variable	Description	Range
RTemp	Temperature of the reaction chamber	150–350 degrees
Press	Pressure of the reaction chamber	10–30 psi
Time	Amount of time for the reaction	3–5 minutes
Solvent	Amount of solvent used	20–25%
Source	Source of raw materials	1, 2, 3, 4, 5

Although there are only two solvent levels of interest, the reaction control factors (RTemp, Press, and Time) might be curvilinearly related to the total yield, and thus require three levels in the experiment. The Source factor is categorical with five levels. In addition, some combinations of the factors are known to be problematic; simultaneously setting all three reaction control factors to their lowest feasible levels will result in worthless sludge, whereas setting them all to their highest levels can damage the reactor. Standard experimental designs do not apply to this situation.

Creating the Candidate Set

You can use the OPTEX procedure to generate a design for this experiment. The first step in generating an optimal design is to prepare a data set that contains the candidate runs (that is, the feasible factor level combinations). In many cases, this step involves the most work. You can use a variety of SAS data manipulation tools to set up the candidate data set. In this example, the candidate runs are all possible combinations of the factor levels except those in which all three control factors are at their low levels and those in which all three are at their high levels. The PLAN procedure (see *SAS/STAT User's Guide*) provides an easy way to create a full factorial data set, which can then be subsetted by using the DATA step, as shown in the following statements:

```
proc plan ordered;
   factors RTemp=3 Press=3 Time=3 Solvent=2 Source=5 / noprint;
   output out=Candidate
      RTemp   nvals=(150 to 350 by 100)
      Press   nvals=( 10 to  30 by  10)
      Time    nvals=(  3 to   5       )
      Solvent nvals=( 20 to  25 by   5)
      Source  nvals=(  1 to   5       );
data Candidate; set Candidate;
   if (^((RTemp = 150) & (Press = 10) & (Time = 3)));
   if (^((RTemp = 350) & (Press = 30) & (Time = 5)));
run;
proc print data=Candidate(obs=10);
run;
```

A partial listing of the candidate data set Candidate is shown in Figure 15.1.

Figure 15.1 Candidate Set of Runs for Chemical Reaction Design

Obs	RTemp	Press	Time	Solvent	Source
1	150	10	4	20	1
2	150	10	4	20	2
3	150	10	4	20	3
4	150	10	4	20	4
5	150	10	4	20	5
6	150	10	4	25	1
7	150	10	4	25	2
8	150	10	4	25	3
9	150	10	4	25	4
10	150	10	4	25	5

Generating the Design

The next step is to invoke the OPTEX procedure, specifying the candidate data set as the input data set. You must also provide a model for the experiment by using the MODEL statement, which uses the linear modeling syntax of the GLM procedure (see *SAS/STAT User's Guide*). Because Source is a classification (qualitative) factor, you need to specify it in a CLASS statement. To detect possible crossproduct effects in the other factors, in addition to the quadratic effects of the three reaction control factors, you can use a modified response surface model, as shown in the following statements:

```
proc optex data=Candidate seed=12345;
   class Source;
   model Source Solvent|RTemp|Press|Time@2
         RTemp*RTemp Press*Press Time*Time;
run;
```

Note that the MODEL statement does not involve a response variable (unlike the MODEL statement in the GLM procedure). The default number of runs for a design is assumed by the OPTEX procedure to be 10 plus the number of parameters (a total of $10 + 18 = 28$ in this case). Thus, the procedure searches for 28 runs among the candidates in Candidate that enable D-optimal estimation of the effects in the model. (For a precise definition of D-optimality, see the section "Optimality Criteria" on page 1034.) Randomness is built into the search algorithm to overcome the problem of local optima. By default, the OPTEX procedure takes 10 random "tries" to find the best design. The output, shown in Figure 15.2, lists efficiency factors for the 10 designs found. These designs are all very close in terms of their D-efficiency.

Figure 15.2 Efficiencies for Chemical Reaction Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	57.0082	32.8139	78.3162	0.8319
2	56.7660	27.3874	75.8168	0.8563
3	56.2145	28.7217	74.9937	0.8594
4	55.8960	28.7509	74.4196	0.8559
5	55.7341	29.9372	74.4554	0.8544
6	55.6224	31.4902	73.6200	0.8626
7	55.5762	28.3016	75.8959	0.8652
8	55.5080	30.3889	78.4385	0.8552
9	55.3366	28.5103	74.7014	0.8614
10	55.2176	26.8133	76.2307	0.8660

The final step is to save the best design in a data set. You can do this interactively by submitting the OUTPUT statement immediately after the preceding statements. Then use the PRINT procedure to list the design. The design is listed in Figure 15.3.

```
   output out=Reactor;
proc print data=Reactor;
run;
```

Figure 15.3 Optimal Design for Chemical Reaction Process Experiment

Obs	Solvent	RTemp	Press	Time	Source
1	20	150	20	4	5
2	20	250	10	5	5
3	20	350	30	3	5
4	25	150	30	5	5
5	25	250	10	3	5
6	25	350	20	5	5
7	20	150	10	5	4
8	20	150	30	3	4
9	20	350	10	3	4
10	20	350	20	5	4
11	25	250	30	4	4
12	20	250	10	3	3
13	20	350	30	4	3
14	25	150	30	3	3
15	25	350	10	5	3
16	25	350	20	3	3
17	20	150	30	5	2
18	20	250	30	3	2
19	20	350	10	5	2
20	25	150	10	4	2
21	25	250	20	5	2
22	25	350	30	4	2
23	20	150	20	3	1
24	20	250	20	4	1
25	20	250	30	5	1
26	25	150	10	5	1
27	25	350	10	4	1
28	25	350	30	3	1

Customizing the Number of Runs

The OPTEX procedure provides options that enable you to customize many aspects of the design optimization process. Suppose the budget for this experiment can accommodate only 25 runs. You can use the N= option in the GENERATE statement to request a design with this number of runs.

```
proc optex data=Candidate seed=12345;
   class source;
   model source Solvent|RTemp|Press|Time@2
         RTemp*RTemp Press*Press Time*Time;
   generate n=25;
run;
```

Including Specific Runs

If there are factor combinations that you want to include in the final design, you can use the OPTEX procedure to *augment* those combinations optimally. For example, suppose you want to force four specific factor combinations to be in the design. If these combinations are saved in a data set, you can force them

into the design by specifying the data set with the AUGMENT= option in the GENERATE statement. This technique is demonstrated in the following statements:

```
data Preset;
   input Solvent RTemp Press Time Source;
   datalines;
20 350 10 5 4
20 150 10 4 3
25 150 30 3 3
25 250 10 5 3
;
proc optex data=Candidate seed=12345;
   class Source;
   model Source Solvent|RTemp|Press|Time@2
         RTemp*RTemp Press*Press Time*Time;
   generate n=25 augment=preset;
   output out=Reactor2;
run;
```

The final design is listed in Figure 15.4.

```
proc print data=Reactor2;
run;
```

Figure 15.4 Augmented Design for Chemical Reaction Process Experiment

Obs	Solvent	RTemp	Press	Time	Source
1	20	150	30	3	5
2	20	350	20	5	5
3	25	150	10	4	5
4	25	250	30	4	5
5	20	350	10	5	4
6	20	350	30	3	4
7	25	150	30	5	4
8	25	250	10	3	4
9	25	350	20	5	4
10	20	150	10	4	3
11	20	150	30	5	3
12	20	350	20	3	3
13	25	150	30	3	3
14	25	250	10	5	3
15	20	150	10	5	2
16	20	250	30	5	2
17	20	350	10	4	2
18	25	150	20	3	2
19	25	350	10	5	2
20	20	250	10	3	1
21	20	250	20	4	1
22	20	350	30	4	1
23	25	150	10	5	1
24	25	350	10	3	1
25	25	350	30	3	1

Note that the points in the AUGMENT= data set appear as observations 7, 11, 15, and 16.

Using an Alternative Search Technique

You can also specify a variety of optimization methods by using the GENERATE statement. The default method is relatively fast; although other methods might find better designs, they take longer to run and the improvement is usually only marginal. The method that generally finds the best designs is the Fedorov procedure (Fedorov 1972). The following statements show how to request this method:

```
proc optex data=Candidate seed=12345;
   class Source;
   model Source Solvent|RTemp|Press|Time@2
         RTemp*RTemp Press*Press Time*Time;
   generate n=25 method=fedorov;
   output out=Reactor2;
run;
```

The efficiencies for the resulting designs are shown in Figure 15.5.

Figure 15.5 Efficiency Factors for the Fedorov Search

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	56.9072	27.6680	75.2161	0.9023
2	56.8715	27.4939	72.8202	0.9058
3	56.6148	27.7799	75.1840	0.9031
4	56.3021	31.4247	76.0654	0.9044
5	56.0569	25.4498	70.2491	0.9290
6	55.9501	26.8714	75.6991	0.9144
7	55.8461	29.0473	74.1291	0.9138
8	55.8355	26.9242	76.8595	0.9062
9	55.7253	27.4625	74.3391	0.9189
10	55.6071	26.3825	74.1827	0.9107

In this case, the Fedorov procedure takes several times longer than the default method, and D-efficiency shows no improvement. On the other hand, the longer search method often does improve the design and might take only a few seconds on a reasonably fast computer.

Optimal Design Scenarios

The following examples briefly describe some additional common situations that call for optimal designs. These examples show how you can use a variety of SAS software tools to generate an appropriate set of candidate runs and use the OPTEX procedure to search the candidate set for an optimal design.

The emphasis here is on the programming techniques; output is omitted.

Constructing a Saturated Second-Order Design

Suppose you want a design for seven two-level factors that is as small as possible but still permits estimation of all main effects and two-factor interactions—that is, a *saturated* design. Among standard orthogonal arrays, the smallest appropriate 2^k design has 64 runs, far more than the 29 parameters you want to estimate. To generate a D-efficient nonorthogonal design, first use the FACTEX procedure to create the full set of $2^7 = 128$ candidate runs, and then invoke the OPTEX procedure with a full second-order model, asking for a saturated design, as follows:

```
proc factex;
   factors x1-x7;
   output out=Candidate1;
run;
proc optex data=Candidate1 seed=12345;
   model x1|x2|x3|x4|x5|x6|x7@2;
   generate n=saturated;
   output out=Design1a;
run;
```

The default search procedure quickly finds a design with a D-efficiency of 82.3%. If search time is not an issue, you can try a more powerful search technique. For example, you can specify 500 tries with the Fedorov method:

```
proc optex data=Candidate1 seed=12345;
   model x1|x2|x3|x4|x5|x6|x7@2;
   generate n=saturated
            method=fedorov
            iter=500;
   output out=Design1b;
run;
```

This takes much longer to run, and the resulting design is only slightly more D-efficient.

Augmenting a Resolution 4 Design

In a situation similar to the previous example, suppose you have performed an experiment for seven two-level factors with a 16-run, fractional factorial design of resolution 4. You can estimate all main effects with this design, but some two-factor interactions will be confounded with each other. You now want to add enough runs to estimate all two-factor interactions as well. You can use the FACTEX procedure to create the original design in addition to the candidate set.

```
proc factex;
   factors x1-x7;
   output out=Candidate2;
run;
   model resolution=4;
   size design=min;
   output out=Augment2;
run;
```

Now specify Augment2 (the data set that contains the design to be augmented) with the AUGMENT= option in the GENERATE statement:

```
proc optex data=Candidate2 seed=12345;
   model x1|x2|x3|x4|x5|x6|x7@2;
   generate n=30 augment=Augment2;
   output out=Design2;
run;
```

Handling Many Variables

When you have many factors, the set of all possible factor level combinations might be too large to work with as a candidate set. Suppose you want a main-effects design for 15 three-level factors. The complete set of $3^{15} = 14,348,907$ candidates is too large to use with the OPTEX procedure; in fact, it might be too large to store in your computer. One solution is to find a subset of the full factorial set to use as candidates. For example, an alternative candidate set is the 81-run orthogonal design of resolution 3, which can easily be constructed by the FACTEX procedure:

```
proc factex;
   factors x1-x15 / nlev=3;
   model resolution=3;
   size design=81;
   output out=Candidate3;
run;
proc optex data=can3 seed=12345;
   class x1-x15;
   model x1-x15;
   generate n=saturated;
   output out=Design3;
run;
```

Constructing an Incomplete Block Design

An incomplete block design is a design for v (qualitative) treatments in b blocks of size k, where $k < v$ so that not all treatments can occur in each block. To construct an incomplete block design with the OPTEX procedure, simply create a candidate data set that contains a treatment variable with t values and then use the BLOCKS statement. For example, the following statements construct a design for seven treatments in seven blocks of size three:

```
data Candidate4;
   do Treatment = 1 to 7;
      output;
   end;
proc optex data=Candidate4 seed=12345;
   class Treatment;
   model Treatment;
   blocks structure=(7)3;
run;
```

The resulting design is *equireplicated* in the sense that each treatment occurs the same number of times and *balanced* in the sense that each pair of treatments occurs together in the same number of blocks. Balanced designs, when they exist, are known to be optimal, and the OPTEX procedure usually succeeds at finding them for small to moderately sized problems.

Constructing a Mixture-Process Design

Suppose you want to design an experiment with three *mixture factors* X1, X2, and X3 (continuous factors that represent proportions of the components of a mixture) and one *process factor* A (a classification factor with five levels). Furthermore, suppose that X1 can account for no more than 50% of the mixture. The following statements create a data set containing the vertices and generalized edge centroids of the region that is defined by the mixture factor constraints and then use the FACTEX procedure (see the section "Overview: FACTEX Procedure" on page 618) to create a candidate set that includes the process factor:

```
data XVert;
   input x1 x2 x3 @@;
datalines;
0.50 0.000 0.500
0.50 0.500 0.000
0.00 1.000 0.000
0.00 0.000 1.000
0.00 0.500 0.500
0.50 0.250 0.250
0.25 0.000 0.750
0.25 0.750 0.000
0.25 0.375 0.375
;
proc factex;
   factors a / nlev=5;
   output out=Candidate5 pointrep=XVert;
run;
```

Analyzing mixture designs with linear models can be problematic because of the constraint that the mixture factors sum to one; however, to generate an optimal design, you can simply drop one of the mixture factors. The following statements use the preceding candidate set to find an optimal design for fitting the main effect of A and a second-order model in the mixture factors:

```
proc optex data=Candidate5 seed=12345;
   class a;
   model a x1|x2 x1*x1 x2*x2;
run;
```

See Example 15.10 for a more detailed example of a mixture experiment.

Syntax: OPTEX Procedure

The following statements are available in the OPTEX procedure. Items within the brackets <> are optional.

PROC OPTEX < *options* > **;**
CLASS *class-variables* **;**
MODEL *effects* < / *options* > **;**
BLOCKS *block-specification* < *options* > **;**
EXAMINE < *options* > **;**
GENERATE < *options* > **;**
ID *variables* **;**
OUTPUT OUT= *SAS-data-set* < *options* > **;**

To generate a design, you must use the PROC OPTEX and MODEL statements. You can use the other statements as needed. The OPTEX procedure is interactive, so you can use all statements (except the PROC OPTEX statement) after the first RUN statement.

Statement Ordering for Covariate Designs

You use the CLASS and MODEL statements to define a linear model for the runs in the candidate data set. You can also use these statements to define a general covariate model. In this case, list the CLASS and MODEL statements that define the model for the candidate points immediately after the PROC OPTEX statement. Then list the CLASS and MODEL statements that define the covariate model after the BLOCKS DESIGN= specification. Thus, in this case, the ordering for these statements should be as follows:

1. PROC OPTEX statement
2. CLASS and MODEL statements for the candidate points
3. BLOCKS DESIGN= statement
4. CLASS and MODEL statements for the covariates

In addition, a CLASS statement that names classification variables must precede the MODEL statement that uses those variables.

Summary of Functions

Table 15.3, Table 15.4, and Table 15.5 classify the OPTEX statements and options by function.

Table 15.3 Summary of Options for Specifying the Design

Function	Statement	Option
Design Characteristics		
Number of design points	GENERATE	N=*number*
Saturated design	GENERATE	N=SATURATED
Augmented design	GENERATE	AUGMENT=*SAS-data-set*
Bayesian optimal design	MODEL	/ PRIOR=$p_1, p_2, \ldots$
Optimality Criteria		
Minimize trace of $(\mathbf{X}'\mathbf{X})^{-1}$	GENERATE	CRITERION=A
Maximize $\lvert\mathbf{X}'\mathbf{X}\rvert$	GENERATE	CRITERION=D
Minimize mean minimum distance to design	GENERATE	CRITERION=U
Maximize mean distance between nearest design points	GENERATE	CRITERION=S
Model Specification		
Specify independent effects	MODEL	*effects*
Exclude intercept term	MODEL	*effects* NOINT
Specify CLASS variables	CLASS	*variables*
Specify CLASS variable parameterization	CLASS	/ PARAM=*method*
Display CLASS variable parameterization	PROC OPTEX	CLASSPARAM
Static coding	PROC OPTEX	CODING=STATIC
Orthogonal coding	PROC OPTEX	CODING=ORTH
Orthogonal coding with respect to candidates only	PROC OPTEX	CODING=ORTHCAN
Suppress coding of effects	PROC OPTEX	NOCODE
Block Specification		
Specify general covariance matrix for runs	BLOCKS	COVAR=*SAS-data-set <options>* VAR=*variables*
Specify general covariate model	BLOCKS	DESIGN=*SAS-data-set <options>*
Specify *b* blocks of size *k*	BLOCKS	STRUCTURE=(*b*)*k <options>*
Options for block specifications		
Repeat the search *n* times		ITER=*n*
Retain best *m* searches		KEEP=*m*
Select initial design at random		INIT=RANDOM
Select initial design in order		INIT=CHAIN
Initial Design Characteristics		
Random and sequential methods	GENERATE	INITDESIGN=PARTIAL< (*m*) >
Random initial design	GENERATE	INITDESIGN=RANDOM
Sequential initial design	GENERATE	INITDESIGN=SEQUENTIAL
Specify initial design	GENERATE	INITDESIGN=*SAS-data-set*

Table 15.4 Summary of Options for Searching for the Design

Function	Statement	Option
Design Search Specification		
Retain best *n* searches	GENERATE	KEEP=*n*
Search *n* times	GENERATE	ITER=*n*
Specify candidate points	PROC OPTEX	DATA=*SAS-data-set*
Specify random seed	PROC OPTEX	SEED=*number*
Specify effective zero	PROC OPTEX	EPSILON=ϵ
Design Search Methods		
DETMAX algorithm with maximum excursion *level*	GENERATE	METHOD=DETMAX*<(level)>*
Exchange algorithm	GENERATE	METHOD=EXCHANGE
k-exchange algorithm	GENERATE	METHOD=EXCHANGE< (*k*) >
Sequential algorithm	GENERATE	METHOD=SEQUENTIAL
Fedorov algorithm	GENERATE	METHOD=FEDOROV
Modified Fedorov algorithm	GENERATE	METHOD=M_FEDOROV

Table 15.5 Summary of Options for Examining and Saving the Design

Function	Statement	Option
Save the Design		
Best design	OUTPUT OUT=*SAS-data-set*	
Specific design	OUTPUT OUT=*SAS-data-set*	NUMBER=*design-number*
Block variable name	OUTPUT OUT=*SAS-data-set*	BLOCK=*variable-name*
Specify transfer variables	ID	*variables*
List the Design		
Design characteristics	EXAMINE	
Design points	EXAMINE	DESIGN
Information matrix $\mathbf{X}'\mathbf{X}$	EXAMINE	INFORMATION
Specific optimal design	EXAMINE	NUMBER=*design-number*
Variance matrix $(\mathbf{X}'\mathbf{X})^{-1}$	EXAMINE	VARIANCE
Suppress all output	PROC OPTEX	NOPRINT

PROC OPTEX Statement

PROC OPTEX < *options* > **;**

The PROC OPTEX statement invokes the procedure. You can specify the following *options*:

CLASSPARAM
displays a table that summarizes the parameterization of classification variables in the model for the design.

CODING=NONE | STATIC | ORTH | ORTHCAN
specifies which type of coding to use for modeling effects in the design. Coding equalizes all model effects as far as the optimization is concerned. You can specify the following values:

NONE	suppresses coding of effects. This option is equivalent to the NOCODE option.
ORTH	specifies orthogonal coding with respect to the points in the candidate data set and in the AUGMENT= and INITDESIGN= data sets.
ORTHCAN	specifies orthogonal coding with respect to the points in the candidate data set only.
STATIC	requests that the values of all effects be coded to have maximum and minimum values of +1 and –1, respectively.

By default, CODING=STATIC. For more information about coding, see the section "Design Coding" on page 1032. Although CODING=STATIC is the default, CODING=ORTH usually produces give more meaningful efficiency values, especially if all possible combinations of factor levels occur in the candidate data set.

DATA=*SAS-data-set*
specifies the input SAS data set that contains the candidate points for the design. By default, the OPTEX procedure uses the most recently created SAS data set. For more information, see the section "DATA= Data Set" on page 1027.

EPSILON=ϵ
specifies the smallest value ϵ that is considered to be nonzero for determining when the search is no longer yielding an improved design and when the information matrix for the design is singular. By default, $\epsilon = 0.00001$.

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. By default, NAMELEN=20.

NOCODE
suppresses the coding of effects in the model for the design. This option is equivalent to CODING=NONE.

NOPRINT
suppresses all output. This option is useful when you only want the final design to be saved in a data set.

SEED=*s*
specifies an integer used to start the pseudorandom number generator for initialization (see the section "Search Methods" on page 1037). If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated by default from reading the time of day from the computer's clock.

STATUS=*status-level*

requests that the status of the search be checked at the specified *status-level*, which must be an integer between 1 and 4, inclusive. If you specify a *status-level*, then a table of the status at each check point is displayed. You can use this table to track the progress of long searches. The allowable *status-levels* are listed in the following table:

status-level	**Checks status after each:**
1	design search (the number of searches is specified in the NITER= option)
2	search loop
3	internal search loop
4	extra internal search loop for METHOD=M_FEDOROV

Each search method loops to produce successively better designs; these are the search loops for STATUS=2. STATUS=3 and STATUS=4 refer to deeper loops within the search methods. You will need to specify STATUS=3 or STATUS=4 only very rarely, because evaluating and displaying the status at either of these levels usually makes the search much slower.

BLOCKS Statement

BLOCKS *block-specification* < *options* > **;**

You use the BLOCKS statement to find a D-optimal design in the presence of fixed covariates (for example, blocks) or covariance. The technique is an extension of the optimal blocking technique of Cook and Nachtsheim (1989); see the section "Optimal Blocking" on page 1039.

For the purposes of optimal blocking, the model for the original candidate points is referred to as the *treatment model*; the candidate points for the part of the design matrix that corresponds to the treatment model form the *treatment set*. If the GENERATE statement is not specified, then the full candidate set is used as the treatment set; otherwise, an optimal design for the treatment model ignoring the blocks is first generated, and the result is used as the treatment set for optimal blocking.

You can specify any of the following three mutually exclusive *block-specifications*:

COVAR=*SAS-data-set* **VAR=(** *variables* **)**

specifies a data set to use in providing a general covariance matrix for the runs, where *variables* names the variables in this data set that contain the columns of the covariance matrix for the runs. For an example, see Example 15.9.

DESIGN=*SAS-data-set*

specifies a data set to use in providing a general covariate model. In addition to this data set, you must use the CLASS and MODEL statements to specify a covariate model. Covariate models are specified in the same way as the treatment model; CLASS and MODEL statements that come after a BLOCKS statement that involves the DESIGN= specification are interpreted as applying to the covariate model. For an example, see Example 15.8.

STRUCTURE=(*b***)** *k*

specifies a block design that has *b* blocks of size *k*. For an example, see Example 15.7.

You can also specify the following *options*:

INIT=RANDOM | CHAIN
specifies the initialization method for constructing the starting design. You can specify the following values:

CHAIN	selects candidate points in the order in which they occur in the original data set.
RANDOM	constructs the starting design by selecting candidates at random without replacement.

By default, INIT=RANDOM.

ITER=*n*
specifies the number of times to repeat the search from different initial designs. Because local optima are common in difficult search problems, it is often a good idea to make several tries for the optimal design with a random or partially random method of initialization (see the preceding INIT= option). By default, $n = 10$. Specify both INIT=CHAIN and ITER=0 to evaluate the initial design itself.

KEEP=*m*
retains only the best m designs. The value m must be less than or equal to the value n of the ITER= option. By default $m = n$, so that all iterations are kept. This option is useful when you want to make many searches to overcome the problem of local optima but you are only interested in the results of the best m designs.

NOEXCHANGE
suppresses the part of the optimal blocking algorithm that exchanges treatment design points for candidate treatment points. When this option is specified, only interchanges between design points are performed. Use this option when you do not want to change which treatment points are included in the design and you only want to find their optimal ordering.

CLASS Statement

CLASS *variable* < **(***v-options***)** > < *variable* < **(***v-options***)** > . . . > < / *v-options* > > **;**

You use the CLASS statement to identify classification (qualitative) variables, which are factors that separate the observations into groups. For example, a completely randomized design has a single *variable* that identifies the groups of observations. A randomized complete block design has two *variables*; one identifies the blocks and one identifies the treatments.

You can specify various *v-options* for each *variable* by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. However, individual CLASS variable *v-options* override the global *v-options*.

The OPTEX procedure uses the formatted values of *variables* (can be either numeric or character) in forming model effects. Any variable in the model that is not listed in the CLASS statement is assumed to be continuous (quantitative). Continuous variables must be numeric.

NOTE: If you use the DESIGN= option in the BLOCKS statement to specify a data set that contains fixed covariate effects, then a CLASS or MODEL statement that follows the BLOCKS statement refers to the model for the fixed covariates. A CLASS or MODEL statement that defines the model for the candidate points (treatment model) should be specified *before* the BLOCKS statement.

DESCENDING

DESC

reverses the sorting order of the classification variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use the CONTRAST statement. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC OPTEX run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. This represents a change from how class levels were ordered before SAS 8, when numeric class levels with no explicit format were ordered by their BEST12. formatted values. In order to revert to the previous ordering, you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically. The following table shows how PROC OPTEX interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value (the sort order is machine-dependent)
FREQ	Descending frequency count; levels with the most observations come first in the order
INTERNAL	Unformatted value (the sort order is machine-dependent)

By default, ORDER=FORMATTED.

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*method*

specifies the parameterization method for the classification variables. Design matrix columns are created from CLASS variables according to the specified coding scheme.

By default, PARAM=ORTHEFFECT. This represents a change from how classification variables were parameterized before SAS 9, when the default was PARAM=EFFECT. In order to revert to the previous parameterization, you can specify PARAM=EFFECT explicitly for the affected classification variables. The change was implemented because an orthogonal parameterization leads to D- and A-efficiency values that more realistically reflect the true efficiency of the design.

You can specify the following parameterization *methods*, all of which are full rank. The orthogonal versions perform a scaled, intercept-augmented Gram-Schmidt orthogonalization on the columns of the corresponding nonorthogonal parameterizations. Each description shows how a model that has one CLASS variable A with four levels (1, 2, 5, and 7) is coded.

EFFECT specifies effect coding. Three columns are created to indicate group membership of the nonreference levels. The REF= option in the CLASS statement determines the reference level. For the reference level, all three dummy variables have a value of –1. For example, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding			
A	**Design Matrix**		
1	1	0	0
2	0	1	0
5	0	0	1
7	–1	–1	–1

Parameter estimates of CLASS main effects that use the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

POLYNOMIAL | POLY specifies polynomial coding. Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored and the internal, unformatted values are used. If the CLASS levels are not numeric, they are translated into 1, 2, 3, . . . according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
A	**Design Matrix**		
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE | REF specifies reference cell coding. Three columns are created to indicate group membership of the nonreference levels. The REF= option in the CLASS statement determines the reference level. For the reference level, all three dummy variables have a value of 0. For example, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
A	**Design Matrix**		
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects that use the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORDINAL | ORD specifies ordinal ("thermometer") coding. Three columns are created to indicate group membership in successive collections of levels after the first. For example, the design matrix columns for A are as follows.

Ordinal Coding			
A	**Design Matrix**		
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

Parameter estimates of CLASS main effects that use the ordinal coding scheme estimate the difference in the average effect of each successive collection of levels compared to the effect of the first level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=EFFECT and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Effects Coding			
A	**Design Matrix**		
1	1.414	–0.816	–0.577
2	0	1.633	–0.577
5	0	0	1.732
7	–1.414	–0.816	–0.577

ORTHPOLY specifies orthogonal polynomial coding. The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=POLY and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
A	**Design Matrix**		
1	–1.153	0.907	–0.921
2	–0.734	–0.540	1.473
5	0.524	–1.370	–0.921
7	1.363	1.004	0.368

If the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored and the internal, unformatted values are used.

ORTHREF specifies orthogonal reference cell coding. The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=REFERENCE and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
A	**Design Matrix**		
1	1.732	0	0
2	–0.577	1.633	0
5	–0.577	–0.816	1.414
7	–0.577	–0.816	–1.414

ORTHORDINAL The columns are obtained by applying the Gram-Schmidt orthogonalization to the mean-centered columns for PARAM=ORDINAL, and then scaling so that the sum of squares for each column equals the number of levels. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
A	**Design Matrix**		
1	–1.732	0	0
2	0.577	–1.633	0
5	0.577	0.816	–1.414
7	0.577	0.816	1.414

REF=*'level'* | **FIRST** | **LAST**
specifies the reference level for PARAM=EFFECT or PARAM=REFERENCE. You can specify the following values:

'*level*' specifies the *level* of the variable to use as the reference level. You can specify this value only for an individual *v-option*. You cannot specify this value for a global *v-option*.

FIRST designates the first ordered level as reference.

LAST designates the last ordered level as reference.

By default, REF=LAST.

TRUNCATE
determines class levels by using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to SAS 9.

EXAMINE Statement

EXAMINE < *options* > **;**

You use the EXAMINE statement to display the characteristics of a selected design. By default, the EXAMINE statement lists certain measures of design efficiency for the best design. (See the section "Output" on page 1041.) You can specify the following *options* to modify the output:

DESIGN
lists the actual points in the selected design. Designs are ordered by the value of the efficiency criterion that is being optimized. Thus, a *design-number* of 1 (specified in the NUMBER= option) corresponds

to the best design found, a *design-number* of 2 corresponds to the second best design, and so on. By default, the first design (NUMBER=1) is examined. You can select a different design to be examined by using the NUMBER= option.

INFORMATION
INFO
I
lists the information matrix $\mathbf{X}'\mathbf{X}$ for the selected design.

NUMBER=*design-number*
selects a design to examine by specifying its *design-number*.

VARIANCE
VAR
V
lists the variance matrix $(\mathbf{X}'\mathbf{X})^{-1}$ for the parameter estimates for the selected design.

For more information about design efficiencies, see the section "Design Efficiency Measures" on page 1031.

If you use the OPTEX procedure interactively, you must enter the *options* for every EXAMINE statement. For example, the following statements list default information and the design points for the best design but only default information for the second-best design:

```
examine number=1 design;
examine number=2;
```

The following statements list default information and design points for both the best and second-best designs:

```
examine number=1 design;
examine number=2 design;
```

GENERATE Statement

GENERATE < *options* > **;**

You use the GENERATE statement to customize the search for a design. By default, the OPTEX procedure searches for a design by doing the following:

- using the exchange algorithm (METHOD=EXCHANGE)
- using D-optimality as the optimality criterion (CRITERION=D)
- using a completely random initial design to start the search (INITDESIGN=RANDOM)
- selecting candidate points only from the DATA= data set (modified by using AUGMENT= or INITDESIGN= data sets)
- performing 10 iterations in the search (ITER=10)

- finding a design with $10 + p$ points, where p is the number of parameters in the model (modified by using the N= or INITDESIGN= option)

You can specify the following *options* to modify these defaults:

AUGMENT=*SAS-data-set*
specifies a data set that contains a design to be augmented—in other words, a set of points that must be contained in the generated design. When creating designs, the OPTEX procedure adds points from the DATA= data set (or the last data set created, if the DATA= option is not specified) to points from the AUGMENT= data set. The number of points in the design to be augmented must be less than the number of points specified in the N= option. For more information, see the section "AUGMENT= Data Set" on page 1027.

CRITERION=D | A | U | S
specifies the optimality criterion used in the search. You can specify any one of the following values:

A specifies A-optimality; the optimal design minimizes the sum of the variances of the estimated parameters for the model, which is the same as minimizing the trace of $(\mathbf{X}'\mathbf{X})^{-1}$.

D specifies D-optimality; the optimal design maximizes the determinant $|\mathbf{X}'\mathbf{X}|$ of the information matrix for the design.

S specifies S-optimality; the optimal design maximizes the harmonic mean of the minimum distance from each design point to any other design point. Mathematically, an S-optimal design maximizes

$$\frac{N_D}{\sum_{\mathbf{y}\in\mathcal{D}} 1/d(\mathbf{y}, \mathcal{D}-\mathbf{y})}$$

where $\mathcal{D}$ is the set of design points and N_D is the number of points in $\mathcal{D}$. This measures how spread out the design points are; thus, an S-optimal design is also called a *maximum spread design*.

U specifies U-optimality; the optimal design minimizes the sum of the minimum distances from each candidate point to the design. That is, if $\mathcal{C}$ is the set of candidate points, $\mathcal{D}$ is the set of design points, and $d(\mathbf{x}, \mathcal{D})$ is the minimum distance from $\mathbf{x}$ to any point in $\mathcal{D}$, then a U-optimal design minimizes

$$\sum_{\mathbf{x}\in\mathcal{C}} d(\mathbf{x}, \mathcal{D})$$

This measures how well the design "covers" the candidate set; thus, a U-optimal design is also called a *uniform coverage design*.

By default, CRITERION=D. For more information about the different criteria, see the section "Optimality Criteria" on page 1034.

INITDESIGN=SEQUENTIAL | RANDOM | PARTIAL <*m*> | *SAS-data-set*
specifies a method of obtaining an initial design for the search procedure. You can specify the following values:

SEQUENTIAL specifies an initial design chosen by a sequential search. The design that is produced by this option is the same as the design that is produced by METHOD=SEQUENTIAL. You can use this option with other values of the

METHOD= option to specify a sequential design as the initial design for various search methods. For more information, see the section "Search Methods" on page 1037.

RANDOM specifies a completely random initial design. The initially generated design consists of a random selection of observations from the DATA= data set.

PARTIAL<(*m*)> specifies an initial design by using a mixture of RANDOM and SEQUENTIAL methods. A small number (n_r) of points for the initial design are chosen at random from the candidates, and the rest of the design points are chosen by a sequential search. (For a definition of the sequential search, see the section "Search Methods" on page 1037.)

You can specify the optional integer *m* to modify the selection of n_r. By default, or if $m = 0$, n_r is randomly chosen between 0 and one less than half the number of parameters in the linear model. If $m > 0$, then n_r is randomly chosen between 0 and m for each try. If $m < 0$, then $n_r = |m|$ for each try. The maximum value for $|m|$ is the number of points in the design. For notes on choosing n_r, see Galil and Kiefer (1980).

SAS-data-set specifies a data set that holds the initial design. Use this option when you have a specific design that you want to improve or when you want to evaluate an existing design. For more information, see the section "INITDESIGN= Data Set" on page 1027.

The default initialization method depends on the search procedure as shown in Table 15.6.

Table 15.6 Default Initialization Methods

Search Procedure (METHOD= Option)	Default Initialization Method (INITDESIGN= Option)
DETMAX	PARTIAL
EXCHANGE	RANDOM
FEDOROV	RANDOM
M_FEDOROV	PARTIAL
SEQUENTIAL	None

If you specify INITDESIGN=*SAS-data-set* and METHOD=SEQUENTIAL, no search is performed; the INITDESIGN= data set is taken as the final design. By specifying these options, you can use the procedure to evaluate an existing design.

ITER=*n*

specifies the number (*n*) of searches to make. Because local optima are common in difficult search problems, it is often a good idea to make several tries for the optimal design by using a random or partially random method of initialization (see the preceding INITDESIGN= option).

The *n* designs that are found are sorted by their respective efficiencies according to the current optimality criterion (see the CRITERION= option on page 1021). The most efficient design is assigned a *design-number* of 1, the second most efficient design is assigned a *design-number* of 2, and so on. You can then specify the *design-number* in the NUMBER= option in the EXAMINE and OUTPUT statements to display the characteristics of a design or to save a design in a data set.

By default, ITER=10.

KEEP=*m*

retains only the best *m* designs. The value *m* must be less than or equal to the value *n* of the ITER= option. By default *m* = *n*, so that all iterations are kept. This option is useful when you want to make many searches to overcome the problem of local optima but are interested only in the results of the best *m* designs.

METHOD=DETMAX<(*level***)> | EXCHANGE <(***k***)> | FEDOROV | M_FEDOROV | SEQUENTIAL**

specifies the procedure used to search for the optimal design. You can specify the following values:

DETMAX<(*level*)> uses the DETMAX algorithm of Mitchell (1974a). This algorithm is the best-known and most widely used optimal design search algorithm. The optional *level* specifies the maximum excursion level for the search, where *level* is an integer greater than or equal to 1. The default value for *level* is 4. In general, larger values of *level* result in longer search times.

EXCHANGE<(*k*)> uses the simple exchange method of Mitchell and Miller (1970). The optional *k* specifies the *k*-exchange search method of Johnson and Nachtsheim (1983), which generalizes the modified Fedorov search algorithm of Cook and Nachtsheim (1980).

FEDOROV uses the Fedorov algorithm (Fedorov 1972), which seeks the pair $(\mathbf{x}, \mathbf{y})$ of one candidate point and one design point that maximizes $\Delta(\mathbf{x}, \mathbf{y})$ and then switches $\mathbf{x}$ for $\mathbf{y}$ in the design.

M_FEDOROV uses the modified Fedorov algorithm of Cook and Nachtsheim (1980), which computes the same number of Δ's on each step but switches each point $\mathbf{y}$ in the design with the candidate point $\mathbf{x}$ that maximizes $\Delta(\mathbf{x},\mathbf{y})$. This procedure is generally as reliable as the simple Fedorov algorithm in finding the optimal design, but it can be up to twice as fast.

SEQUENTIAL uses the sequential search of Dykstra (1971), which starts with an empty design and adds successive candidate points so that the chosen criterion is optimized at each step. The is the simplest and fastest algorithm.

By default, METHOD=EXCHANGE. From fastest to slowest, the methods are as follows:

$$\text{SEQUENTIAL} \rightarrow \text{EXCHANGE} \rightarrow \text{DETMAX} \rightarrow \text{M\_FEDOROV} \rightarrow \text{FEDOROV}$$

In general, slower methods result in more efficient designs. Although the default method (METHOD=EXCHANGE) always works relatively quickly, you might want to specify a more reliable method, such as METHOD=M_FEDOROV, when you have a fast computer or a small to moderately sized problem.

For more information about the algorithms, see the section "Search Methods" on page 1037.

N=*n* **| SATURATED**

specifies the number of points in the final design. The default design size is 10 + *p*, where *p* is the number of parameters in the model. If you use the INITDESIGN= option, the default number is the number of points in the initial design. Specify N=*n* to search for a design that has *n* points. Specify N=SATURATED to search for a design whose number of points is equal to the number of parameters in the model. A saturated design has no degrees of freedom to estimate error and should be used with caution.

ID Statement

ID *variables* **;**

You use the ID statement to name the *variables* in the DATA= data set that are not involved in the model but are to be transferred from the input data set to the output data set.

The *variables* must be contained in the DATA= data set, which is specified in the PROC OPTEX statement. They can also be contained in other input data sets. If a *variable* is also contained in an AUGMENT= or INITDESIGN= data set and an observation from that data set is used in the final design, the values of the *variables* for that observation are transferred to the OUT= data set. For more information, see the section "Input Data Sets" on page 1026.

MODEL Statement

MODEL *effects* < / *options* > **;**

You use the MODEL statement to specify the independent effects used to model data that are to be collected with the design that is being constructed. The *effects* can be any of the following:

- simple continuous regressor effects
- polynomial continuous effects
- main effects of classification variables
- interactions of classification variables
- continuous-by-class effects

The variables that are used to form *effects* in the MODEL statement must be present in all input data sets. For more information about input data sets, see the section "Input Data Sets" on page 1026. For more information about the specification of different types of effects and about how the design matrix is defined with respect to the effects, see the section "Specifying Effects in MODEL Statements" on page 1029.

If you use the DESIGN= option in the BLOCKS statement to specify a data set that contains fixed covariate effects, then a CLASS or MODEL statement that *follows* the BLOCKS statement refers to the model for the fixed covariates. A CLASS or MODEL statement that defines the model for the candidate points (treatment model) should occur *before* the BLOCKS statement.

You can specify the following *options*:

NOINT
: excludes the intercept parameter from the model. By default, the OPTEX procedure includes the intercept parameter in the model.

PRIOR=*num-list*
: specifies prior precision values that correspond to groups of effects in the model. Groups of effects in the MODEL statement that have the same prior precision must be separated by commas. Then use the

PRIOR= option, listing as many prior precision values as there are groups of effects. See Example 15.6 for an example.

When you specify prior precision values, the information matrix for estimating the linear parameters is $\mathbf{X}'\mathbf{X} + \mathbf{P}$, where $\mathbf{X}$ is the design matrix and $\mathbf{P}$ is a diagonal matrix whose diagonal contains the prior precision values that you specify. Thus, in terms of a prior distribution, the inverses of the prior precision values can be interpreted as prior variances for the linear parameters that correspond to each effect. As an alternative interpretation, note that with orthogonal coding the value of the prior for an effect says approximately how many prior "observations' worth" of information you have for that effect. For more information about orthogonal coding, see the section "Design Coding" on page 1032.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set* < *options* > **;**

You use the OUTPUT statement to save a design in an output data set. By default, the saved design is the best design found. You specify the data set name as follows:

OUT=*SAS-data-set*
gives a name for the output data set. The OUT= data set is required in the OUTPUT statement.

You can specify the following *options*:

BLOCKNAME=*variable-name*
specifies the name to be given to the blocking variable in the output data set. The default name is BLOCK. You can use this option in conjunction with a STRUCTURE= option in the BLOCKS statement. See Example 15.7 for an example.

NUMBER=*design-number* | **DBEST** | **ABEST** | **GBEST** | **VBEST**
specifies how to select the design to output. You can specify the following values:

design-number	selects a design to output by specifying its *design-number*. Designs are ordered by the value of the efficiency criterion that is being optimized. Thus, a *design-number* of 1 corresponds to the best design found, a *design-number* of 2 corresponds to the second best design, and so on. To modify the number of designs created, see the ITER= option.
DBEST	selects the design that has the highest D-efficiency value.
ABEST	selects the design that has the highest A-efficiency value.
GBEST	selects the design that has the highest G-efficiency value.
VBEST	selects the design that has the minimum average standard error for prediction.

By default, NUMBER=1.

The DBEST, ABEST, GBEST, and VBEST options can be used to find designs that are efficient for more than one criterion. For example, you can use the default CRITERION=D option in the GENERATE statement with the NUMBER=GBEST option in the OUTPUT statement to find the D-optimal design that has maximal G-efficiency. In fact, this is the best way to use the OPTEX procedure to find G-efficient designs; for more information, see the section "G- and I-Optimality" on page 1035.

Details: OPTEX Procedure

Input Data Sets

This section discusses the five input data sets for the OPTEX procedure. Three of the data sets provide points to be used to generate the design according to the effects you specify in the MODEL statement. Two other data sets provide points to be used to generate a model for fixed covariates.

Only the DATA= data set is required. If you do not specify a DATA= data set in the PROC OPTEX statement, the procedure uses the last data set created as a set of candidate points for the design. The AUGMENT= data set is optional and contains points that are guaranteed to be included in the final design. The INITDESIGN= data set is also optional and provides an initial design to be used by a search procedure. Variables listed in the MODEL statement must be present in all three of these data sets, and the variable characteristics (type and length) must match across data sets.

Figure 15.6 is a schematic diagram of the roles of the DATA=, AUGMENT=, and INITDESIGN= data sets in constructing the design. Figure 15.7 presents the role of the DESIGN= data set for block designs.

Figure 15.6 Choosing from DATA= Points

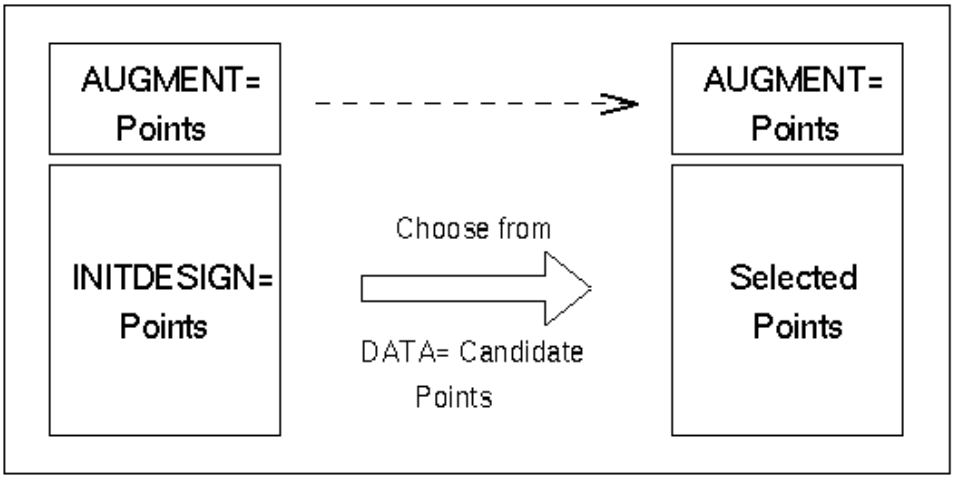
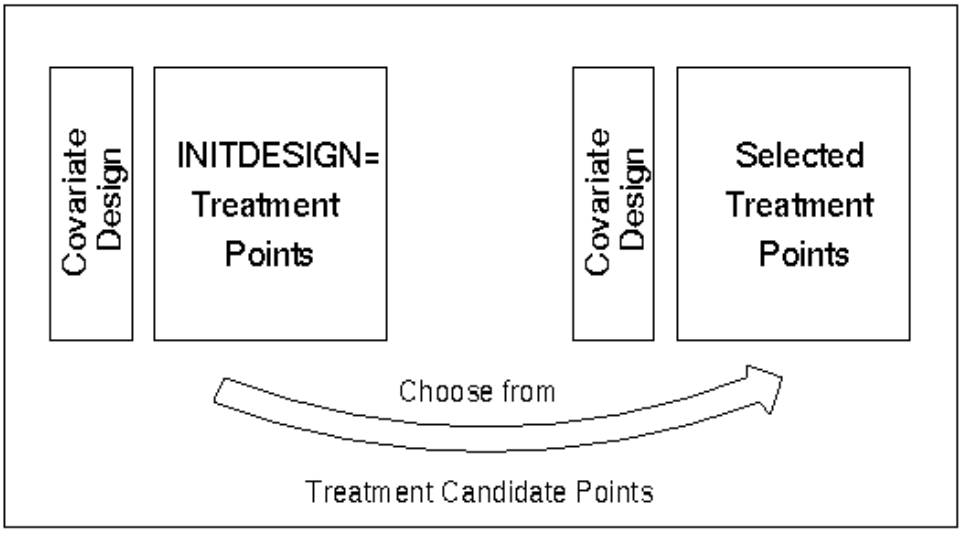


Figure 15.7 Choosing Treatment Candidates



DATA= Data Set

The DATA= data set provides a set of candidate points to be used to create a design. The OPTEX procedure uses the variables listed in the MODEL statement when creating a design.

The effects specified in a MODEL statement determine the variables to be used when generating a design. For example, if the DATA= data set contains the variables A, B, and C, but the MODEL statement specifies effects that involve only A and B, then the variable C is not considered when generating designs.

Variables in the DATA= data set that are listed in the ID statement are transferred to the OUT= data set (if one is created).

AUGMENT= Data Set

The AUGMENT= data set provides a set of points that must be included in the final design. The OPTEX procedure adds candidate points from the DATA= data set to the points from the AUGMENT= data set when generating designs. The number of points in the AUGMENT= data set must be less than or equal to the number of points for the design (either the default or the number specified by the N= option in the GENERATE statement).

As with the DATA= data set, the effects specified in a MODEL statement determine the variables used when generating a design. The types and lengths of variables in an AUGMENT= data set that are used in the MODEL and ID statements must match the types and lengths of the same variables in the DATA= data set. If you use an ID statement and the AUGMENT= data set contains the ID variables, these variables are transferred to the OUT= data set (if one is created). For an example that uses an AUGMENT= data set, see the section "Including Specific Runs" on page 1004.

INITDESIGN= Data Set

The INITDESIGN= data set provides a set of points that are used as an initial design in the search for an optimal design. These points are not necessarily contained in the final design. The OPTEX procedure uses these points to begin the search for an optimal design. The number of points in the INITDESIGN= data set

must be the same as the number of points in the design (either the default or the number specified by the N= option in the GENERATE statement).

As with the DATA= data set, the effects specified in a MODEL statement determine the variables used when generating a design. The types and lengths of variables in an INITDESIGN= data set that are used in the MODEL and ID statements must match the types and lengths of the same variables in the DATA= data set. If you use an ID statement and the INITDESIGN= data set contains the ID variables, these variables are transferred to the OUT= data set (if one is created). See Example 15.3 for an example that uses an INITDESIGN= data set.

If you use an INITDESIGN= data set and also specify METHOD=SEQUENTIAL in the GENERATE statement, no search is performed (you do not have to specify ITER=0 in this case). The INITDESIGN= data set is the final design. In this way, you can use the OPTEX procedure to evaluate an existing design.

BLOCKS DESIGN= Data Set

The DESIGN= data set in the BLOCKS statement contains a set of points that are used to generate a model for fixed covariates. These points are contained in the final design and are transferred to the OUT= data set (if one is created). See Example 15.8 for an example that uses a BLOCKS DESIGN= data set.

BLOCKS COVAR= Data Set

If you specify a COVAR= data set in the BLOCKS statement, the observations for the variables listed in the VAR= option are used to define the assumed variance-covariance matrix for the experimental runs. These observations are *not* transferred to the OUT= data set (if one is created). Because covariance matrices are necessarily square, the number of observations in the COVAR= data set must be the same as the number of variables listed in the VAR= option. See Example 15.9 for an example that uses a BLOCKS COVAR= data set.

Output Data Sets

You typically use the OPTEX procedure to create an output data set that contains the design for your experiment. If you use an OUTPUT statement, the variables in the output data set are the factors of the design in addition to any ID variables. The values for the ID variables are taken from the input data set (the DATA=, AUGMENT=, or INITDESIGN= data set) that provided the design point. ID variables must be contained in the DATA= data set and can also be contained in the AUGMENT= or INITDESIGN= data set. If an AUGMENT= or INITDESIGN= data set does not contain the ID variables and points from the data set are used in the final design, values of ID variables for those points are missing.

Because the input data sets provide candidate points for the design, all the observations in the OUT= data set originate in one of the input data sets. The OPTEX procedure does not change the values of variables in the input data sets.

Because you can use multiple OUTPUT statements with the OPTEX procedure, you can create multiple OUT= data sets in a single run of the procedure.

Specifying Effects in MODEL Statements

This section discusses how to specify the linear model that you plan to fit with the design. The OPTEX procedure provides for the same general linear models as the GLM procedure, although it does not use the GLM procedure's *overparameterized* technique for generating the design matrix (see the section "Static Coding" on page 1032.)

Each term in a model, called an *effect*, is a variable or combination of variables. To specify effects, you use a special notation that involves variables and operators. There are two kinds of variables: *classification variables* and *continuous variables*. *Classification variables* separate observations into groups, and the model depends on them through these groups; on the other hand, the model depends on the actual (or coded) values of *continuous variables*. There are two primary operators: *crossing* and *nesting*. A third operator, the *bar operator*, simplifies the specification for multiple crossed terms, as in a factorial model. The `@` operator, used in combination with the bar operator, further simplifies specification of crossed terms.

When specifying a model, you must list the classification variables in a CLASS statement. Any variables in the model that are not listed in the CLASS statement are assumed to be continuous. Continuous variables must be numeric.

Types of Effects

Five types of effects can be specified in the MODEL statement. Each row of the design matrix is generated by combining values for the independent variables according to effects that are specified in the MODEL statement. This section discusses how to specify different types of effects and explains how they relate to the columns of the design matrix.

In the following list of effect types, assume that A, B, and C are classification variables and X1, X2, and X3 are continuous variables:

- Regressor effects are specified by writing continuous variables by themselves, as follows:

  ```
  X1  X2  X3
  ```

  For regressor effects, the actual values of the variable are used in the design matrix.

- Polynomial effects are specified by joining two or more continuous variables with asterisks, as follows:

  ```
  X1*X1  X1*X1*X1  X1*X2  X1*X2*X3  X1*X1*X2
  ```

  Polynomial effects are also referred to as interactions or crossproducts of continuous variables. When a variable is joined with itself, polynomial effects are referred to as quadratic effects, cubic effects, and so on. In the preceding examples, the first two effects are the quadratic and cubic effects for X1, respectively. The remaining effects are crossproducts.

  For polynomial effects, the value used in the design matrix is the product of the values of the constituent variables.

- Main effects are specified by writing classification variables by themselves. as follows:

  ```
  A  B  C
  ```

  If a classification variable A has k levels, then its main effect has $k - 1$ degrees of freedom, corresponding to $k - 1$ independent differences between the mean response at different levels.

Most designs involve main effects because they correspond to the factors in your experiment. For example, in a factorial experiment for a chemical process, the main effects can be metal type, temperature, pressure, and the level of a catalyst.

For information about how the OPTEX procedure generates the $k - 1$ columns in the design matrix that correspond to the main effects of a classification variable, see the section "Design Coding" on page 1032.

- Crossed effects (interactions) are specified by joining class variables with asterisks, as follows:

  A*B B*C A*B*C

  The number of degrees of freedom for a crossed effect is the product of the numbers of degrees of freedom for the constituent main effects. The columns in the design matrix that correspond to a crossed effect are formed by the horizontal direct products of the constituent main effects.

- Continuous-by-class effects are specified by joining continuous variables and classification variables with asterisks, as follows:

  X1*A

  The design columns for a continuous-by-class effect are constructed by multiplying the values in the design columns for the continuous variables and the classification variable.

All design matrices start with a column of ones for the assumed intercept term unless you use the NOINT option in the MODEL statement.

Bar and @ Operators

You can shorten the specification of a factorial model by using the bar operator. For example, the following statements show two ways of specifying a full three-way factorial model:

```
model a b c a*b a*c b*c a*b*c;
model a|b|c;
```

When the vertical bar (**|**) is used, the right- and left-hand sides become effects, and their cross becomes an effect. Multiple bars are permitted. The expressions are expanded from left to right by using rules given by Searle (1971). For example, **A|B|C** is evaluated as follows:

```
A | B | C  →  { A | B } | C
           →  { A B A*B } | C
           →  A  B  A*B  C  A*C  B*C  A*B*C
```

The bar operator does not cross a variable with itself. To produce a quadratic term, you must specify it directly.

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by putting it at the end of a bar effect, preceded by an **@** sign. For example, the specification **A|B|C@2** results in only those effects that contain two or fewer variables (in this case A, B, A*B, C, A*C, and B*C).

Examples of Models

Main Effects Model For a three-factor main effects model with A, B, and C as the factors, the MODEL statement is

```
model a b c;
```

Factorial Model with Interactions To specify interactions in a factorial model, join effects with asterisks, as described previously. For example, the following statements show two ways of specifying a complete factorial model, which includes all the interactions:

```
model a b c a*b a*c b*c a*b*c;
model a|b|c;
```

Quadratic Model The following statements show two ways of specifying a model with crossed and quadratic effects (for a central composite design, for example):

```
model x1 x2 x1*x2 x3 x1*x3 x2*x3
      x1*x1 x2*x2 x3*x3;
model x1|x2|x3@2 x1*x1 x2*x2 x3*x3;
```

Design Efficiency Measures

The output from the OPTEX procedure includes efficiency measures for the resulting designs according to various criteria. This section gives the precise definitions for these measures.

By default, the OPTEX procedure calculates the following efficiency measures for each design that it finds in its search for an optimum design:

$$\begin{aligned}
\text{D-efficiency} &= 100 \times \left(\frac{|\mathbf{X}'\mathbf{X}|^{1/p}}{N_D} \right) \\
\text{A-efficiency} &= 100 \times \left(\frac{p/N_D}{\text{trace}(\mathbf{X}'\mathbf{X})^{-1}} \right) \\
\text{G-efficiency} &= 100 \times \left(\sqrt{\frac{p/N_D}{\max_{\mathbf{x}\in\mathcal{C}} \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}}} \right)
\end{aligned}$$

where p is the number of parameters in the linear model, N_D is the number of design points, and $\mathcal{C}$ is the set of candidate points. The D- and A-efficiencies are the relative number of runs (expressed as percentages) that are required by a hypothetical orthogonal design to achieve the same $|\mathbf{X}'\mathbf{X}|$ and $\text{trace}(\mathbf{X}'\mathbf{X})^{-1}$, respectively (Mitchell 1974b).

When you specify a BLOCKS statement, the D- and A-efficiencies for the treatment part of the model are calculated. They are calculated similarly to the preceding efficiencies, except that they are based on the information matrix after correcting for block and covariate effects. This matrix can be written as $\mathbf{X}'\mathbf{A}^{-1}\mathbf{X}$ for a symmetric, positive definite matrix A that depends on the model for the block and covariate effects. If you specify a block structure or a covariate model, then $\mathbf{A} = \mathbf{A}^{-1} = \mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'$, where $\mathbf{Z}'$ is the design

matrix for the block and covariate effects. Alternatively, you can use the COVAR= option to specify the matrix **A** directly. Given **A**, the efficiencies in the presence of covariates are defined as follows:

$$\begin{array}{rcl@{\qquad}rcl} \text{D-efficiency} & = & 100 \times c_D^{-1} \cdot |\mathbf{X}'\mathbf{A}^{-1}\mathbf{X}|^{1/p}/N, & c_D & = & \prod_{i=1}^{p} \lambda_i^{1/p} \\ \text{A-efficiency} & = & 100 \times c_A^{-1} \cdot (p/N)/\text{trace}(\mathbf{X}'\mathbf{A}^{-1}\mathbf{X})^{-1}, & c_A & = & \sum_{i=1}^{p} \lambda_i/p \end{array}$$

where $\lambda_1, \ldots, \lambda_p$ are the p largest eigenvalues of $\mathbf{A}^{-1}$. If you use the STRUCTURE= block model specification and the treatment model has only one classification variable, then the design fits into the traditional block design framework. In this case, the D-efficiency relative to a balanced incomplete block design is also listed.

Because these efficiencies measure the goodness of the design relative to theoretical designs that might be far from possible in many cases, they are typically not useful as absolute measures of design goodness. Instead, efficiency measures should be used relatively, to compare one design to another for the same situation.

For the distance-based criteria, there are no simple measures of design efficiency that can be scaled from 0 to 100. For a definition of the design measures tabulated for these criteria, see the section "Output" on page 1041.

Design Coding

The way the independent effects of the model are interpreted to generate a linear model is called *coding*. The OPTEX procedure provides for different types of coding. For D-optimality, the type of coding affects only the absolute value of the computed efficiency criteria, not the relative values for two different designs. Thus, different codings do not affect the choice of D-optimal design. In this section, the details and ramifications of the different types of coding are discussed.

Coding the points in a design involves selecting linearly independent columns that correspond to each model term, turning particular values of the factors into a row vector **x**. The OPTEX procedure requires a *nonsingular* coding for the design matrix. Because of this, any two coding schemes are related by a nonsingular transformation.

Static Coding

The default coding for the design points is as follows:

- Unless you specify CODING=NONE (or NOCODE) in the PROC OPTEX statement, continuous variables are centered and scaled so that their maximum and minimum values are 1 and –1, respectively.
- The $k-1$ columns that correspond to the main effect of a classification variable A are computed as follows: For a design point with A at its ith level, for $1 \leq i \leq k-1$, the columns of the design matrix associated with A are all 0 except for the ith column, which is 1. When A is at its kth level, all $k-1$ columns associated with A are –1. Thus, if α_i denotes the expected response at the ith level of A, the $k-1$ columns yield estimates of $\alpha_1 - \alpha_k, \alpha_2 - \alpha_k, \ldots, \alpha_{k-1} - \alpha_k$.
- Columns for crossed effects are computed by taking the horizontal direct product of columns that correspond to the constituent effects.

This coding corresponds to modeling without *overparameterization*, by using the same method as the CATMOD procedure in SAS/STAT software uses. This is different from the method used by the GLM procedure, which uses an overparameterized model.

Orthogonal Coding

If you specify CODING=ORTH or CODING=ORTHCAN, the points are first coded as described in the previous section and then recoded so that $\mathbf{X}_C'\mathbf{X}_C = N_C \cdot \mathbf{I}$, where $\mathbf{X}_C$ is the design matrix for the candidate points, N_C is the number of candidates, and $\mathbf{I}$ is the identity matrix. This is required in order for the D- and A-efficiency measures to make sense. For the CODING=ORTHCAN option, this recoding is accomplished by computing a square matrix $\mathbf{R}$ such that $\mathbf{X}_C'\mathbf{X}_C = \mathbf{R}'\mathbf{R}$ and then transforming each row vector $\mathbf{x}$ as

$$\mathbf{x} \rightarrow \mathbf{x}\mathbf{R}^{-1}\sqrt{N_C}$$

If you specify CODING=ORTH, the recoding is done in a similar fashion, except that the matrix $\mathbf{R}$ is computed according to $\mathbf{X}_C'\mathbf{X}_C + \mathbf{X}_A'\mathbf{X}_A + \mathbf{X}_I'\mathbf{X}_I = \mathbf{R}'\mathbf{R}$, where $\mathbf{X}_A$ and $\mathbf{X}_I$ are the design matrices for the AUGMENT= and INITDESIGN= data sets, respectively (coded as described in the previous section.) Thus, these two orthogonal coding options differ only when there is an AUGMENT= or an INITDESIGN= data set; the CODING=ORTH option includes points from these data sets in computing the orthogonal coding, whereas the CODING=ORTHCAN option uses only the candidates themselves.

Example of Coding

For example, consider a main effect model that has one continuous variable X and one three-level classification variable A. The results of the various coding options are shown in Table 15.7.

Table 15.7 Different Types of Design Coding

Original Data		Design Matrix with CODING=NONE				Design Matrix with CODING=STATIC				Design Matrix with CODING=ORTH			
X	**A**		**X**	**A1**	**A2**		**X**	**A1**	**A2**		**X**	**A1**	**A2**
1	1	1	1	1	0	1	–1	1	0	1	–1.464	0.598	–0.707
2	2	1	2	0	1	1	–0.6	0	1	1	–0.878	–0.478	1.414
3	3	1	3	–1	–1	1	–0.2	–1	–1	1	–0.293	–1.554	–0.707
4	1	1	4	1	0	1	0.2	1	0	1	0.293	1.554	–0.707
5	2	1	5	0	1	1	0.6	0	1	1	0.878	0.478	1.414
6	3	1	6	–1	–1	1	1	–1	–1	1	1.464	–0.598	–0.707

The first column in each design matrix is an all-ones vector that corresponds to the intercept, the next column corresponds to the linear effect of X, and the last two columns correspond to the two degrees of freedom for the main effect of A.

General Recommendations

Coding does not affect the relative ordering of designs by D-efficiency, and the same is true for G-efficiency and the average standard error of prediction. This is easy to see for the latter two measures, which are based on the variance of prediction, because how accurately a point is predicted should not be affected by how the independent variables are coded. For D-optimality, note again that coding corresponds to multiplying the design matrix on the right by some nonsingular transformation A, which changes the determinant of the information matrix as follows:

$$|\mathbf{X}'\mathbf{X}| \rightarrow |\mathbf{A}'\mathbf{X}'\mathbf{X}\mathbf{A}| = |\mathbf{A}'\mathbf{A}||\mathbf{X}'\mathbf{X}| = |\mathbf{A}|^2|\mathbf{X}'\mathbf{X}|$$

Thus, recoding simply multiplies the D-criterion by a constant that is the same for all designs. However, A-optimality is *not* invariant to coding.

Orthogonal coding will usually be the right one; it is not the default because it depends on the candidate set. However, for the distance-based criteria, if the distance between two points should be computed in terms of the actual values of the model variables instead of centered and scaled values, then you should specify CODING=NONE or NOCODE. The NOCODE option can also be useful when the NOINT option is specified.

Optimality Criteria

An optimality criterion is a single number that summarizes how good a design is, and it is maximized or minimized by an optimal design. This section discusses in detail the optimality criteria available in the OPTEX procedure.

Types of Criteria

Two general types of criteria are available: *information-based* criteria and *distance-based* criteria.

The information-based criteria that are directly available are D- and A-optimality; they are both related to the information matrix $\mathbf{X}'\mathbf{X}$ for the design. This matrix is important because it is proportional to the inverse of the variance-covariance matrix for the least squares estimates of the linear parameters of the model. Roughly, a good design should "minimize" the variance $(\mathbf{X}'\mathbf{X})^{-1}$, which is the same as "maximizing" the information $\mathbf{X}'\mathbf{X}$. D- and A-efficiency are different ways of saying how large $(\mathbf{X}'\mathbf{X})$ or $(\mathbf{X}'\mathbf{X})^{-1}$ are.

For the distance-based criteria, the candidates are viewed as comprising a point cloud in p-dimensional Euclidean space, where p is the number of terms in the model. The goal is to choose a subset of this cloud that "covers" the whole cloud as uniformly as possible (in the case of U-optimality) or that is as broadly "spread" as possible (in the case of S-optimality). These ideas of coverage and spread are defined in detail in the section "Distance-Based Criteria" on page 1036. The distance-based criteria thus correspond to the intuitive idea of filling the candidate space as well as possible.

The rest of this section discusses different optimality criteria in detail.

D-Optimality

D-optimality is based on the determinant of the information matrix for the design, which is the same as the reciprocal of the determinant of the variance-covariance matrix for the least squares estimates of the linear parameters of the model.

$$|\mathbf{X}'\mathbf{X}| \quad = \quad 1/|(\mathbf{X}'\mathbf{X})^{-1}|$$

The determinant is thus a general measure of the size of $(\mathbf{X}'\mathbf{X})^{-1}$. D-optimality is the most commonly used criterion for generating optimal designs and is therefore the default criterion for the OPTEX procedure.

The D-optimality criterion has the following characteristics:

- D-optimality is the most computationally efficient criterion to optimize for the low-rank update algorithms of the OPTEX procedure, because each update depends only on the variance of prediction for the current design; see the section "Useful Matrix Formulas" on page 1037.

- $|\mathbf{X}'\mathbf{X}|$ is inversely proportional to the size of a $100(1-\alpha)\%$ confidence ellipsoid for the least squares estimates of the linear parameters of the model.
- $|\mathbf{X}'\mathbf{X}|^{1/p}$ is equal to the geometric mean of the eigenvalues of $\mathbf{X}'\mathbf{X}$.
- The D-optimal design is invariant to nonsingular recoding of the design matrix.

$$|\mathbf{X}'\mathbf{X}| \;\rightarrow\; |\mathbf{A}'\mathbf{X}'\mathbf{X}\mathbf{A}| \;=\; |\mathbf{A}'\mathbf{A}||\mathbf{X}'\mathbf{X}| \;=\; |\mathbf{A}|^2|\mathbf{X}'\mathbf{X}|$$

A-Optimality

A-optimality is based on the sum of the variances of the estimated parameters for the model, which is the same as the sum of the diagonal elements, or trace, of $(\mathbf{X}'\mathbf{X})^{-1}$. Like the determinant, the A-optimality criterion is a general measure of the size of $(\mathbf{X}'\mathbf{X})^{-1}$. A-optimality is less commonly used than D-optimality as a criterion for computer optimal design, partly because it is more computationally difficult to update (see the section "Useful Matrix Formulas" on page 1037). Also, A-optimality is *not* invariant to nonsingular recoding of the design matrix; different designs will be optimal with different codings.

G- and I-Optimality

Both G-efficiency and the average prediction variance are well-known criteria for optimal design. Both are based on the variance of prediction of the candidate points, which is proportional to $\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}$. As this formula shows, these two criteria are also related to the information matrix $\mathbf{X}'\mathbf{X}$. Minimizing the average prediction variance has also been called *I-optimality*, the "I" denoting integration over the candidate space.

It is possible to apply the search techniques available in the OPTEX procedure to these two criteria, but this turns out to be a poor way to find G- and I-optimal designs. One reason for this is that there are no efficient low-rank update rules (see the section "Useful Matrix Formulas" on page 1037), so that the searches can take a very long time. More seriously, for G-optimality such a search often does not converge on a design with good G-efficiency. G-efficiency is simply too "rough" a criterion to be optimized by the relatively short steps of the search algorithms available in the OPTEX procedure.

However, the OPTEX procedure does offer an approach for finding G-efficient designs. Begin by searching for designs according to the default D-optimality criterion. Then, from the various designs found on the different tries, you can save the one that has the best G-efficiency by specifying the NUMBER=GBEST option in the OUTPUT statement. Because D- and G-efficiency are highly correlated over the space of all designs, this method usually results in adequately G-efficient designs, especially when the number of tries is large (see Nguyen and Piepel (2005)). For more information about specifying the number of tries, see the ITER= option.

To find I-optimal designs, note that if the design is orthogonally coded then I-optimality is equivalent to the A-optimality, because the sum of the prediction variances of all points $\mathbf{x}$ in the candidate space $\mathcal{C}$ is

$$\begin{aligned}
\sum_{\mathbf{x}\in\mathcal{C}} \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x} &= \sum_{\mathbf{x}\in\mathcal{C}} \text{trace}\left(\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}\right) \\
&= \text{trace}\left((\mathbf{X}'\mathbf{X})^{-1}\sum_{\mathbf{x}\in\mathcal{C}}\mathbf{x}\mathbf{x}'\right) \\
&= \text{trace}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_{\mathcal{C}}'\mathbf{X}_{\mathcal{C}}\right) \\
&= N_{\mathcal{C}} \cdot \text{trace}\left((\mathbf{X}'\mathbf{X})^{-1}\right)
\end{aligned}$$

where N_C is the number of candidate points and $\mathbf{X}_C$ is the design matrix for the candidate points. Thus, you can use the CODING=ORTH option in the PROC OPTEX statement together with the CRITERION=A option in the GENERATE statement to search for I-optimal designs.

Note that both G- and I-optimality are invariant to nonsingular recoding of the design matrix, because the coding does not affect how well a point is predicted.

Distance-Based Criteria

The distance-based criteria are based on the distance $d(\mathbf{x}, \mathcal{A})$ from a point $\mathbf{x}$ in the p-dimensional Euclidean space $\mathcal{R}^p$ to a set $\mathcal{A} \subset \mathcal{R}^p$. This distance is defined as follows:

$$d(\mathbf{x}, \mathcal{A}) = \min_{\mathbf{y} \in \mathcal{A}} ||\mathbf{x} - \mathbf{y}||$$

where $||\mathbf{x} - \mathbf{y}||$ is the usual p-dimensional Euclidean distance,

$$||\mathbf{x} - \mathbf{y}|| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_p - y_p)^2}$$

U-optimality seeks to minimize the sum of the distances from each candidate point to the design

$$\sum_{\mathbf{x} \in \mathcal{C}} d(\mathbf{x}, \mathcal{D})$$

where $\mathcal{C}$ is the set of candidate points and $\mathcal{D}$ is the set of design points. You can visualize the U criterion by associating with any design point those candidates to which it is closest. Thus, the design defines a *clustering* of the candidate set, and indeed cluster analysis has been used in this context. Johnson, Moore, and Ylvisaker (1990) consider a similar measure of design efficiency, but over infinite rather than finite candidate spaces. Computationally, the U-optimality criterion can be *very* difficult to optimize, especially if the matrix of all pairwise distances between candidate points does not fit in memory. In this case, the OPTEX procedure recomputes each distance as needed. When searching for a U-optimal design, you should start with a small version of the problem to get an idea of the computing resources required.

S-optimality seeks to maximize the harmonic mean distance from each design point to all the other points in the design.

$$\frac{N_D}{\sum_{\mathbf{y} \in \mathcal{D}} 1/d(\mathbf{y}, \mathcal{D} - \mathbf{y})}$$

For an S-optimal design, the distances $d(\mathbf{y}, \mathcal{D} - \mathbf{y})$ are large, so the points are as spread out as possible. Because the S-optimality criterion depends only on the distances between design points, it is usually computationally easier to compute and optimize than the U-optimality criterion, which depends on the distances between all pairs of candidate points.

Memory and Run-Time Considerations

The OPTEX procedure provides a computationally intensive approach to designing an experiment, and therefore some finesse is called for to make the most efficient use of computer resources.

The OPTEX procedure must retain the entire set of candidate points in memory because all the search algorithms access these points repeatedly. If this requires more memory than is available, consider using knowledge of the problem to reduce the set of candidate points. For example, for first- or second-order models, it is usually adequate to restrict the candidates to just the center and the edges of the experimental region or perhaps an even smaller set; see the introductory examples in the sections "Handling Many Variables" on page 1008 and "Constructing a Mixture-Process Design" on page 1009.

The distance-based criteria (CRITERION=U and CRITERION=S) also require repeated access to the distance between candidate points. PROC OPTEX tries to fit the matrix of these distances in memory; if it cannot, it recomputes them as needed, but this causes the search to be dramatically slower.

The run time of each search algorithm depends primarily on N_D, the size of the target design, and on N_C, the number of candidate points. For a particular model, the run times of the sequential, exchange, and DETMAX algorithms are all roughly proportional to both N_D and N_C—that is, $O(N_D) + O(N_C)$. The run times for the two simultaneous switching algorithms (FEDOROV and M_FEDOROV) are approximately proportional to the product of N_D and N_C—that is, $O(N_C N_D)$. The constant of proportionality is larger when searching for A-optimal designs because the update formulas are more complicated (see the section "Search Methods," which follows).

For problems where either N_D or N_C is large, it is a good idea to make a few test runs with a faster algorithm and a small number of tries before attempting to use one of the slower and more reliable search algorithms. For most problems, the efficiency of a design that a faster algorithm finds will be within 1–2% of that for the best possible design, and this is usually sufficient if it appears that searching with a slower algorithm is infeasible.

Search Methods

The search procedures available in the OPTEX procedure offer various compromises between speed and reliability in finding the optimum. In general, the longer an algorithm takes to arrive at an answer, the more efficient is the resulting design, although this is not invariably true. The right search procedure for any specific case depends on the size of the problem, the relative importance of using the best possible design as opposed to a very good one, and the computing resources available.

Useful Matrix Formulas

All of the search algorithms are based on adding candidate points to a design and deleting them from this design. If $\mathbf{V} = (\mathbf{X}'\mathbf{X})^{-1}$ is the inverse of the information matrix for the design at any stage, then the change in $\mathbf{V}$ that results from adding a point $\mathbf{x}$ to a design (which adds a new row $\mathbf{x}$ to the design matrix) is

$$\mathbf{V} \quad \rightarrow \quad \mathbf{V} - \frac{\mathbf{V}\mathbf{x}\mathbf{x}'\mathbf{V}}{1 + \mathbf{x}'\mathbf{V}\mathbf{x}}$$

and the change in $\mathbf{V}$ that results from deleting the point $\mathbf{y}$ from this design is

$$\mathbf{V} \quad \rightarrow \quad \mathbf{V} + \frac{\mathbf{V}\mathbf{y}\mathbf{y}'\mathbf{V}}{1 - \mathbf{y}'\mathbf{V}\mathbf{y}}$$

It follows that adding $\mathbf{x}$ multiplies the determinant of the information matrix by $1 + \mathbf{x}'\mathbf{V}\mathbf{x}$. Likewise, deleting $\mathbf{y}$ multiplies the determinant by $1 - \mathbf{y}'\mathbf{V}\mathbf{y}$. For any point $\mathbf{z}$, the quantity $\mathbf{z}'\mathbf{V}\mathbf{z}$ is proportional to the prediction

variance at the point $\mathbf{z}$. Thus, the point $\mathbf{x}$ whose addition to the design maximizes the determinant of the information is the point whose prediction variance calculated from the present design is largest. The point whose deletion from the design costs the least in terms of the determinant is the point with the smallest prediction variance.

Similar rank-one update formulas can be derived for A-optimality, which is based on the trace of the inverse of the information matrix instead of its determinant. However, in this case there is no single quantity that can be examined for both adding and deleting a point. Here, the trace that results from adding a point $\mathbf{x}$ to a design depends on

$$\frac{\mathbf{x}'V^2\mathbf{x}}{1+\mathbf{x}'V\mathbf{x}}$$

and the trace that results from deleting a point $\mathbf{y}$ to this design depends on

$$\frac{\mathbf{y}'V^2\mathbf{y}}{1-\mathbf{y}'V\mathbf{y}}$$

This complication makes A-optimal designs harder to search for than D-optimal ones.

There are no useful rank-one update formulas for the distance-based design criteria.

Sequential Search Algorithm

The simplest and fastest algorithm is the sequential search due to Dykstra (1971), which starts with an empty design and adds successive candidate points so that the chosen criterion is optimized at each step. You can use the sequential procedure as a first step in finding a design to judge the size of the problem in terms of time and space requirements and to determine the number of design points needed to estimate the parameters of the model.

The sequential algorithm requires no initial design; in fact, it can be used to provide an initial design for the other search procedures (see the INITDESIGN= option on page 1021). If you specify a data set for an initial design for this search procedure, no search will be made; in this way, you can use the OPTEX procedure to evaluate an existing design.

Because the sequential search method involves no randomness, it requires only one try to find a design. The sequential procedure is by far the fastest of any of the search methods, but in difficult design situations it is also the least reliable in finding a globally optimal design. Also, the fact that it always finds the same design (due to the lack of randomness mentioned previously) makes it inappropriate when you want to find a design that represents a compromise between several optimality criteria.

Exchange Algorithm

The next fastest algorithm is the simple exchange method of Mitchell and Miller (1970). This technique tries to improve an initial design by adding a candidate point and then deleting one of the design points, stopping when the chosen criterion ceases to improve. This method is relatively fast (though typically much slower than the sequential search) and fairly reliable. METHOD=EXCHANGE is the default.

DETMAX Algorithm

The DETMAX algorithm of Mitchell (1974a) is the best-known and most widely used optimal design search algorithm. It generalizes the simple exchange method. Instead of requiring that each addition of a point be followed directly by a deletion, the algorithm provides for *excursions* in which the size of the design might vary between $N_D + k$ and $N_D - k$, where N_D is the specified size of the design and k is the maximum allowed size for an excursion. By default k is 4, but you can change this (see the METHOD=DETMAX(*level*) option on page 1023). For the precise stopping rules for each excursion and for the entire search, see Mitchell (1974a). Due to the mentioned excursions, the DETMAX algorithm might not be a good choice when the design you want to construct is saturated or near-saturated.

Fedorov and Modified Fedorov Algorithms

The three algorithms discussed so far add and delete points one at a time. By contrast, the Fedorov and modified Fedorov algorithms are based on simultaneous switching—that is, adding and deleting points simultaneously. These two algorithms usually find a better design than the others, but because each step involves a search over all possible pairs of candidate and design points, they generally run much slower.

From the equations in the section "Useful Matrix Formulas" on page 1037 (see also Nguyen and Piepel (2005, sec. 4)), it follows that simultaneously adding a point $\mathbf{x}$ and deleting a point $\mathbf{y}$ multiplies the determinant of the information matrix by $1 + \Delta(x, y)$, where:

$$\Delta(\mathbf{x}, \mathbf{y}) = \mathbf{x}'\mathbf{V}\mathbf{x} - \mathbf{y}'\mathbf{V}\mathbf{y} + (\mathbf{x}'\mathbf{V}\mathbf{y})^2 - (\mathbf{x}'\mathbf{V}\mathbf{x})(\mathbf{y}'V\mathbf{y})$$

The quantity $\Delta(\mathbf{x}, \mathbf{y})$ is often referred to as Fedorov's delta function.

At each step, the Fedorov algorithm (Fedorov 1972) seeks the pair $(\mathbf{x}, \mathbf{y})$ of one candidate point and one design point that maximizes $\Delta(\mathbf{x}, \mathbf{y})$ and then switches $\mathbf{x}$ for $\mathbf{y}$ in the design. Thus, after computing $\Delta(\mathbf{x}, \mathbf{y})$ for all possible pairs of candidate and design points, the Fedorov algorithm performs only one switch.

The modified Fedorov algorithm of Cook and Nachtsheim (1980) computes the same number of Δ's on each step but switches each point $\mathbf{y}$ in the design with the candidate point $\mathbf{x}$ that maximizes $\Delta(\mathbf{x},\mathbf{y})$. This procedure is generally as reliable as the simple Fedorov algorithm in finding the optimal design, but it can be up to twice as fast.

Johnson and Nachtsheim (1983) introduce a generalization of both the simple exchange algorithm and the modified Fedorov search algorithm of Cook and Nachtsheim (1980), which is described later in this list. In the modified Fedorov algorithm, each of the points in the current design is considered for exchange with a candidate point; in the generalized version, only the k design points that have smallest variance in the current design are considered for exchange. You can specify k-exchange as the search procedure for OPTEX by specifying a value for k in parentheses after METHOD=EXCHANGE. When $k = N_D$ (the size of the design), k-exchange is equivalent to the modified Fedorov algorithm; when $k = 1$, it is equivalent to the simple exchange algorithm. Cook and Nachtsheim (1980) indicate that $k < N_D/4$ is typically sufficient.

For a detailed review of the preceding search methods, see Nguyen and Miller (1992).

Optimal Blocking

Building on the work of Harville (1974), Cook and Nachtsheim (1989) give an algorithm for finding D-optimal designs in the presence of fixed block effects. In this case, the design for the original candidate points

is called the *treatment* design. The information matrix for the treatment design has the form $\mathbf{X}'\mathbf{A}\mathbf{X}$ for a certain symmetric, nonnegative-definite matrix $\mathbf{A}$ that depends on the blocks. The algorithm is based on two kinds of low-rank changes to the treatment design matrix $\mathbf{X}$: *exchanging* a point in the design with a potential treatment point, and *interchanging* two points in the design. Cook and Nachtsheim (1989) give formulas for computing the resulting change in $\mathbf{X}'\mathbf{A}\mathbf{X}$ and $|\mathbf{X}'\mathbf{A}\mathbf{X}|$. These update formulas can be generalized to apply whenever the information matrix for the treatment design has the form $\mathbf{X}'\mathbf{A}\mathbf{X}$, not just when $\mathbf{A}$ is derived from fixed blocks. This is the basis for the optimal blocking algorithm in the OPTEX procedure.

Notice that you can combine several options to use the OPTEX procedure to evaluate a design with respect to the fixed covariates. Assume the design you want to evaluate is in a data set named Edesign. Then first specify the GENERATE statement to make the data set Edesign the treatment design:

```
generate initdesign=Edesign method=sequential;
```

Then specify the following BLOCKS statement options:

```
blocks {block-specification} init=chain iter=0;
```

The INIT=CHAIN option ensures that the starting ordering for the treatment points is the same as in the Edesign data set, and the ITER=0 specification causes the procedure simply to output the efficiencies for the initial design, without trying to optimize it.

Search Strategies

General Recommendations

As with all combinatorial optimization problems, finding efficient experimental designs can be difficult. For this reason, the OPTEX procedure provides a variety of ways to customize the search.

Although default settings make the procedure simple to use "as is," you can usually improve the search by using knowledge of the specific design problem. For example, if the default algorithm (METHOD=EXCHANGE) runs quickly but does not clearly indicate it finds the best design, you can try a slower but more reliable search method or use more iterations than the default number of 10.

Set of Candidate Points

The choice of candidate points can profoundly affect both the speed with which the search converges at a local optimum and the likelihood that this local optimum is indeed the global optimum. Up to a point, the more candidate points there are, the better the resulting optimum design will be but the longer it will take to find. Any prior knowledge that can be brought to bear on the choice of candidates will almost certainly improve the search. For example, for first- or second-order models it is usually adequate to restrict the candidates to just the center and the edges of the experimental region, or perhaps even less; see Snee (1985), and see the introductory examples in the sections "Handling Many Variables" on page 1008 and "Constructing a Mixture-Process Design" on page 1009.

Initial Design

The reliability of the search algorithms in finding the optimal design can be quite sensitive to the choice of initial design. The default method of initialization for each search procedure should achieve good results for a wide variety of situations (see the INITDESIGN= option on page 1021). However, in certain

situations it is better to override the defaults. For example, if there are many local optima and you want to find the exact global optimum, it is probably best to start each try with a completely random design (INITDESIGN=RANDOM). On the other hand, prior knowledge might provide a specific initial design, which can be placed in a SAS data set and specified with the INITDESIGN= option.

Output

By default, the OPTEX procedure lists the following information for each attempt to find the optimum design:

- the D-efficiency of the design
- the A-efficiency of the design
- the G-efficiency of the design
- the square root of the average variance for prediction over the candidate points

If you specify a BLOCKS statement, then the covariate-adjusted D- and A-efficiencies are also listed.

For more information about the efficiencies, see the section "Design Efficiency Measures" on page 1031. The OPTEX procedure orders the designs first by the optimality criteria with which they were generated and then by optimality with respect to the other three preceding measures.

If you use the NOCODE option, the OPTEX procedure lists the following:

- $\log|\mathbf{X}'\mathbf{X}|$
- $\mathrm{trace}(\mathbf{X}'\mathbf{X})^{-1}$
- the G-efficiency of the design
- the square root of the average variance for prediction over the candidate points

If you specify one of the distance-based optimality criteria (CRITERION=U or CRITERION=S), then PROC OPTEX lists alternative measures of coverage and spread instead of the preceding efficiencies. For U-optimality the following measures are listed:

- the average distance from each candidate to the nearest design point (this is the U criterion)
- the average harmonic mean distance from each candidate to the design

For S-optimality, the following alternative measures of spread are listed:

- the harmonic mean distance from each design point to the nearest other design point (this is the S criterion)
- the average distance from each design point to the nearest other design point

In addition, the OPTEX procedure can create an output data set, as described in the sections "OUTPUT Statement" on page 1025 and "Output Data Sets" on page 1028.

ODS Tables

The following table summarizes the ODS tables that you can request with the PROC OPTEX statement.

Table 15.8 ODS Tables Produced in PROC OPTEX

ODS Table Name	Description	Statement	Option
ClassLevels	Classification variable levels	CLASS	Default
FactorRanges	Continuous variable ranges	Default	Default
BlockDesignEfficiencies	Block design efficiency criteria	BLOCK	Default
Efficiencies	Efficiency criteria for all designs	GENERATE	Default
Criteria	Efficiency criteria for a single design	EXAMINE	Default
Points	Design points	EXAMINE	POINTS
Information	Information matrix XPX	EXAMINE	INFORMATION
Variance	Inverse information matrix inv(XPX)	EXAMINE	VARIANCE
Status	Optimization status	PROC	STATUS
Distances	Distance criteria for all designs	GENERATE	CRITERION=U or S

Examples: OPTEX Procedure

Example 15.1: Nonstandard Linear Model

NOTE: See *A Nonstandard Linear Model* in the SAS/QC Sample Library.

This example is based on an example in Mitchell (1974a). An animal scientist wants to compare wildlife densities in four different habitats over a year. However, due to the cost of experimentation, only 12 observations can be made. The following model is postulated for the density $y_j(t)$ in habitat j during month t:

$$y_j(t) = \mu_j + \beta t + \sum_{i=1}^{4} a_i \cos(i\pi t/4) + \sum_{i=1}^{3} b_i \sin(i\pi t/4)$$

This model includes the habitat as a classification variable, the effect of time with an overall linear drift term βt, and cyclic behavior in the form of a Fourier series. There is no intercept term in the model.

The OPTEX procedure is used because there are no standard designs that cover this situation. The candidate set is the full factorial arrangement of four habitats by 12 months, which can be generated with a DATA step, as follows:

```
data a;
   drop theta pi;
   array c{4} c1-c4;
   array s{3} s1-s3;
   pi = arcos(-1);
   do Habitat=1 to 4;
      do Month=1 to 12;
         theta = pi * Month / 4;
         do i=1 to 4; c{i} = cos(i*theta); end;
         do i=1 to 3; s{i} = sin(i*theta); end;
         output;
      end;
   end;
run;
```

Data set a contains the 48 candidate points and includes the four cosine variables (c1, c2, c3, and c4) and three sine variables (s1, s2, and s3). The following statements produce Output 15.1.1:

```
proc optex seed=193030034 data=a;
   class    Habitat;
   model    Habitat Month c1-c4 s1-s3 / noint;
   generate n=12;
run;
```

Output 15.1.1 Sampling Wildlife Habitats over Time

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	31.6103	19.7379	57.7350	1.3229
2	31.6103	19.7379	57.7350	1.3229
3	31.6103	19.7379	57.7350	1.3229
4	31.6103	19.3793	57.7350	1.3229
5	31.6103	19.2916	57.7350	1.3229
6	31.6103	19.0335	57.7350	1.3229
7	30.1304	14.8837	44.7214	1.4907
8	30.1304	14.2433	44.7214	1.5092
9	30.1304	13.1687	44.7214	1.5456
10	28.1616	9.8842	40.8248	1.7559

The best determinant (D-efficiency) was found in 6 out of the 10 tries. Thus, you can be confident that this is the best achievable determinant. Only the A-efficiency distinguishes among the designs listed in Output 15.1.1. The best design has an A-efficiency of 19.74%, whereas another design has the same D-efficiency but a slightly smaller A-efficiency of 19.03%, or about 96% relative A-efficiency. To explore the differences, you can save the designs in data sets and print them. Because the OPTEX procedure is interactive, you need to submit only the following statements (immediately after the preceding statements) to produce Output 15.1.2 and Output 15.1.3:

```
   output out=d1 number=1;
run;
   output out=d6 number=6;
run;

proc sort data=d1;
   by  Month Habitat;
run;
proc print data=d1;
   var Month Habitat;
run;

proc sort data=d6;
   by  Month Habitat;
run;
proc print data=d6;
   var Month Habitat;
run;
```

Output 15.1.2 The Best Design

Obs	Month	Habitat
1	1	3
2	2	2
3	3	4
4	4	1
5	5	4
6	6	1
7	7	2
8	8	3
9	9	4
10	10	1
11	11	2
12	12	3

Output 15.1.3 Design with Lower A-Efficiency

Obs	Month	Habitat
1	1	4
2	2	2
3	3	3
4	4	1
5	5	1
6	6	4
7	7	4
8	8	1
9	9	2
10	10	1
11	11	4
12	12	3

Note the structure of the best design in Output 15.1.2. One habitat is sampled in each month, each habitat is sampled three times, and the habitats are sampled in consecutive complete blocks. Even though the design in Output 15.1.3 is as D-efficient as the best, it has almost none of this structure; one habitat is sampled each month, but habitats are not sampled an equal number of times. This demonstrates the importance of choosing a final design on the basis of more than one criterion.

You can try searching for the A-optimal design directly. This takes more time but with only 48 candidate points is not too large a problem. The following statements produce Output 15.1.4:

```
proc optex seed=193030034 data=a;
   class    Habitat;
   model    Habitat Month c1-c4 s1-s3 / noint;
   generate n=12 criterion=A;
run;
```

Output 15.1.4 Searching Directly for an A-Efficient Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	31.6103	19.7379	57.7350	1.3229
2	30.1304	17.8273	52.2233	1.3894
3	30.1304	17.7943	52.2233	1.3944
4	30.1304	17.6471	52.2233	1.4093
5	28.1616	15.7055	44.7214	1.4860
6	28.1616	14.5289	44.7214	1.5343
7	28.1616	13.8603	39.2232	1.5811
8	25.0891	11.6152	37.7964	1.8143
9	25.0891	10.7563	37.7964	1.8143
10	25.0891	10.5437	33.3333	1.8930

The best design found is no more A-efficient than the one found previously.

Example 15.2: Comparing the Fedorov Algorithm to the Sequential Algorithm

NOTE: See *Engine Mapping Problem* in the SAS/QC Sample Library.

An automotive engineer wants to fit a quadratic model to fuel consumption data in order to find the values of the control variables that minimize fuel consumption (Vance 1986). The three control variables AFR (air fuel ratio), EGR (exhaust gas recirculation), and SA (spark advance) and their possible settings are shown in the following table:

Variable	Values							
AFR	15	16	17	18				
EGR	0.020	0.177	0.377	0.566	0.921	1.117		
SA	10	16	22	28	34	40	46	52

Rather than run all 192 ($4 \times 6 \times 8$) combinations of these factors, the engineer would like to see whether the total number of runs can be reduced to 50 in an optimal fashion.

Because the factors have different numbers of levels, you can use the PLAN procedure (see *SAS/STAT User's Guide*) to generate the full factorial set to serve as a candidate data set for the OPTEX procedure:

```
proc plan;
   factors AFR=4 ordered EGR=6 ordered SA=8 ordered
       / noprint;
   output out=a
       AFR nvals=(15, 16, 17, 18)
       EGR nvals=(0.020, 0.177, 0.377, 0.566, 0.921, 1.117)
       SA  nvals=(10, 16, 22, 28, 34, 40, 46, 52);
run;
```

The Fedorov algorithm (Fedorov 1972) is generally the most successful optimal design search algorithm, although it also typically can take relatively much longer to run than other algorithms. This algorithm is not the default search method for the OPTEX procedure. However, you can request that it be used by specifying the METHOD=FEDOROV option in the GENERATE statement. For example, the following statements produce Output 15.2.1:

```
proc optex data=a seed=61552;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=fedorov iter=100 keep=10;
run;
```

Output 15.2.1 Efficiencies with the Fedorov Algorithm

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.5246	24.5897	96.3915	0.4231
2	46.5241	24.5901	96.3926	0.4233
3	46.5238	24.5844	96.2306	0.4231
4	46.5237	24.5855	96.2318	0.4233
5	46.5219	24.5866	96.4790	0.4233
6	46.5192	24.5832	96.3070	0.4231
7	46.5192	24.5832	96.3070	0.4231
8	46.5190	24.5741	96.1695	0.4232
9	46.5189	24.5841	96.3062	0.4233
10	46.5188	24.5755	96.3020	0.4234

The Fedorov search method for the preceding problem requires a few seconds for 100 tries on a 2.8GHz desktop PC.

For comparison, you can use the METHOD=SEQUENTIAL option in the GENERATE statement, as shown in the following statements, which produce Output 15.2.2:

```
proc optex data=a seed=33805;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=sequential iter=100 keep=10;
run;
```

Output 15.2.2 Efficiencies with Sequential Algorithm

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.5246	24.5897	96.3915	0.4231
2	46.5241	24.5901	96.3926	0.4233
3	46.5238	24.5844	96.2306	0.4231
4	46.5237	24.5855	96.2318	0.4233
5	46.5219	24.5866	96.4790	0.4233
6	46.5192	24.5832	96.3070	0.4231
7	46.5192	24.5832	96.3070	0.4231
8	46.5190	24.5741	96.1695	0.4232
9	46.5189	24.5841	96.3062	0.4233
10	46.5188	24.5755	96.3020	0.4234

In a fraction of the run time required by the Fedorov method, the sequential algorithm finds a design with a relative D-efficiency of $46.4009/46.5246 = 99.73\%$ compared to the best design found by the Fedorov method, and with *better* A-efficiency. As this demonstrates, if absolute D-optimality is not required, a faster, simpler search might be sufficient.

Example 15.3: Using an Initial Design to Search an Optimal Design

NOTE: See *Engine Mapping Problem* in the SAS/QC Sample Library.

This example is a continuation of Example 15.2.

You can customize the runs used to initialize the search in the OPTEX procedure. For example, you can use the INITDESIGN=SEQUENTIAL option to use an initial design chosen by the sequential search. Or you can place specific points in a data set and use the INITDESIGN=*SAS-data-set* option. In both cases, the search time can be significantly reduced because the search only has to be done once. This example illustrates both of these options.

The previous example compared the results of the DETMAX and sequential search algorithms. You can use the design chosen by the sequential search as the *starting point* for the DETMAX algorithm. The following statements specify the DETMAX search method, replacing the default initialization method with the sequential search:

```
proc optex data=a seed=33805;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=detmax initdesign=sequential;
run;
```

The results, which are displayed in Output 15.3.1, show an improvement over the sequential design itself (Output 15.2.2) but not over the DETMAX algorithm with the default initialization method (Output 15.2.1).

Evidently the sequential design represents a local optimum that is not the global optimum, which is a common phenomenon in combinatorial optimization problems such as this one.

Output 15.3.1 Initializing with a Sequential Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4333	25.0321	95.1371	0.4199

Prior knowledge of the design problem at hand might also provide a specific set of factor combinations to use as the initial design. For example, many D-optimal designs are composed of replications of the optimal saturated design—that is, the optimal design with exactly as many points as there are parameters to be estimated. In this case, there are 10 parameters in the model. Thus, you can find the optimal saturated design in 10 points, replicate it five times, and use the resulting design as an initial design, as follows:

```
proc optex data=a seed=33805;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=saturated method=detmax;
   output out=b;
run;

data c;
   set b;
   drop i;
   do i=1 to 5; output; end;
run;

proc optex data=a seed=33805;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=detmax initdesign=c;
run;
```

The results are displayed in Output 15.3.2 and Output 15.3.3. The resulting design is 99.9% D-efficient and 98.4% A-efficient relative to the best design found by the straightforward approach (Output 15.2.1), and it takes considerably less time to produce.

Output 15.3.2 Efficiencies for the Unreplicated Saturated Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	41.6990	24.8480	67.6907	0.9508
2	41.4931	22.2840	70.8532	0.9841
3	40.9248	20.7672	62.2177	1.0247
4	40.7447	21.6253	52.7537	1.0503
5	39.9563	20.1557	46.4244	1.0868
6	39.9287	19.5856	45.9023	1.0841
7	39.9287	19.5856	45.9023	1.0841
8	38.9078	13.5976	37.7964	1.2559
9	38.9078	13.5976	37.7964	1.2559
10	37.6832	12.5540	45.3315	1.3036

Output 15.3.3 Initializing with a Data Set

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4388	24.4951	96.0717	0.4242

Example 15.4: Optimal Design Using an Augmented Best Design

NOTE: See *Engine Mapping Problem* in the SAS/QC Sample Library.

This example is a continuation of Example 15.2.

You can specify a set of points that you want to be included in the final design that the OPTEX procedure finds by using the AUGMENT= option in the GENERATE statement to specify a data set that contains a design to be augmented.

In this case, you can try to speed up the search for a 50-run design by first finding an optimal 25-run design and then augmenting that design with another 25 runs, as shown in the following statements:

```
proc optex data=a seed=36926;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=25 method=detmax;
   output out=b;
run;

proc optex data=a seed=37034;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=detmax augment=b;
run;
```

The result (see Output 15.4.1 and Output 15.4.2) is a design with almost 100% D-efficiency and A-efficiency relative to the best design found by the first attempt. However, this approach is not much faster than the original approach because the run time for the DETMAX algorithm is essentially linear in the size of the design (see the section "Memory and Run-Time Considerations" on page 1036).

Output 15.4.1 Efficiencies for the 25-Point Design to Be Augmented

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.2975	26.0374	91.1822	0.5849
2	46.2171	25.9733	86.4608	0.5859
3	46.1720	25.9378	88.3293	0.5860
4	46.1374	25.9128	86.1895	0.5866
5	46.0808	22.6647	86.1502	0.6169
6	46.0620	24.7326	89.7179	0.6012
7	45.9992	25.4549	90.3330	0.5946
8	45.9630	24.7610	88.2701	0.5991
9	45.9627	25.5310	88.5737	0.5894
10	45.7994	24.5645	87.7544	0.6005

Output 15.4.2 Efficiencies for the Augmented 50-Point Design

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.4957	25.0858	94.8160	0.4195
2	46.4773	25.0696	95.0646	0.4195
3	46.4684	24.5519	96.1259	0.4234
4	46.4676	24.5002	95.6830	0.4238
5	46.4587	25.0709	94.6650	0.4196
6	46.4555	24.8087	95.7768	0.4209
7	46.4471	24.5460	95.0073	0.4240
8	46.4373	25.0740	94.4640	0.4194
9	46.3899	25.0007	95.2162	0.4201
10	46.3662	24.4013	94.9539	0.4242

Example 15.5: Optimal Design Using a Small Candidate Set

NOTE: See *Engine Mapping Problem* in the SAS/QC Sample Library.

This example is a continuation of Example 15.4.

A well-chosen initial design can speed up the search procedure, as illustrated in Example 15.2. Another way to speed up the search is to reduce the candidate set. The following statements generate the optimal design

with a fast, sequential search and then use the FREQ procedure to examine the frequency of different factor levels in the final design:

```
proc optex data=a seed=33805 noprint;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=sequential;
   output out=b;
run;
proc freq;
   table AFR EGR SA / nocum;
run;
```

Output 15.5.1 Factor-Level Frequencies for Sequential Design

The FREQ Procedure

AFR	Frequency	Percent
15	19	38.00
16	6	12.00
17	6	12.00
18	19	38.00

EGR	Frequency	Percent
0.02	20	40.00
0.566	9	18.00
1.117	21	42.00

SA	Frequency	Percent
10	19	38.00
28	6	12.00
34	5	10.00
52	20	40.00

From Output 15.5.1, it is evident that most of the factor values lie in the middle or at the extremes of their respective ranges. This suggests looking for an optimal design by using a candidate set that includes only those points in which the factors have values in the middle or at the extremes of their respective ranges. The following statements illustrate this approach (see Output 15.5.2):

```
proc plan;
   factors AFR=4 ordered EGR=4 ordered SA=4 ordered
           / noprint;
   output out=a AFR nvals=(15, 16, 17, 18)
                EGR nvals=(0.020, 0.377, 0.566, 1.117)
                SA  nvals=(10, 28, 34, 52);
run;
proc optex seed=61552;
   model AFR|EGR|SA@2 AFR*AFR EGR*EGR SA*SA;
   generate n=50 method=detmax;
run;
```

Output 15.5.2 Optimal Design Using a Smaller Candidate Set

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	46.5151	24.9003	96.7226	0.4442
2	46.4997	24.5549	96.1157	0.4478
3	46.4920	24.5530	95.9941	0.4480
4	46.4657	24.8653	95.5627	0.4446
5	46.4547	24.5071	96.0385	0.4481
6	46.4333	25.0321	95.1371	0.4448
7	46.4333	25.0321	95.1371	0.4448
8	46.4333	25.0321	95.1371	0.4448
9	46.3916	24.3617	95.0041	0.4489
10	46.3379	24.8695	94.3115	0.4458

The resulting design is about as good as the best one obtained from a complete candidate set ($> 99.9\%$ relative D-efficiency and marginally higher relative A-efficiency) and takes much less time to find.

For another example of reducing the candidate set for the optimal design search, see the section "Handling Many Variables" on page 1008.

Example 15.6: Bayesian Optimal Design

NOTE: See *Bayesian Optimal Design* in the SAS/QC Sample Library.

Suppose you want a design in 20 runs for seven two-level factors. There are 29 terms in a full second-order model, so you will not be able to estimate all main effects and two-factor interactions. If the number of runs were a power of 2, a design of resolution 4 could be used to estimate all main effects free of the two-factor interactions, as well as to provide partial information on the interactions. However, when the number of runs is not a power of two, as in this case, DuMouchel and Jones (1994) suggest searching for a *Bayesian optimal design* by specifying nonzero prior precision values for the interactions. You can specify these values in the OPTEX procedure with the PRIOR= option in the MODEL statement. This option says that you want to consider all main effects and interactions as potential effects but you are willing to sacrifice information on the interactions to obtain maximal information on the main effects. When an orthogonal design of resolution 4 exists, it is optimal according to this Bayesian criterion. You can use the following statements to generate the Bayesian D-optimal design:

```
proc factex;
   factors x1-x7;
   output out=Candidates;
run;

proc optex data=Candidates seed=57922 coding=orth;
   model x1-x7,
         x1|x2|x3|x4|x5|x6|x7@2 / prior=0,16;
   generate n=20 method=m_fedorov;
   output out=Design;
run;
```

With orthogonal coding, the value of the prior for an effect indicates approximately how many prior "observations' worth" of information you have for that effect. In this case, the PRIOR= precision values and the use of commas to group effects in the MODEL statement indicate that there is no prior information for the main effects and 16 runs' worth of information for each two-factor interaction. For more information about orthogonal coding, see the section "Design Coding" on page 1032.

The efficiencies are shown in Output 15.6.1.

Output 15.6.1 Efficiencies for Bayesian Optimal Designs

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	85.1815	74.6705	85.2579	1.1476
2	85.1815	74.6705	85.2579	1.1476
3	85.1815	74.6705	85.2579	1.1476
4	85.0424	73.3109	81.0800	1.1582
5	85.0424	73.3109	81.0800	1.1582
6	84.5680	73.5053	84.1376	1.1566
7	84.4931	72.1671	81.7855	1.1673
8	84.4239	72.4979	81.7431	1.1646
9	84.3919	74.6097	89.3631	1.1480
10	84.3919	74.6097	89.3631	1.1480

Notice that the best design was found in 3 tries out of 10. It might be a good idea to repeat the search with more tries (see the ITER= option). You can use the ALIASING option of the GLM procedure to list the aliasing structure for the design:

```
data Design; set Design;
   y = ranuni(654231);
proc glm data=Design;
   model y = x1-x7 x1|x2|x3|x4|x5|x6|x7@2 / e aliasing;
run;
```

The relevant part of the output is shown in Output 15.6.2. Most of the main effects are indeed unconfounded with two-factor interactions, although many two-factor interactions are confounded with each other.

Output 15.6.2 Aliasing Structure for Bayesian Optimal Design

The GLM Procedure

General Form of Aliasing Structure
Intercept
x1 - 0.5*x3*x7
x2
x3
x4 + 0.5*x3*x7
x5
x6
x7
x1*x2 - x3*x6 + 0.5*x3*x7 - x4*x7
x1*x3 - x2*x6 - x5*x7
x2*x3 + x3*x7
x1*x4 - x5*x6 + x5*x7 + x6*x7
x2*x4 - x3*x6 + 0.5*x3*x7 - x4*x7
x3*x4 - x2*x6 - x5*x7
x1*x5 - x4*x6 - x3*x7
x2*x5 + x2*x6 + x5*x7 + x6*x7
x3*x5 + x3*x6 - x3*x7
x4*x5 - x1*x6 - x3*x7
x1*x7 - x4*x7
x2*x7 + x5*x7 + x6*x7

Example 15.7: Balanced Incomplete Block Design

NOTE: See *Balanced Incomplete Block Design* in the SAS/QC Sample Library.

This example uses the BLOCKS statement to construct a balanced incomplete block design (BIBD). An incomplete block design is a design for v (qualitative) treatments in b blocks of k runs each, where $k < v$ so that not all treatments can occur in each block. An incomplete block design is said to be *balanced* when all pairs of treatments occur equally often in the same block. A balanced design is always optimal for any criterion based on the information matrix, although there are many values of (v, b, k) for which no balanced design exists.

One way to construct an incomplete block design with the OPTEX procedure is to include the blocking factor in the candidate set and in the model. For example, the following statements search for a BIBD for seven treatments in seven blocks of size three—that is, $(v, b, k) = (7, 7, 3)$—using the full set of 49 treatment-by-block combinations for candidates:

```
data Candidates;
   do Treatment = 1 to 7;
      do Block = 1 to 7;
         output;
      end;
   end;
run;
```

```
proc optex data=Candidates seed=8327 coding=orth;
   class Treatment Block;
   model Treatment Block;
   generate n=21;
run;
```

By default, the OPTEX procedure performs the search 10 times from different random starting designs. The various efficiencies for each design are listed in Output 15.7.1.

Output 15.7.1 Efficiency Factors for $v = b = 7$, $k = 3$ Designs

The OPTEX Procedure

Design Number	D-Efficiency	A-Efficiency	G-Efficiency	Average Prediction Standard Error
1	89.0483	79.1304	82.7170	0.8845
2	89.0483	79.1304	82.7170	0.8845
3	88.4669	76.9882	78.6796	0.8967
4	88.4669	76.9882	78.6796	0.8967
5	88.4669	76.9882	78.6796	0.8967
6	88.4669	76.9882	78.6796	0.8967
7	88.4669	76.9882	78.6796	0.8967
8	88.4669	76.9882	78.6796	0.8967
9	88.1870	76.0262	78.7612	0.9024
10	87.7681	74.2459	73.9544	0.9131

Because the efficiency factors compare the designs to a (hypothetical) orthogonal design, values of 100% are not possible in this case. The OPTEX procedure includes facilities for examining the information matrix for the design; you can use these to verify that the best design found here is, in fact, balanced.

Searching for an optimal design for both treatments and blocks simultaneously has its limitations. Note that the balanced design was found on only two of the 10 tries. A more serious limitation is that this approach sometimes fails to find a design that has equal-sized blocks. A more efficient and flexible way to construct a block design with the OPTEX procedure is to use the BLOCKS statement.

The following statements use the BLOCKS statement to solve the preceding incomplete block design problem. In this case, the candidate set simply consists of the seven treatment levels.

```
data Candidates;
   do Treatment = 1 to 7;
      output;
   end;
run;

proc optex data=Candidates seed=73462 coding=orth;
   class Treatment;
   model Treatment;
   blocks structure=(7)3;
run;
```

The output again consists of efficiency factors for 10 different tries, but this time the factors are computed from the information matrix for only the treatment effects. In this special case (a single classification effect in the treatment model together with the STRUCTURE= option in the BLOCKS statement), the efficiency of each design as an incomplete block design is also listed (Output 15.7.2).

Output 15.7.2 Efficiency Factors for $v = b = 7$, k = 3 Optimal Blocking Designs

The OPTEX Procedure

Design Number	Treatment D-Efficiency	Treatment A-Efficiency	Block Design D-Efficiency
1	77.7778	77.7778	100.0000
2	77.7778	77.7778	100.0000
3	77.7778	77.7778	100.0000
4	77.7778	77.7778	100.0000
5	77.7778	77.7778	100.0000
6	77.7778	77.7778	100.0000
7	77.7778	77.7778	100.0000
8	77.7778	77.7778	100.0000
9	77.7778	77.7778	100.0000
10	77.7778	77.7778	100.0000

The 100% efficiency in the fourth column of the output shows that the balanced design was found on all 10 tries.

Because the OPTEX procedure is interactive, you can save the final design in a data set by submitting the OUTPUT statement immediately after the preceding statements. The following statements use the BLOCKNAME= option to rename the block variable:

```
   output out=BIBD blockname=Block;
proc print data=BIBD;
run;
```

The final design is shown in Output 15.7.3.

Output 15.7.3 Balanced Incomplete Block Design for $v = b = 7$, $k = 3$

Obs	BLOCK	Treatment
1	1	1
2	1	4
3	1	7
4	2	6
5	2	3
6	2	1
7	3	2
8	3	5
9	3	1
10	4	6
11	4	2
12	4	7
13	5	5
14	5	4
15	5	6
16	6	5
17	6	7
18	6	3
19	7	4
20	7	3
21	7	2

Although there is no guarantee that the OPTEX procedure will find the globally optimal block design by this method, it usually does find small to medium-sized balanced designs, and it always finds a very efficient design. For example, for the designs given in Table 9.5 of Cochran and Cox (1957), the OPTEX procedure consistently finds the theoretically optimal BIBD in all cases with 10 or fewer treatments. Furthermore, in no case is the D-efficiency relative to the balanced design less than 99%.

Example 15.8: Optimal Design with Fixed Covariates

NOTE: See *Optimal Design with Fixed Covariates* in the SAS/QC Sample Library.

In addition to finding optimal block designs, you can use the BLOCKS statement to find designs that are optimal with respect to more general covariate models. You can use the DESIGN= option in the BLOCKS statement to specify the data set that contains the covariates. Covariate models are specified in the same way as the treatment model.

This example is based on an example in Harville (1974). Suppose you want a design for five qualitative treatments in 10 runs. The value of a covariate that is thought to be related to the response has been recorded for each of the experimental units. For example, if the treatments are different types of animal feed, a typical covariate might be the initial weight of each animal. The following statements create the data sets Cov and Treatment, which contain the covariate values and the candidate treatment levels, respectively. Then the OPTEX procedure is invoked with a simple one-way model for the treatment effect and a quadratic model for the covariate effect.

```
data Cov;
   input u @@;
   datalines;
0.46 0.54 0.58 0.60 0.73 0.77 0.82 0.84 0.89 0.95
;
data Treatment;
   do t = 1 to 5; output; end;
run;
proc optex data=Treatment seed=17364 coding=orthcan;
   class t;
   model t;
   blocks design=Cov;
   model u u*u;
   output out=Design;
run;

proc print data=Design;
run;
```

In this case, the CODING=ORTHCAN option in the PROC OPTEX statement has the same effect as CODING=ORTH, which is to produce orthogonal coding with respect to the candidates. Note the following:

- The CLASS and MODEL statements that define the treatment model precede the BLOCKS statement.
- The MODEL statement that defines the covariate model follows the BLOCKS statement.

As a general rule, CLASS and MODEL statements that come before a BLOCKS statement are interpreted as applying to the treatment model, whereas CLASS and MODEL statements that come after a BLOCKS statement that involves the DESIGN= blocks specification are interpreted as applying to the covariate model.

Output 15.8.1 shows the listing of the efficiency values for the 10 designs that are found. Note that the efficiencies are the same for all tries. A listing of the design is shown in Output 15.8.2.

Output 15.8.1 Optimal Treatment Efficiency Factors with a Quadratic Covariate Effect

The OPTEX Procedure

Design Number	Treatment D-Efficiency	Treatment A-Efficiency
1	91.6621	91.1336
2	91.6621	91.1336
3	91.6621	91.1336
4	91.6621	91.1336
5	91.6621	91.1336
6	91.6621	91.1336
7	91.6621	91.1336
8	91.6621	91.1336
9	91.6621	91.1336
10	91.6621	91.1336

Output 15.8.2 Optimal Design with a Quadratic Covariate Effect

Obs	u	t
1	0.46	4
2	0.54	3
3	0.58	1
4	0.60	2
5	0.73	5
6	0.77	4
7	0.82	3
8	0.84	1
9	0.89	2
10	0.95	5

When you use the BLOCKS statement without specifying the GENERATE statement, the full candidate set is used as the treatment set for optimal blocking. If you specify both statements, an optimal design for the treatments that ignores the blocks is first generated, and the result is used as the treatment set for optimal blocking. This enables several options to be combined to evaluate existing designs. For example, the following statements evaluate the optimal design given in Harville (1974) for the preceding situation:

```
data Harville;
   input t @@;
   datalines;
1   2   3   4   5   1   2   3   4   5
;
proc optex data=Treatment coding=orthcan;
   class t;
   model t;
   generate initdesign=Harville method=sequential;
   blocks design=Cov init=chain iter=0;
   model u u*u;
run;
```

The efficiency values for Harville's design are shown in Output 15.8.3. They are the same as for the design found by the OPTEX procedure.

Output 15.8.3 Treatment Efficiency Factors for Harville's Design

The OPTEX Procedure

Design Number	Treatment D-Efficiency	Treatment A-Efficiency
1	91.6621	91.1336

In fact, the optimal design found by OPTEX can be derived from Harville's design simply by relabeling treatments. In order of increasing U, both designs consist of two consecutive replicates of the treatments, with treatments in both replicates occurring in the same order.

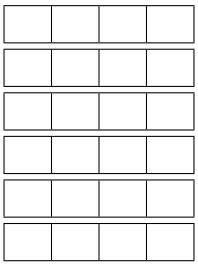
Example 15.9: Optimal Design in the Presence of Covariance

NOTE: See *Optimal Design in Presence of Covariance* in the SAS/QC Sample Library.

The BLOCKS statement finds a design that maximizes the determinant $|\mathbf{X}'\mathbf{A}\mathbf{X}|$ of the treatment information matrix, where $\mathbf{A}$ depends on the block or covariate model. Alternatively, you can directly specify the matrix $\mathbf{A}$ to find the D-optimal design when $\mathbf{A}$ is the variance-covariance matrix for the runs. You can specify the data set containing the covariance matrix with the COVAR= option in the BLOCKS statement, listing the variables that correspond to the columns of the covariance matrix in the VAR= option. If you specify n variables in the VAR= option, the values of these variables in the first n observations in the data set will be used to define $\mathbf{A}$.

For example, suppose you want to compare the effects of seven different fertilizers on crop yield, by using seven long, narrow blocks of four plots each, as depicted in Figure 15.8.

Figure 15.8 Block Structure for Neighbor Balance



In this case, it is reasonable to conjecture that closer plots within each block are more correlated. In particular, suppose that the plots are *autocorrelated*, so that the correlation matrix for the four plots in each block is of the form

$$\mathbf{R} = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

where $-1 \leq \rho \leq 1$. If there is also an overall fixed effect due to blocks, the information matrix for the effect of fertilizer has the form $\mathbf{X}'\mathbf{A}\mathbf{X}$, where

$$\mathbf{A} = \left(\mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{Z}\left(\mathbf{Z}'\mathbf{V}^{-1}\mathbf{Z}\right)^{-1}\mathbf{Z}'\mathbf{V}^{-1}\right)^{-}$$

In this formula, $\mathbf{V}$ is the block diagonal matrix of the plot-by-plot correlation structure, with seven copies of R_4 on the diagonal. The matrix $\mathbf{Z}$ is the design matrix that corresponds to the block effect. The optimal design should take into account this neighbor covariance structure in addition to the block structure.

The following code uses the SAS/IML matrix language to construct $\mathbf{A}$ by using $\rho = 0.1$ and saves it in a data set named a:

```
proc iml;
   Blocks = int(((1:28)`-1)/4) + 1;
   z = j(28,1) || designf(Blocks);
   r = toeplitz(0.1**(0:3));
   v = r;
   do i = 2 to 7; v = block(v,r); end;
   iv = inv(v);
   a = ginv(iv-iv*z*inv(z`*iv*z)*z`*iv);
   create A from a;
   append from a;
quit;
```

The data set is created with variables named COL1, COL2 . . . , COL28, by default.

To find an allocation of fertilizers to plots that is optimal for detecting the fertilizer effect in the presence of this autocorrelation, simply specify a one-way model for the treatment effects and use the COVAR= option in the BLOCKS statement to specify the data set A as the covariance matrix for the runs, as follows:

```
data Fertilizer;
   do f = 1 to 7; output; end;
run;
proc optex data=Fertilizer seed=56672 coding=orth;
   class f;
   model f;
   blocks covar=A var=(COL1-COL28);
   output out=NBD;
run;
```

The SAS/IML matrix language also provides a convenient way of listing the design:

```
proc iml;
   use NBD;
   read all var {f};
   NBD = shape(f,7,4);
   print NBD [format=2.];
```

These PROC IML statements read in the selected levels of fertilizer and the reshape them into seven 4-run blocks before printing them. The resulting design is shown in Output 15.9.1. Note that it is not only a balanced incomplete block design, but it is also balanced for first neighbors—that is, every pair of treatments occurS equally often on horizontally adjacent plots.

Output 15.9.1 Neighbor-Balanced BIBD for $v = b = 7$, $k = 4$, Found by Optimal Blocking

NBD
7 2 1 5
6 1 7 3
4 7 6 2
1 4 6 5
6 3 5 2
1 3 2 4
7 5 4 3

Example 15.10: Adding Space-Filling Points to a Design

NOTE: See *Adding Space-filling Points to a Design* in the SAS/QC Sample Library.

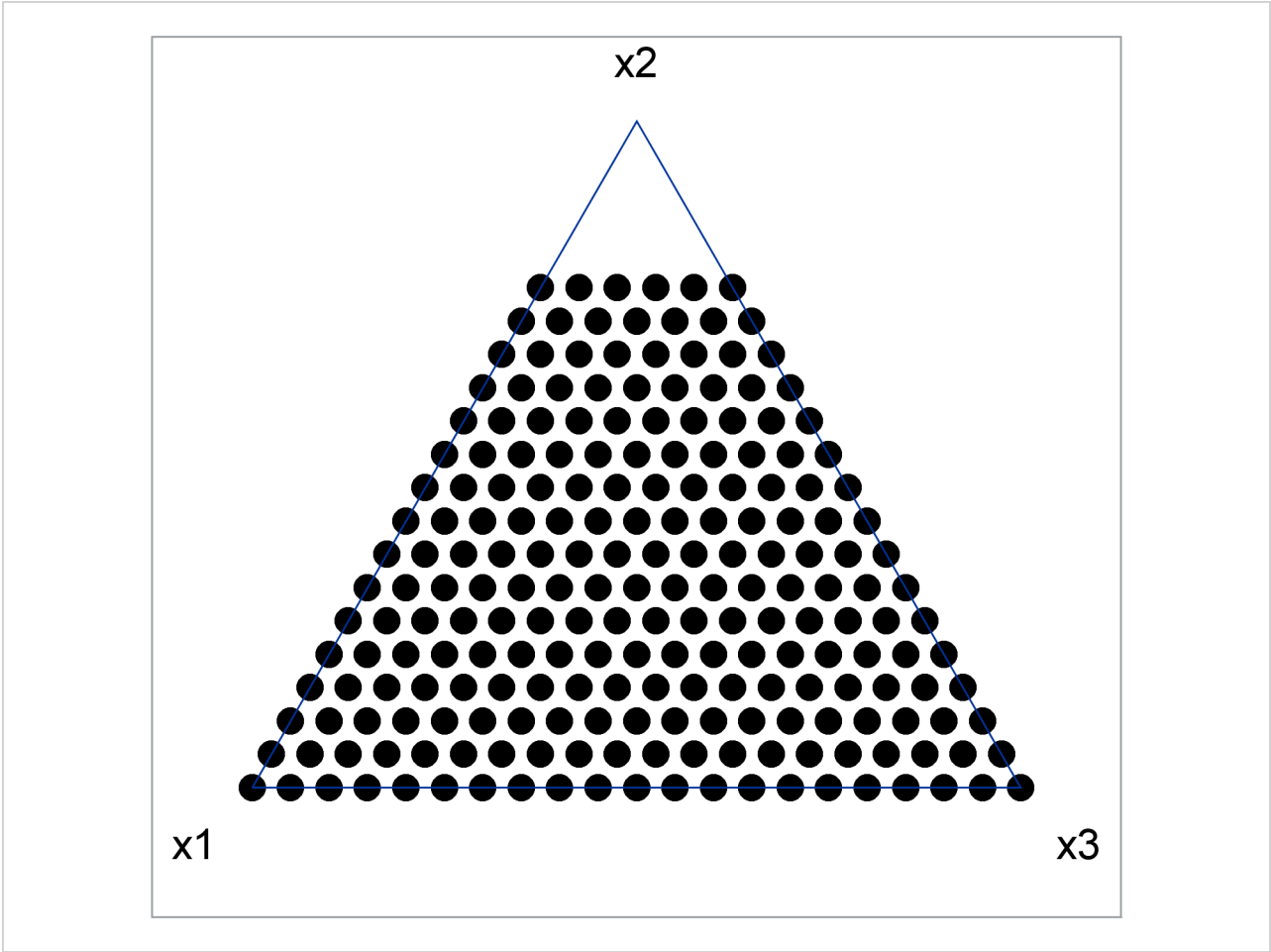
Suppose you want a 15-run experiment for three mixture factors x1, x2, and x3; furthermore, suppose that x3 cannot account for any more than 75% of the mixture. The vertices and generalized edge centroids of the region defined by these constraints make up a good candidate set to use with the OPTEX procedure for finding a D-optimal design for such an experiment. However, information-based criteria such as D- and A-efficiency tend to push the design to the edges of the candidate space, leaving large portions of the interior relatively uncovered. For this reason, it is often a good idea to augment a D-optimal design with some points that are chosen according to U-optimality, which seeks to cover the candidate region as well as possible.

The following statements create a candidate data set that contains 216 points in the region that is defined by the constraints $x_1 + x_2 + x_3 = 1$ and $x_3 \leq 0.75$ on the factors:

```
data a;
   do x1 = 0 to 100 by 5;
      do x3 = 0 to 100 by 5;
         x2 = 100 - x1 - x3;
         if (0<= x2 <= 75) then output;
      end;
   end;
run;
data a; set a;
   x1 = x1 / 100;
   x2 = x2 / 100;
   x3 = x3 / 100;
run;
```

The constraint that the factor levels sum to 1 means that the candidate points all lie on a plane. Thus, the values of all three variables can be displayed in a two-dimensional "mixture plot," as shown in Output 15.10.1.

Output 15.10.1 Points in the Feasible Region for Constrained Mixture Design

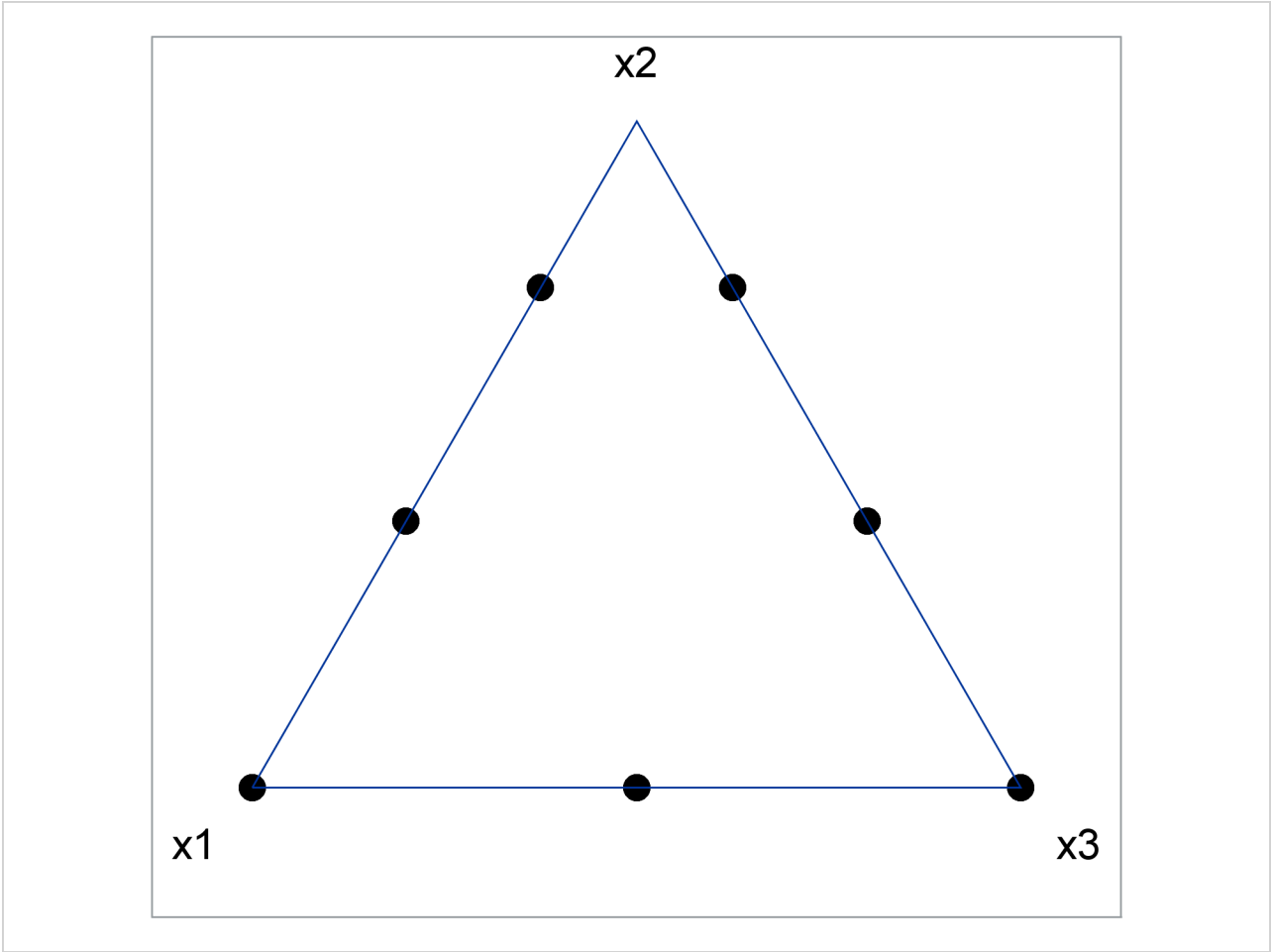


You can use the OPTEX procedure to select 10 points from the mentioned candidate points optimal for estimating a second-order model in the mixture factors:

```
proc optex data=a seed=60868 nocode;
   model x1|x2|x3@2 / noint;
   generate n=10;
   output out=b;
run;
```

The resulting points are plotted in Output 15.10.2. There are only seven unique points, indicating that the D-optimal design replicates some chosen candidate points.

Output 15.10.2 D-Optimal Constrained Mixture Design

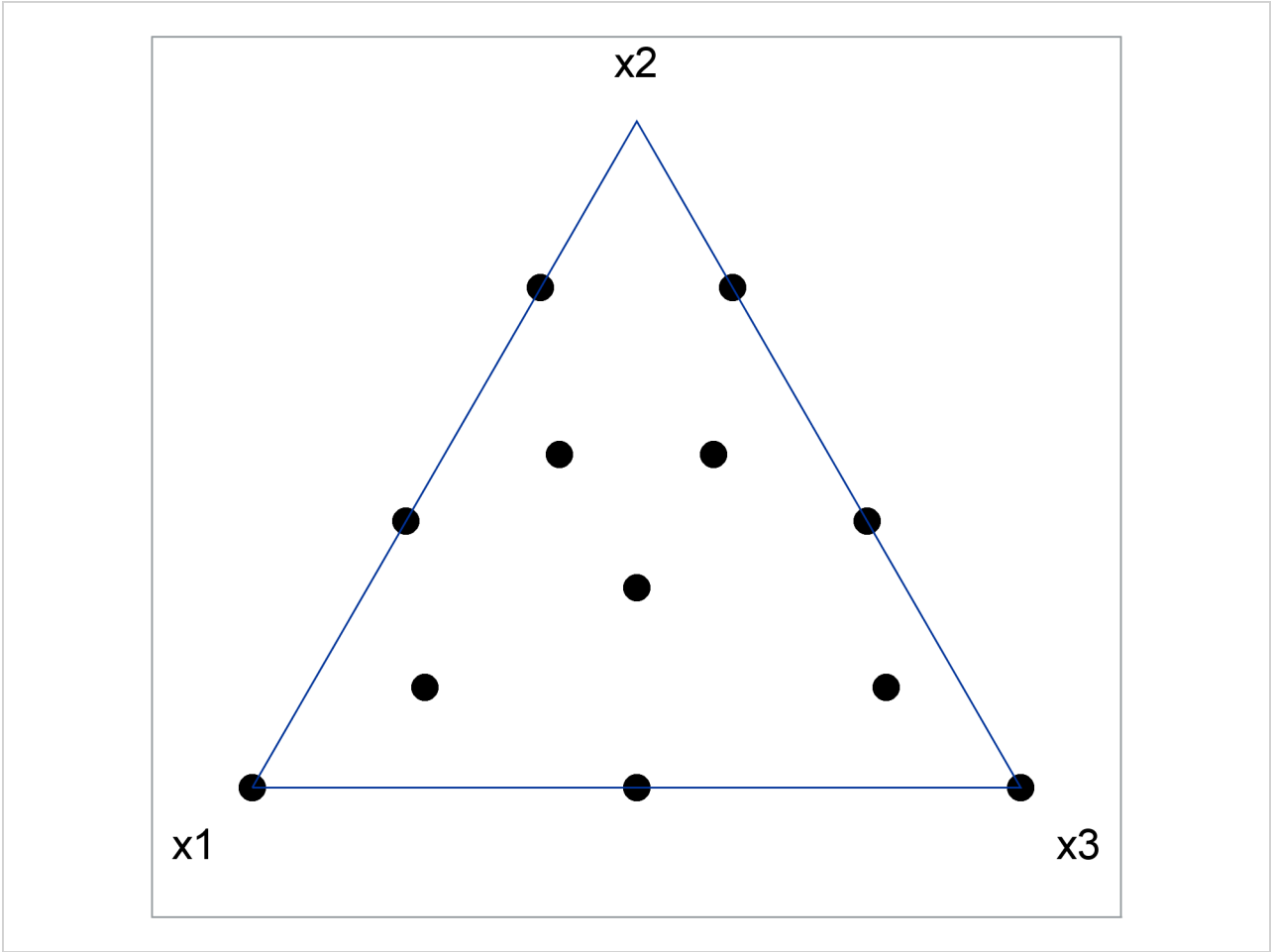


The D-optimal design leaves a large “hole” in the feasible region. The following statements “fill in the hole” in the optimal design that is saved in B by augmenting it with points chosen from the candidate data set a to optimize the U-criterion:

```
proc optex data=a seed=4321 nocode;
   model x1 x2 x3 / noint;
   generate n=15 augment=b criterion=u;
   output out=c;
run;
```

The resulting points are shown in Output 15.10.3. The U-optimal design fills in the candidate region in much the same way that you might construct the design by visually assigning points. That is, the general approach that uses the OPTEX procedure agrees with visual intuition for this small problem. This indicates that the general approach will yield an appropriate design for higher-dimensional problems that cannot be visualized.

Output 15.10.3 D-optimal Constrained Mixture Design Filled in U-optimally



References

Cochran, W. G., and Cox, G. M. (1957). *Experimental Designs*. 2nd ed. New York: John Wiley & Sons.

Cook, R. D., and Nachtsheim, C. J. (1980). "A Comparison of Algorithms for Constructing Exact D-Optimal Designs." *Technometrics* 22:315–324.

Cook, R. D., and Nachtsheim, C. J. (1989). "Computer-Aided Blocking of Factorial and Response-Surface Designs." *Technometrics* 31:339–346.

DuMouchel, W., and Jones, B. (1994). "A Simple Bayesian Modification of D-Optimal Designs to Reduce Dependence on an Assumed Model." *Technometrics* 36:37–47.

Dykstra, O., Jr. (1971). "The Augmentation of Experimental Data to Maximize $|\mathbf{X}'\mathbf{X}|$." *Technometrics* 13:682–688.

Fedorov, V. V. (1972). *Theory of Optimal Experiments*. Translated and edited by W. J. Studden and E. M. Klimko. New York: Academic Press.

Galil, Z., and Kiefer, J. (1980). "Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-Optimum Designs." *Technometrics* 22:301–313.

Harville, D. A. (1974). "Nearly Optimal Allocation of Experimental Units Using Observed Covariate Values." *Technometrics* 16:589–599.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990). "Minimax and Maximin Distance Designs." *Journal of Statistical Planning and Inference* 26:131–148.

Johnson, M. E., and Nachtsheim, C. J. (1983). "Some Guidelines for Constructing Exact D-Optimal Designs on Convex Design Spaces." *Technometrics* 25:271–277.

Mitchell, T. J. (1974a). "An Algorithm for the Construction of D-Optimal Experimental Designs." *Technometrics* 16:203–210.

Mitchell, T. J. (1974b). "Computer Construction of 'D-Optimal' First-Order Designs." *Technometrics* 20:211–220.

Mitchell, T. J., and Miller, F. L., Jr. (1970). *Use of Design Repair to Construct Designs for Special Linear Models*. Mathematics Division Annual Progress Report ORNL-4661, Oak Ridge National Laboratory.

Nguyen, N.-K., and Miller, A. J. (1992). "A Review of Exchange Algorithms for Constructing Discrete D-Optimal Designs." *Computational Statistics and Data Analysis* 14:489–498.

Nguyen, N.-K., and Piepel, G. F. (2005). "Computer-Generated Experimental Designs for Irregular-Shaped Regions." *Quality Technology and Quantitative Management* 2:147–160.

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

Snee, R. D. (1985). "Computer-Aided Design of Experiments—Some Practical Experiences." *Journal of Quality Technology* 17:222–236.

Vance, L. C. (1986). *Computer Construction of Experimental Designs*. General Motors Research Report GMR-5411, General Motors Laboratories, Warren, MI.

Chapter 16
The PARETO Procedure

Contents

Overview: PARETO Procedure **1068**
Getting Started: PARETO Procedure **1069**
Creating a Pareto Chart from Raw Data 1069
Creating a Pareto Chart from Frequency Data 1073
Restricting the Number of Pareto Categories 1074
Displaying Summary Statistics on a Pareto Chart 1077
Syntax: PARETO Procedure **1078**
PROC PARETO Statement 1079
BY Statement 1080
HBAR Statement 1081
INSET Statement 1085
VBAR Statement 1090
Dictionary of HBAR and VBAR Statement Options 1095
Details: PARETO Procedure **1117**
Terminology 1117
Labels for Chart Features 1120
Scaling the Cumulative Percentage Curve 1120
Positioning Insets 1121
Creating Output Data Sets 1126
ODS Graphics 1127
Constructing Effective Pareto Charts 1128
Missing Values 1129
Role of Variable Formats 1129
Large Data Sets 1129
Examples: PARETO Procedure **1130**
Example 16.1: Creating Before-and-After Pareto Charts 1130
Example 16.2: Creating Two-Way Comparative Pareto Charts 1134
Example 16.3: Highlighting the “Vital Few” 1141
Example 16.4: Highlighting Combinations of Categories 1142
Example 16.5: Highlighting Combinations of Cells 1144
Example 16.6: Ordering Rows and Columns in a Comparative Pareto Chart 1147
Example 16.7: Merging Columns in a Comparative Pareto Chart 1150
Example 16.8: Creating Weighted Pareto Charts 1152
Example 16.9: Creating Alternative Pareto Charts 1154
Example 16.10: Customizing Inset Labels and Formatting Values 1157
Example 16.11: Specifying Inset Headers and Positions 1159

Example 16.12: Managing a Large Number of Categories 1160
References **1168**

Overview: PARETO Procedure

The PARETO procedure creates Pareto charts, which display the relative frequencies of quality-related problems in a process or operation. The frequencies are represented by bars that are ordered in decreasing magnitude. Thus, you can use a Pareto chart to decide which subset of problems you should solve first or which problem areas deserve the most attention.

Pareto charts provide a tool for visualizing the Pareto principle,[1] which states that a small subset of problems tend to occur much more frequently than the remaining problems. In Japanese industry, the Pareto chart is one of the "seven basic QC tools" that are heavily used by workers and engineers. Ishikawa (1976) discusses how to construct and interpret a Pareto chart. Examples of Pareto charts are also given by Kume (1985) and Wadsworth, Stephens, and Godfrey (1986).

You can use the PARETO procedure to do the following:

- construct Pareto charts from unsorted raw data (for example, a set of quality problems that have not been classified into categories) or from a set of distinct categories and corresponding frequencies
- construct Pareto charts that are based on the percentage of occurrence of each problem, the frequency (number of occurrences), or a weighted frequency (such as frequency that is weighted by the cost of each problem)
- add a curve that indicates the cumulative percentage across categories
- construct side-by-side Pareto charts or stacked Pareto charts
- construct *comparative Pareto charts*, which enable you to compare the Pareto frequencies across the levels of one or two classification variables. For example, you can compare the frequencies of problems that occur on three different machines for five consecutive days.
- highlight the "vital few" and the "useful many"[2] categories by using different colors for bars that correspond to the n most frequently occurring categories or the m least frequently occurring categories.
- restrict the number of categories that are displayed to the n most frequently occurring categories
- create charts whose bars are oriented vertically or horizontally
- highlight special categories by using different colors for specific bars
- display sample sizes and other statistics on Pareto charts
- label the bars with their frequency values

[1] Both the chart and the principle are named after Vilfredo Pareto (1848–1923), an Italian economist and sociologist. His first work, *Cours d'Économie Politique* (1895–1897), applied what is now termed the *Pareto distribution* to the study of income size.

[2] Juran originally referred to these categories as the "trivial many"; however, because all problems merit attention, the term "useful many" is preferred (Burr 1990).

- create charts as ODS Graphics output or as traditional graphics
- annotate traditional graphics charts
- save traditional graphics output in a graphics catalog for subsequent replay
- save information that is associated with the categories (such as the frequencies) in an output data set
- create variations on traditional Pareto charts, as described by Wilkinson (2006)

A Pareto chart has three axes, whose display depends on whether the Pareto chart is a traditional vertical Pareto or a horizontal bar chart. A horizontal bar chart that is produced by the PARETO procedure is essentially a vertical Pareto chart that is rotated 90 degrees clockwise. Table 16.1 shows how the three axes are displayed on the two types of Pareto charts.

Table 16.1 Pareto Chart Axes

Axis	Displayed on a Vertical Pareto Chart	Displayed on a Horizontal Pareto Chart
Category axis	Horizontally at the bottom of the chart	Vertically at the left side of the chart
Frequency axis	On the left (also called the primary vertical axis)	At the top of the chart (also called the primary horizontal axis)
Cumulative percentage axis	On the right (also called the secondary vertical axis)	At the bottom of the chart (also called the secondary horizontal axis)

Getting Started: PARETO Procedure

Creating a Pareto Chart from Raw Data

NOTE: See *Basic Pareto Chart from Raw Data* in the SAS/QC Sample Library.

In the fabrication of integrated circuits, common causes of failures include improper doping, corrosion, surface contamination, silicon defects, metallization, and oxide defects. The causes of 31 failures were recorded in a SAS data set called Failure1:

```
data Failure1;
   length Cause $ 16;
   label  Cause = 'Cause of Failure';
   input  Cause & $;
   datalines;
Corrosion
Oxide Defect
Contamination
Oxide Defect
Oxide Defect
Miscellaneous
Oxide Defect
```

```
Contamination
Metallization
Oxide Defect
Contamination
Contamination
Oxide Defect
Contamination
Contamination
Contamination
Corrosion
Silicon Defect
Miscellaneous
Contamination
Contamination
Contamination
Miscellaneous
Contamination
Contamination
Doping
Oxide Defect
Oxide Defect
Metallization
Contamination
Contamination
;
```

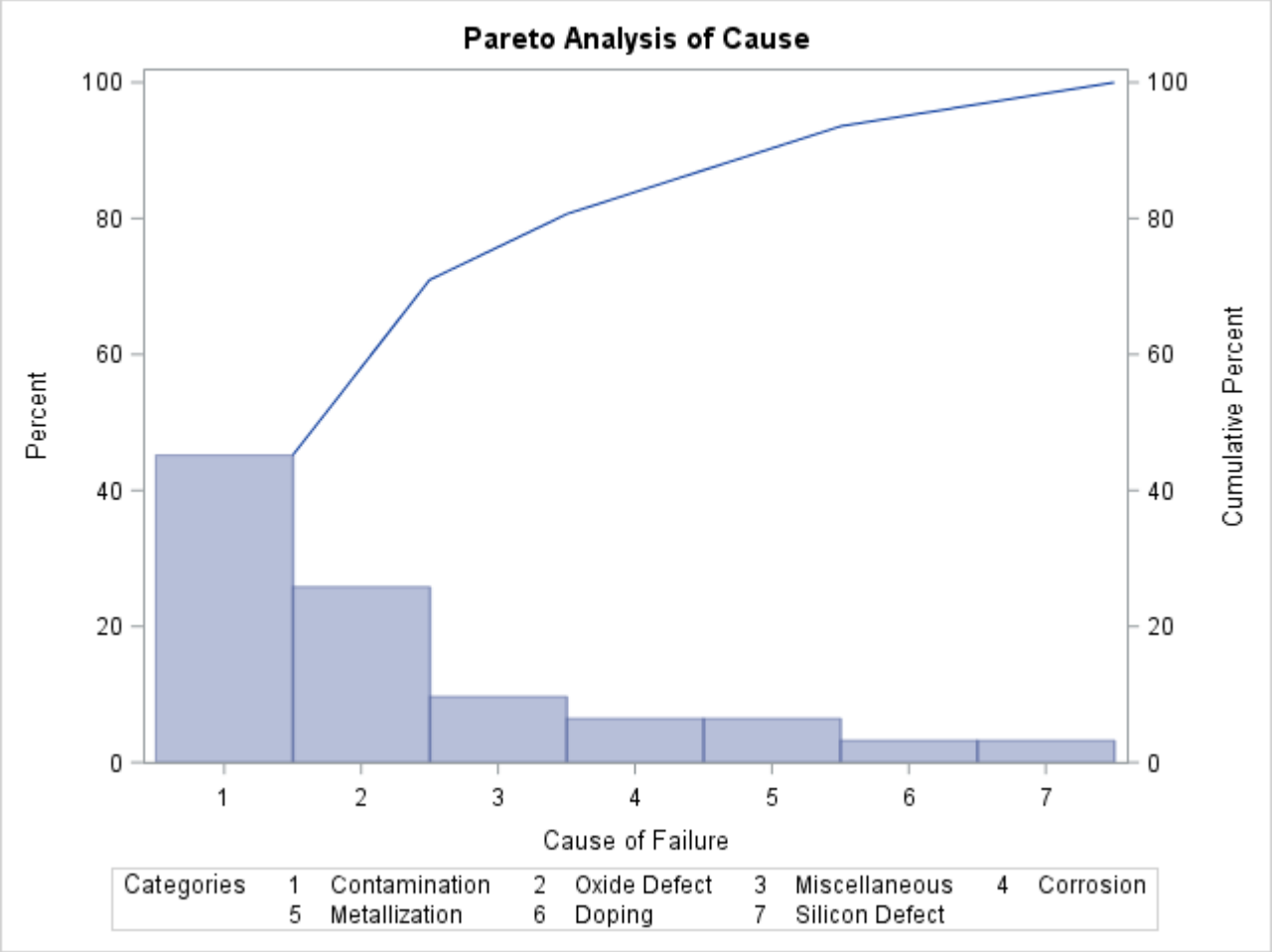
Each of the 31 observations corresponds to a different circuit, and the value of Cause provides the cause for the failure. These are raw data in the sense that more than one observation has the same value of Cause and that the observations are not sorted by Cause. The following statements produce a basic Pareto chart for the failures:

```
ods graphics on;
proc pareto data=Failure1;
   vbar Cause;
run;
```

The PROC PARETO statement (referred to as the PROC statement) invokes the PARETO procedure and identifies the input data set. You specify one or more process variables to be analyzed in the VBAR statement. The ODS GRAPHICS ON statement that is specified before the PROC statement enables ODS Graphics, so the Pareto chart is created using ODS Graphics instead of traditional graphics.

The Pareto chart is shown in Figure 16.1.

Figure 16.1 Pareto Chart for Integrated Circuit Failures in the Data Set Failure1



PROC PARETO has classified the values of Cause into seven distinct categories. The bars represent the percentage of failures in each category, and they are arranged in decreasing order. Thus, the most frequently occurring category is `'Contamination'`, which accounts for 45% of the failures. The Pareto curve indicates the cumulative percentage of failures from left to right; for example, `'Contamination'` and `'Oxide Defect'` together account for 71% of the failures.

If there is insufficient space to label the bars along the category axis, PROC PARETO numbers the bars from left to right and adds a legend to identify the categories, as in Figure 16.1. A category legend is likely to be introduced in the following cases:

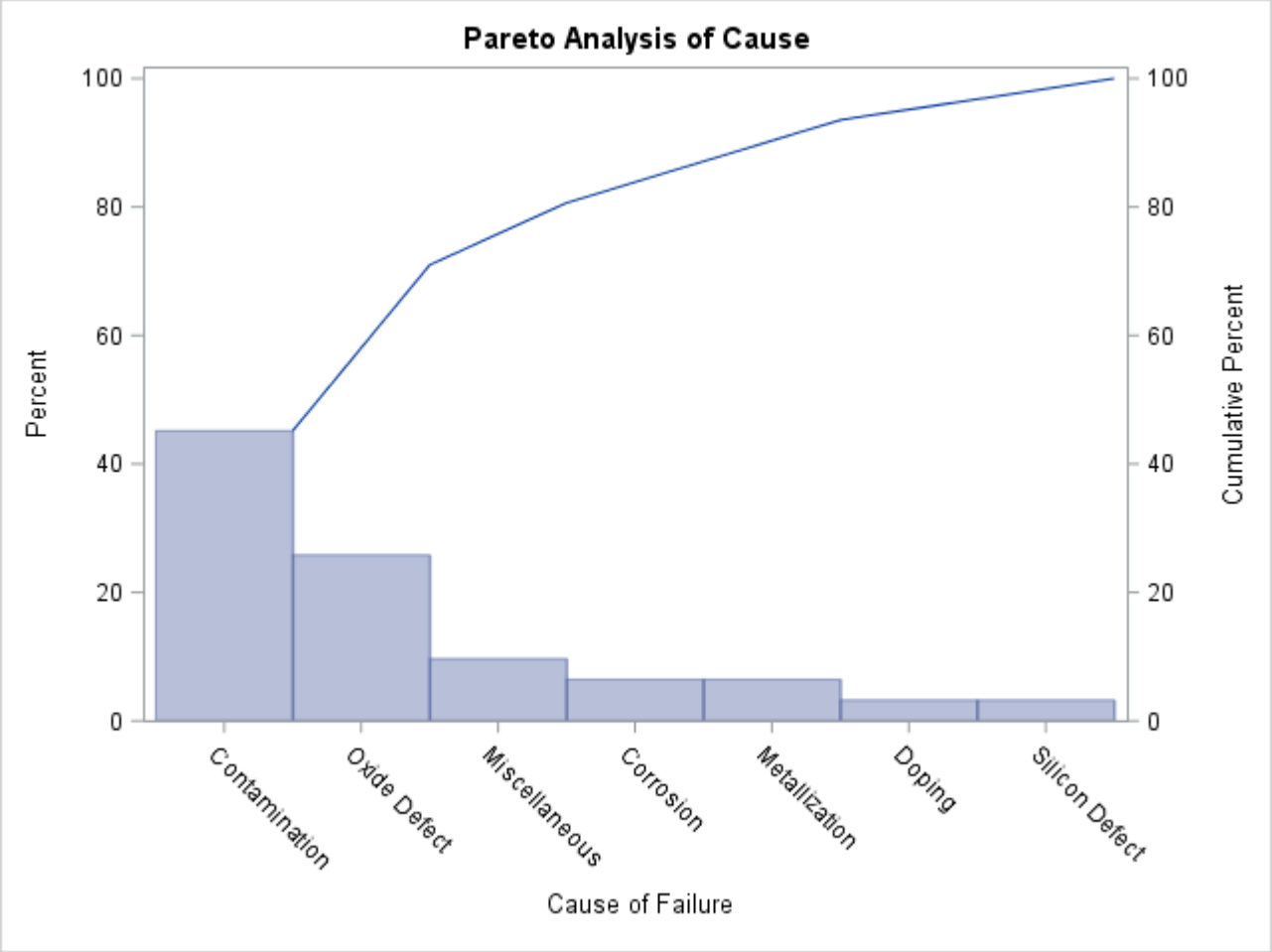
- The number of categories is large.
- The category labels are lengthy. Category labels can be up to 64 characters.
- You specify a large text height. In traditional graphics output, you can specify the text height in the HEIGHT= option in the HBAR or VBAR statement or in the HTEXT= option in a GOPTIONS statement.

The following statements suppress the category legend by specifying the CATLEGEND=OFF option:

```
proc pareto data=Failure1;
   vbar Cause / catlegend=off;
run;
```

A slash (/) is used to separate the process variable Cause from the options that are specified in the VBAR statement. The resulting chart is show in Figure 16.2.

Figure 16.2 Pareto Chart with Category Legend Suppressed



Because the category legend is turned off, PROC PARETO displays the category labels at an angle so that they do not collide.

Creating a Pareto Chart from Frequency Data

NOTE: See *Basic Pareto Chart from Frequency Data* in the SAS/QC Sample Library.

In some situations, a count (frequency) is available for each category, or you can compress a large data set by creating a frequency variable for the categories before applying the PARETO procedure.

For example, you can use the FREQ procedure to obtain the compressed data set Failure2 from the data set Failure1:

```
proc freq data=Failure1;
   tables Cause / noprint out=Failure2;
run;
```

A listing of Failure2 is shown in Figure 16.3.

Figure 16.3 Data Set Failure2, Which Is Created by Using PROC FREQ

Obs	Cause	COUNT	PERCENT
1	Contamination	14	45.1613
2	Corrosion	2	6.4516
3	Doping	1	3.2258
4	Metallization	2	6.4516
5	Miscellaneous	3	9.6774
6	Oxide Defect	8	25.8065
7	Silicon Defect	1	3.2258

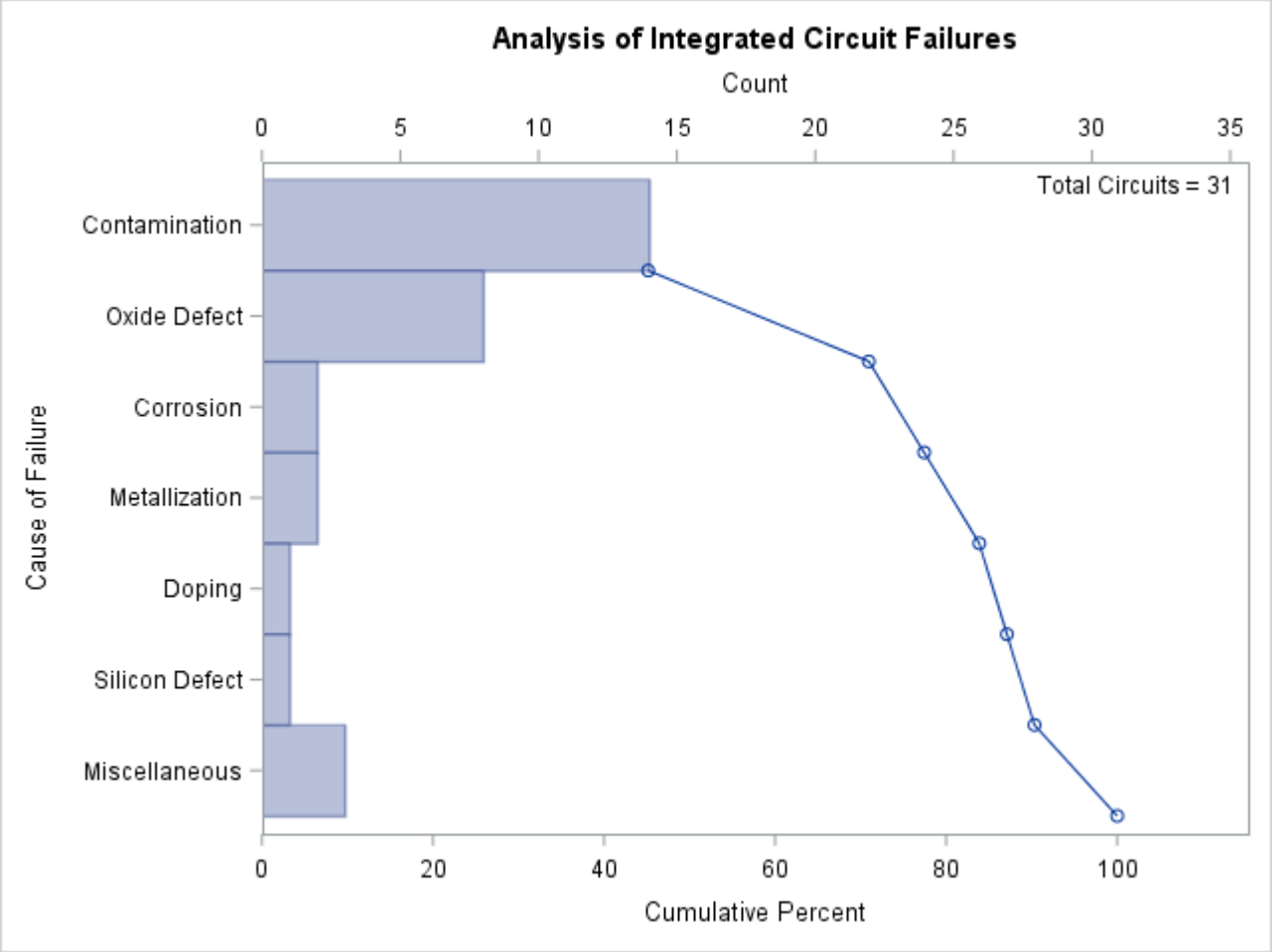
The following statements produce a horizontal Pareto chart for the data in Failure2:

```
title 'Analysis of Integrated Circuit Failures';
proc pareto data=Failure2;
   hbar Cause / freq     = Count
                scale    = count
                last     = 'Miscellaneous'
                nlegend  = 'Total Circuits'
                odstitle = title1
                markers;
run;
```

The frequency variable Count is specified in the FREQ= option. Specifying SCALE=COUNT requests a frequency scale for the frequency axis (at the top of the chart). Specifying LAST='Miscellaneous' causes the `'Miscellaneous'` category to be displayed last regardless of its frequency. The NLEGEND= option adds a sample size legend labeled "Total Circuits." Specifying ODSTITLE=TITLE replaces the default graph title with the title that is specified in the TITLE statement. The MARKERS option places markers at the points on the cumulative percentage curve.

The chart is displayed in Figure 16.4.

Figure 16.4 Pareto Chart with Frequency Scale



Note that in a horizontal Pareto chart categories are listed in decreasing frequency order from top to bottom on the category axis.

There are two sets of tied categories in this example: `'Corrosion'` and `'Metallization'` each occur twice, and `'Doping'` and `'Silicon Defect'` each occur once. PROC PARETO displays tied categories alphabetically in order of their formatted values. Thus, `'Corrosion'` appears before `'Metallization'`, and `'Doping'` appears before `'Silicon Defect'` in Figure 16.4. This is simply a convention, and no practical significance should be attached to the order in which tied categories are arranged.

Restricting the Number of Pareto Categories

NOTE: See *Pareto Chart with Restricted Number of Categories* in the SAS/QC Sample Library.

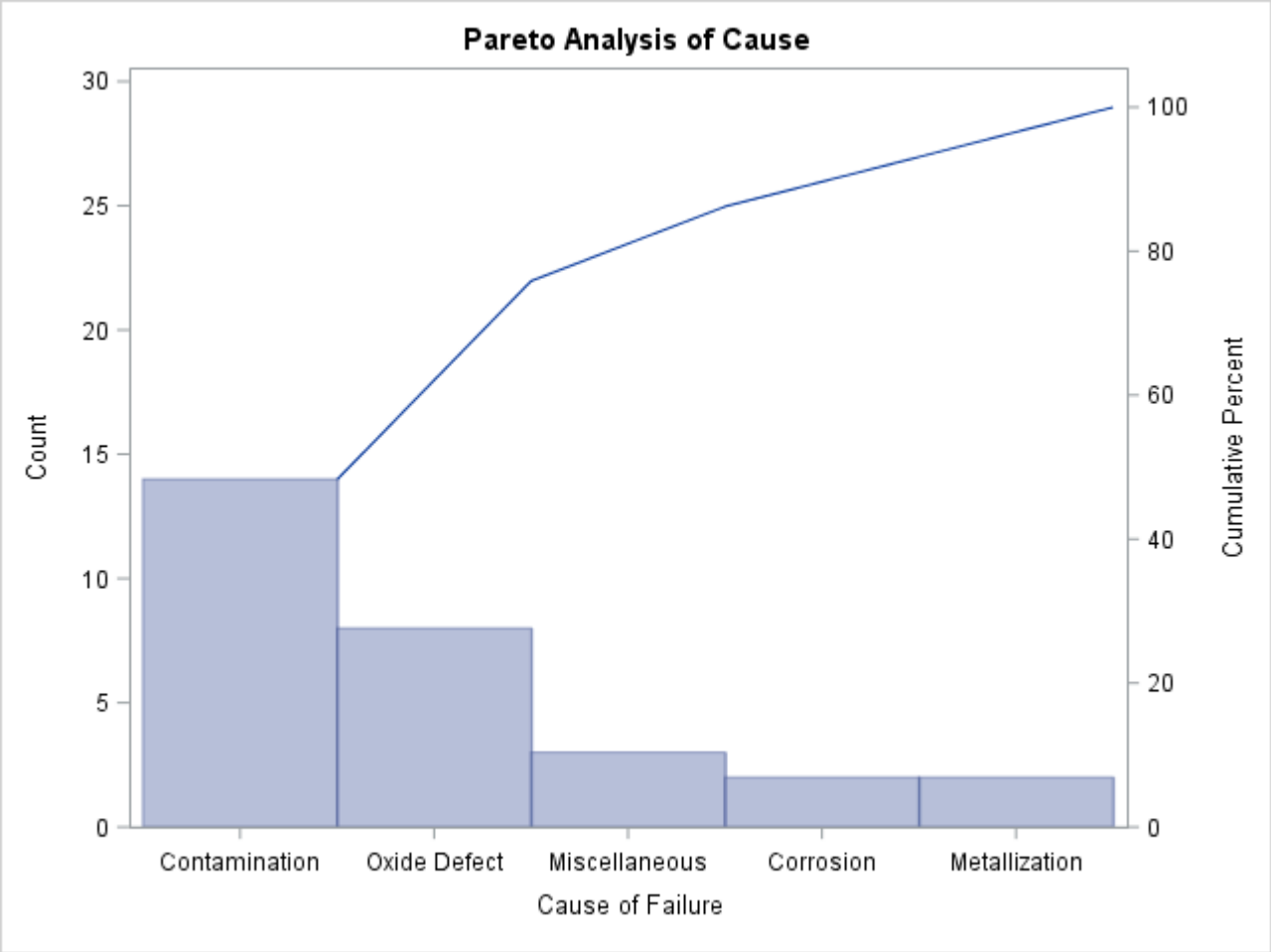
Unlike the previous examples, some applications involve too many categories to display on a chart. The solution presented here is to create a restricted Pareto chart that displays only the most frequently occurring categories.

The following statements create a Pareto chart for the five most frequently occurring levels of Cause in the data set Failure2 (which is listed in Figure 16.3):

```
proc pareto data=Failure2;
   vbar Cause / freq    = Count
                scale   = count
                maxncat = 5;
run;
```

The MAXNCAT= option specifies the number of categories to be displayed. The chart, shown in Figure 16.5, does not display the categories `'Doping'` and `'Silicon Defect'`.

Figure 16.5 Restricted Pareto Chart

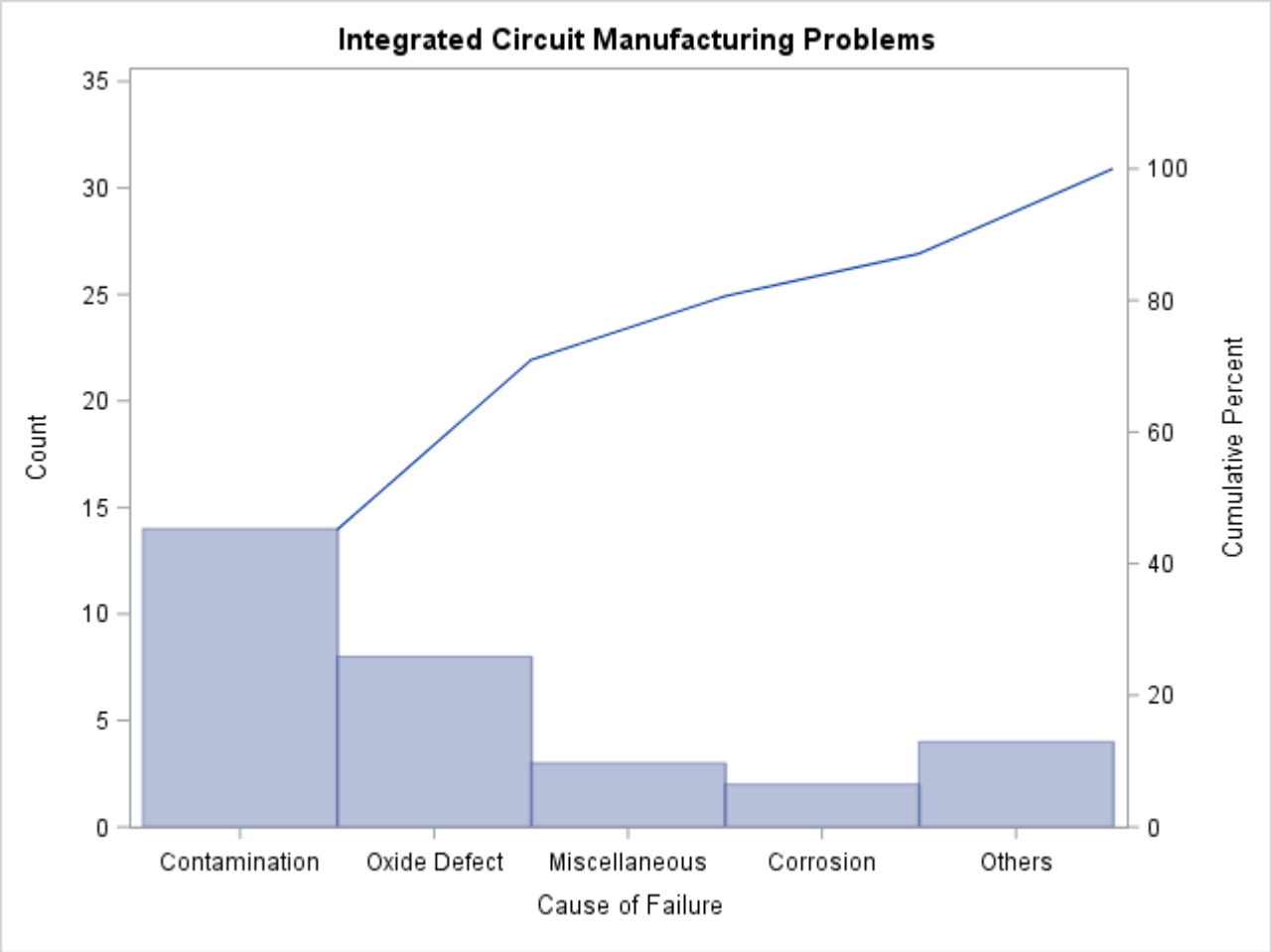


You can also display the most frequently occurring categories and merge the remaining categories into a single *other* category that is represented by a bar. You can specify the name for the new category with the OTHER= option. If, in addition, you specify that name in the LAST= option, the category is positioned at the bottom of the chart. The following statements illustrate both options:

```
title 'Integrated Circuit Manufacturing Problems';
proc pareto data=Failure2;
   vbar Cause / freq     = Count
                scale    = count
                maxncat  = 5
                other    = 'Others'
                last     = 'Others'
                odstitle = title1;
run;
```

The chart is shown in Figure 16.6.

Figure 16.6 Restricted Pareto Chart with *Other* Category



The number of categories displayed is five, which is the number specified in the MAXNCAT= option. The first four categories are the four most frequently occurring problems in Failure2, and the fifth category merges the remaining problems.

Note that `'Corrosion'` and `'Metallization'` both have a frequency of two. When the MAXNCAT= option is applied to categories with tied frequencies, PROC PARETO breaks the tie by using the order of the formatted values. Thus `'Corrosion'` is displayed, whereas `'Metallization'` is merged into the `'Other'` category. The MAXNCAT= and related options are described in the section "Restricted Pareto Charts" on page 1118.

Displaying Summary Statistics on a Pareto Chart

NOTE: See *Displaying Summary Statistics on a Pareto Chart* in the SAS/QC Sample Library.

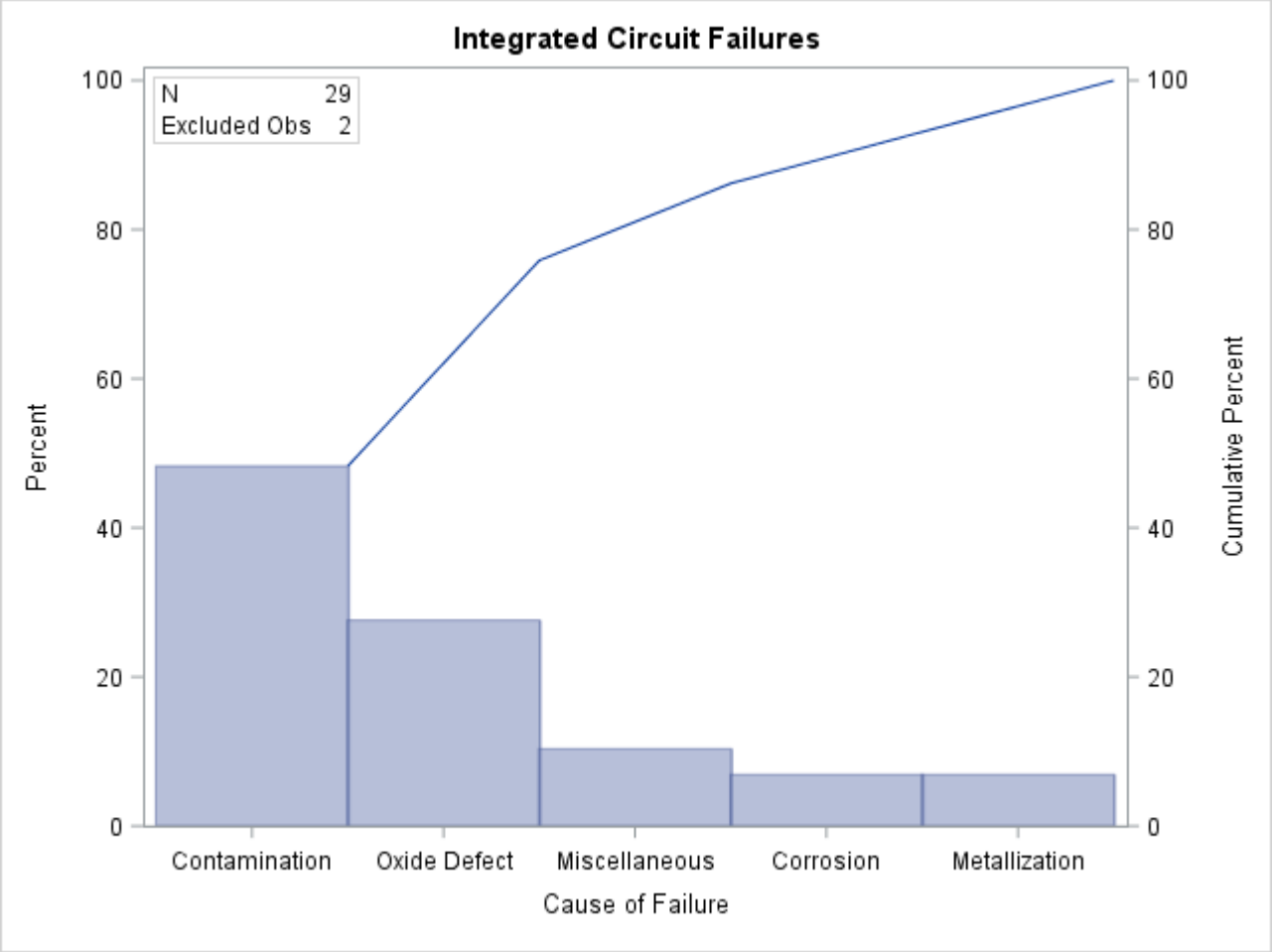
You can use an INSET statement to add a box or table (referred to as an *inset*) of summary statistics on a Pareto chart. The following statements generate a chart from the Failure2 data set and limit the number of categories to five:

```
data Failure2;
   length Cause $ 16 ;
   label  Cause = 'Cause of Failure' ;
   input  Cause $ 1-16 Count;
   datalines;
Contamination    14
Corrosion         2
Doping            1
Metallization     2
Miscellaneous     3
Oxide Defect      8
Silicon Defect    1
;


title 'Integrated Circuit Failures';
proc pareto data=Failure2;
   vbar Cause /
      freq     = Count
      maxncat  = 5
      odstitle = title;
   inset n nexcl;
run;
```

An INSET statement produces an inset on the chart that is created by the preceding HBAR or VBAR chart statement. You specify inset keywords to request summary statistics, and the statistics appear in the order in which you specify the keywords. The keyword N displays the number of categories that are displayed in the chart; the keyword NEXCL displays the number of categories that are excluded. A complete list of keywords available with the INSET statement is provided in the section "INSET Statement Keywords" on page 1086.

The resulting chart is displayed in Figure 16.7.

Figure 16.7 A Pareto Chart with an Inset



Syntax: PARETO Procedure

The following statements are available in the PARETO procedure:

PROC PARETO < *options* > **;**
BY *variables* **;**
HBAR (*variable-list*)< */ options* > **;**
VBAR (*variable-list*)< */ options* > **;**
INSET *keyword-list* < */ options* > **;**

You must specify the PROC PARETO statement and at least one HBAR or VBAR chart statement. A *chart statement* specifies the process variables that you want to analyze and produces a Pareto chart for each. You can specify any number of chart statements, and all other statements are optional.

The following statements request a vertical Pareto chart for the process variable Reason from the data set Failures. When the process *variable-list* contains only one variable, you do not need to enclose it in parentheses.

```
proc pareto data=Failures;
   vbar Reason;
run;
```

The following sections describe the PROC PARETO statement and then describe the other statements in alphabetical order.

PROC PARETO Statement

PROC PARETO < *options* > **;**

The PROC PARETO statement invokes the PARETO procedure. Table 16.2 summarizes the *options* available in the PROC PARETO statement.

Table 16.2 PROC PARETO Statement Options

Option	Description
General Option	
DATA=	Specifies the input SAS data set
Traditional Graphics Options	
ANNOTATE=	Specifies the annotation data set for the frequency axis
ANNOTATE2=	Specifies the annotation data set for the cumulative percentage axis
GOUT=	Specifies the graphics catalog for saving traditional graphics output
Legacy Line Printer Chart Options	
FORMCHAR=	Specifies the formatting characters that are used to construct line printer charts
LINEPRINTER	Creates line printer charts

You can specify the following *options*:

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains annotation variables as described in *SAS/GRAPH: Reference*. You can use *SAS-data-set* to customize traditional graphics charts with features such as labels that explain critical categories. The ANNOTATE= data set is associated with the frequency axis. If the annotation is based on data coordinates, you must use the same units as the frequency axis uses. Features provided in this data set are added to every chart that PROC PARETO produces in its current run. This option has no effect when ODS Graphics is enabled.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an input data set that contains annotation variables as described in *SAS/GRAPH: Reference*. You can use *SAS-data-set* to customize traditional graphics charts with features such as labels that explain critical categories. The ANNOTATE2= data set is associated with the cumulative percentage axis. If the annotation is based on data coordinates, you must use the same units as the cumulative percentage axis uses. Features provided in this data set are added to every chart that PROC PARETO produces in its current run. This option has no effect when ODS Graphics is enabled.

DATA=*SAS-data-set*
specifies an input data set that contains the process variables and related variables. If you do not specify a DATA= data set, PROC PARETO uses the most recently created data set.

FORMCHAR='*string*'
specifies a list of corner characters and other special characters that enhance the appearance of legacy line printer charts.

If your device supports the ASCII symbol set (1 or 2), use the following list:

```
formchar = 'B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9'X
```

The FORMCHAR= option overrides (but does not alter) the FORMCHAR= option that is specified in an OPTIONS statement such as in the following statement:

```
options formchar = 'B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9'X;
```

You can place the OPTIONS statement at the top of your SAS program or in an AUTOEXEC.SAS file. The FORMCHAR= has no effect unless you specify LINEPRINTER option.

GOUT=*graphics-catalog*
specifies the graphics catalog in which to save traditional graphics output. This option has no effect when ODS Graphics is enabled.

LINEPRINTER
requests that legacy line printer charts be produced. The HBAR statement does not produce line printer output, so you cannot use an HBAR statement when you specify the LINEPRINTER option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PARETO to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the PARETO procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

HBAR Statement

HBAR (*variable-list* **)**< **/** *options*> **;**

The HBAR statement creates a Pareto chart that uses horizontal bars to represent the frequencies of problems in a process or operation. The HBAR statement does not produce line printer charts, so you cannot specify it when you specify the LINEPRINTER option in the PROC PARETO statement.

A horizontal Pareto chart has a vertical category axis. The frequency axis appears at the top of the chart and measures the lengths of the bars on the chart. The cumulative percentage axis is at the bottom of the chart and measures the cumulative percentage curve.

The HBAR statement produces two types of output for Pareto charts:

- It produces ODS Graphics output if ODS Graphics is enabled (for example, by specifying the ODS GRAPHICS ON statement prior to the PROC statement).
- Otherwise, it produces traditional graphics if SAS/GRAPH is licensed.

For more information about producing these different types of graphs. see Chapter 4, "SAS/QC Graphics,"

The *variable-list* specifies the process variables to be analyzed. PROC PARETO creates a chart for each variable, and the values of each variable determine the Pareto categories for that chart. If *variable-list* contains only one process variable, you do not need to enclose it in parentheses.

The variables can be numeric or character, and the maximum length of a character variable is 64. Formatted values determine the categories and are displayed in labels and legends. The maximum format length is 64.

Table 16.3 lists the HBAR statement *options* by function. For complete descriptions, see the section "Dictionary of HBAR and VBAR Statement Options" on page 1095.

Table 16.3 HBAR Statement Options

Option	Description
Data Processing Options	
FREQ=	Specifies the frequency variable
MISSING	Requests that missing values of the process variable be treated as a Pareto category
MISSING1	Requests that missing values of the first CLASS= variable be analyzed as a level
MISSING2	Requests that missing values of the second CLASS= variable be analyzed as a level
OUT=	Creates an output data set that saves the information that is displayed in the Pareto chart
WEIGHT=	Specifies weight variables used to weight frequencies
Options for Restricting the Number of Categories	
LOTHER=	Specifies a label for the OTHER= bar
MAXCMPCT=	Displays only the categories whose cumulative percentage is less than the specified percentage

Table 16.3 (*continued*)

Option	Description
MAXNCAT=	Displays only the categories that have the *n* highest values
MINPCT=	Displays only the categories whose percentages are greater than the specified percentage
OTHER=	Merges all categories that are not displayed
OTHERCVAL=	Specifies an OUT= data set character variable value for the OTHER= category
OTHERNVAL=	Specifies an OUT= data set numeric variable value for the OTHER= category
Options for Displaying Bars	
BARLABEL=	Displays labels for bars
BARS=	Specifies a variable that groups bars for a display by using ODS style colors
CHIGH(*n*)	Specifies color for bars that have the *n* highest values
CLOW(*n*)	Specifies color for bars that have the *n* lowest values
LABOTHER=	Specifies a label for the OTHER= category
LAST=	Specifies the bottommost category
Options for the Cumulative Percentage Curve	
ANCHOR=	Specifies the corner of topmost bar to which the curve is anchored
CMPCTLABEL	Labels curve points with their values
NOCURVE	Suppresses the cumulative percentage curve
NOCUMLABEL	Suppresses the cumulative percentage axis label
NOCUMTICK	Suppresses the cumulative percentage axis tick marks and tick mark labels
Options for Comparative Pareto Charts	
CLASS=	Specifies classification variables
CLASSKEY=	Specifies the key cell
CPROP	Requests proportion-of-frequency bars
INTERTILE=	Specifies the distance in screen percentage units between tiles
MISSING1	Requests that missing values of the first CLASS= variable be analyzed as a level
MISSING2	Requests that missing values of the second CLASS= variable be analyzed as a level
NCOLS=	Specifies the number of columns
NOKEYMOVE	Suppresses the placement of the key cell in the top left corner
NROWS=	Specifies the number of rows
ORDER1=	Specifies the order in which values of the first CLASS= variable are displayed
ORDER2=	Specifies the order in which values of the second CLASS= variable are displayed

Table 16.3 (*continued*)

Option	Description
Options for Controlling Axes	
AXISFACTOR=	Specifies the distance factor between the longest bar and the right frame
FREQAXIS=	Specifies tick mark values for the frequency axis
FREQAXISLABEL=	Labels the frequency axis
CUMAXIS=	Specifies tick mark values for the cumulative percentage axis
CUMAXISLABEL=	Specifies a label for the cumulative percentage axis
FREQOFFSET=	Specifies the frequency axis offset in screen percentage units
GRID	Adds a grid that corresponds to the frequency axis
GRID2	Adds a grid that corresponds to the cumulative percentage axis
NOCHART	Suppresses the Pareto chart
NOFREQLABEL	Suppresses the frequency axis label
NOCUMLABEL	Suppresses the cumulative percentage axis label
NOFREQTICK	Suppresses tick marks and tick mark labels for the frequency axis
NOCUMTICK	Suppresses tick marks and tick mark labels for the cumulative percentage axis
NOCATLABEL	Suppresses the category axis label
SCALE=	Specifies the units in which the frequency axis is scaled
CATOFFSET=	Specifies the category axis offset in screen percentage units
Options for Reference Lines	
CATREF=	Requests reference lines perpendicular to the category axis
CATREFLABELS=	Specifies labels for CATREF= lines
CUMREF=	Requests reference lines perpendicular to the cumulative percentage axis
CUMREFLABELS=	Specifies labels for CUMREF= lines
FREQREF=	Requests reference lines perpendicular to the frequency axis
FREQREFLABELS=	Specifies labels for FREQREF= lines
HREFLABPOS=	Specifies the position of FREQREFLABELS= and CUMREFLABELS= labels
VREFLABPOS=	Specifies the position of CATREFLABELS= labels
Options for Displaying Legends	
BARLEGEND=	Displays a legend for the BARS=, CBARS=, or PBARS= options
BARLEGLABEL=	Displays a label for the BARLEGEND= legend
CATLEGLABEL=	Specifies a label for the Pareto categories legend
CFRAMENLEG	Frames the sample size legend
HLLEGLABEL=	Displays a label for the legend that describes colors and patterns of the highest or lowest bars
NLEGEND=	Requests a sample size legend
NOHLLEG	Suppresses the legend that describes colors and patterns of the highest and lowest bars

Table 16.3 (*continued*)

Option	Description
Options for ODS Graphics Output	
CATLEGEND=	Controls the display of the Pareto categories legend
CHARTTYPE=	Specifies the type of Pareto chart to be produced
MARKERS	Requests markers on the cumulative percentage curve
ODSFOOTNOTE=	Specifies a footnote to be displayed on the chart
ODSFOOTNOTE2=	Specifies a secondary footnote to be displayed on the chart
ODSTITLE=	Specifies a title to be displayed on the chart
ODSTITLE2=	Specifies a secondary title to be displayed on the chart
URL=	Specifies a variable whose values are URLs to be associated with bars
Options for Traditional Graphics	
ANGLE=	Rotates category axis tick mark labels
ANNOKEY	Applies annotation only to the key cell
ANNOTATE=	Specifies an annotation data set that uses frequency axis data units
ANNOTATE2=	Specifies an annotation data set that uses cumulative percentage axis data units
BARLABPOS=	Specifies the position of BARLABEL= labels
BARWIDTH=	Specifies the width (vertical dimension) of the bars in screen percentage units
CAXIS=	Specifies the axis color
CAXIS2=	Specifies the color for the cumulative percentage axis and tick marks
CBARLINE=	Specifies the color for bar outlines
CBARS=	Specifies the color for bars
CCATREF=	Specifies the color for CATREF= lines
CCONNECT=	Specifies the color for the curve
CCUMREF=	Specifies the color for CUMREF= lines
CFRAME=	Specifies the color for the area enclosed by axes and frame
CFRAMESIDE=	Specifies the frame color for row labels
CFRAMETOP=	Specifies the frame color for column labels
CFREQREF=	Specifies the color for FREQREF= lines
CGRID=	Specifies the color for frequency axis grid lines
CGRID2=	Specifies the color for cumulative percentage axis grid lines
CLIPREF	Draws reference lines behind bars
COTHER=	Specifies the color for OTHER= bar
CTEXT=	Specifies the color for text
CTEXTSIDE=	Specifies the color for row labels
CTEXTTOP=	Specifies the color for column labels
CTILES=	Specifies the colors for tile backgrounds
DESCRIPTION=	Specifies a description of the Pareto chart's GRSEG catalog entry
FONT=	Specifies the font for text

Table 16.3 (*continued*)

Option	Description
FRONTREF	Draws reference lines in front of bars
HEIGHT=	Specifies the text height in screen percentage units
HTML=	Specifies a variable whose values create links that are associated with bars in traditional graphics output
INFONT=	Specifies the font for text inside the frame
INHEIGHT=	Specifies the text height in screen percentage units for text inside the frame
INTERBAR=	Specifies the distance between bars in screen percentage units
LCATREF=	Specifies the line type for CATREF= lines
LCUMREF=	Specifies the line type for CUMREF= lines
LFREQREF=	Specifies the line type for FREQREF= lines
LGRID=	Specifies the line type for frequency axis grid lines
LGRID2=	Specifies the line type for cumulative percentage axis grid lines
NAME=	Specifies the name of the Pareto chart's GRSEG catalog entry
NOFRAME	Suppresses the axis frame
PBARS=	Specifies the pattern for the bars
PHIGH(*n*)=	Specifies the pattern for bars that have the *n* highest values
PLOW(*n*)=	Specifies the pattern for bars that have the *n* lowest values
POTHER=	Specifies the pattern for the OTHER= bar
TILELEGEND=	Specifies a legend for the CTILES= colors
TILELEGLABEL=	Specifies label for TILELEGEND= legend
WAXIS=	Specifies the width in pixels for the axes and frame
WBARLINE=	Specifies the width for bar outlines
WGRID=	Specifies the width of frequency axis grid lines
WGRID2=	Specifies the width of cumulative percentage axis grid lines

INSET Statement

INSET *keyword-list* < / *options* > **;**

The INSET statement enables you to enhance a Pareto chart by adding a box or table (called an *inset*) of summary statistics directly to the graph. An inset can display statistics that are calculated by the PARETO procedure or arbitrary values that are provided in a SAS data set.

An INSET statement must follow a chart statement, and it produces an inset on that chart. More than one INSET statement can apply to the same chart statement. When the chart statement produces a comparative chart, an associated INSET statement produces an inset in every cell of the chart. Statistics that are displayed in the inset of a cell are computed from the data that are associated with that cell.

NOTE: When ODS Graphics is enabled, only one INSET statement can be associated with a comparative Pareto chart. Insets are not available with legacy line printer charts, so the INSET statement is not applicable when you specify the LINEPRINTER option in the PROC PARETO statement.

The *keyword-list* can include any of the keywords listed in Table 16.4. Statistics are displayed in the order in which the keywords are specified. Each *keyword-list* entry has the following form:

keyword <='*label*' > <(*format*) >

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide a customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotation marks. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. If you specify both a label and a format for a statistic, the label must appear before the format. See Example 16.10.

Note the difference between *keywords* and *options*: *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset. You can use *options* in the INSET statement to do the following:

- specify the position of the inset
- specify a header for the inset
- specify enhancements for traditional graphics, such as background colors, text colors, text height, text font, and drop shadows

Table 16.5 lists available INSET statement *options*.

The following statements produce a vertical Pareto chart with insets in the upper left (northwest) and upper right (northeast) corners, and a horizontal comparative Pareto chart with insets in each cell.

```
proc pareto data=Failure3;
   vbar Cause / maxncat = 5 other = 'Others';
      inset nothercat / position = nw;
      inset nother / position = ne;
   hbar Cause / class = Stage;
      inset n;
run;
```

INSET Statement Keywords

Table 16.4 lists the *keywords* available in the INSET statement.

Table 16.4 INSET Statement Keywords

Keyword	Description
DATA=*SAS-data-set*	Reads (label, value) pairs from a SAS data set
N	Specifies the sample size
NEXCL	Specifies the number of observations excluded from a restricted Pareto chart
NOTHER	Specifies the number of observations in the OTHER= category
NOTHERCAT	Specifies the number of categories merged to form the OTHER= category

Table 16.4 (*continued*)

Keyword	Description
SUMWGTS	Specifies the sum of weighted frequencies across all categories

The NOTHERCAT and NOTHER statistics are 0 if the OTHER= option is not specified. The NEXCL statistic is 0 if the OTHER= option is specified.

All INSET keywords request a single statistic in an inset, except for the DATA= keyword. The DATA= keyword specifies a SAS data set that contains (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ (a character variable whose values provide labels for inset entries) and _VALUE_ (which can be character or numeric and provides values displayed in the inset). The label and value from each observation in the DATA= data set occupy one line in the inset. Example 16.11 illustrates the use of the DATA= keyword.

INSET Statement Options

Figure 16.8 illustrates the terms that are used in this section.

Figure 16.8 Inset Terms

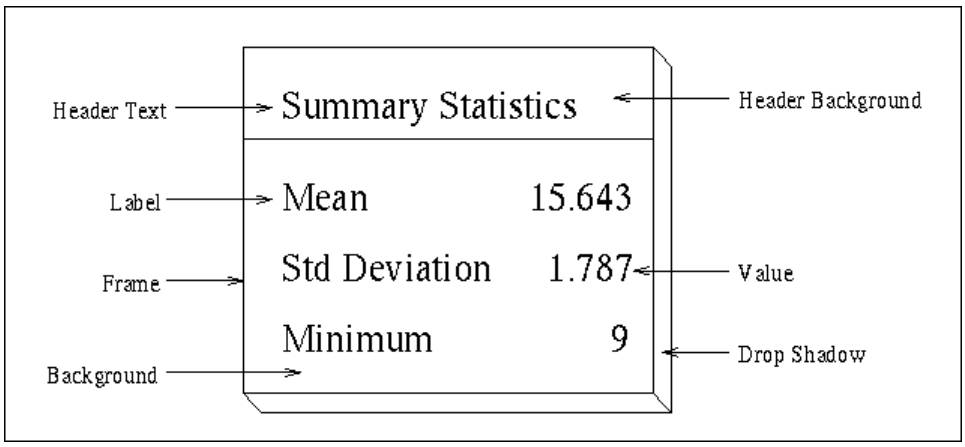


Table 16.5 lists the *options* available in the INSET statement.

Table 16.5 INSET Statement Options

Keyword	Description
General Options	
FORMAT=	Specifies the format for numeric values in the inset
HEADER=	Specifies the header text
NOFRAME	Suppresses the frame around the inset
POSITION=	Specifies the position of the inset

Table 16.5 (*continued*)

Option	Description
Options for ODS Graphics Output	
CFILL	Specifies the color of the inset background
GUTTER=	Specifies the gutter width for an inset in the top or bottom margin
NCOLS=	Specifies the number of columns for an inset in the top or bottom margin
Options for Traditional Graphics	
CFILL=	Specifies the color of the inset background
CFILLH=	Specifies the color of the header background
CFRAME=	Specifies the color of the frame
CHEADER=	Specifies the color of the header text
CSHADOW=	Specifies the color of the drop shadow
CTEXT=	Specifies the color of the inset text
DATA	Specifies that POSITION=(x,y) coordinates are in data units
FONT=	Specifies the text font
HEIGHT=	Specifies the height of the inset text
REFPOINT=	Specifies the reference point of an inset that is positioned by POSITION=(x,y) coordinates

The following entries provide detailed descriptions of *options* in the INSET statement.

General Options

You can specify the following *options* when you use either ODS Graphics or traditional graphics:

FORMAT=*format*
specifies a format for all the values that are displayed in an inset. If you specify a format for a particular statistic, then that format overrides the format you specify in this option.

HEADER= *'string'*
specifies the header text. The *string* cannot exceed 40 characters. If you do not specify this option, no header line appears in the inset.

NOFRAME
suppresses the frame drawn around the inset.

POSITION=*position*

POS=*position*
determines the position of the inset. The *position* can be a compass point keyword (N, NE, E, SE, S, SW, W, or NW), a margin keyword (TM, RM, BM, or LM), or a pair of coordinates *(x,y)*. You can specify coordinates in axis percentage units or axis data units. For more information, see the section "Positioning Insets" on page 1121. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

NOTE: You cannot use the POSITION= option to specify coordinates when producing ODS Graphics output.

ODS Graphics Options

You can specify the following *options* when you use ODS Graphics:

CFILL

CFILL=BLANK
specifies the color of the inset background. If you do not specify this option, the inset background is transparent. This means that items that are overlapped by the inset (such as Pareto bars or the cumulative percentage curve) show through the inset. If you specify this option without an argument, the background is opaque and its color is specified by the Color attribute of the GraphBackground style element in the current ODS style. If you specify CFILL=BLANK, the background is opaque and its color is specified by the Color attribute of the GraphWalls style element in the current ODS style.

GUTTER=*value*
specifies the gutter width in screen percentage units for an inset that is located in the top or bottom margin. The gutter is the space between columns of values in an inset.

NCOLS=*n*
specifies the number of columns of (label, value) pairs that are displayed in an inset that is located in the top or bottom margin.

Traditional Graphics Options

You can specify the following *options* when you produce traditional graphics.

CFILL=*color* | **BLANK**
specifies the color of the inset background (including the header background if you do not specify the CFILLH= option). If you specify CFILL=BLANK, the background color is determined by the Color attribute of the GraphWalls style element in the current ODS style. If you do not specify this option, the inset background is transparent. This means that items overlapped by the inset (such as Pareto bars or the cumulative percentage curve) show through the inset.

CFILLH=*color*
specifies the color of the header background. If you do not specify this option, the CFILL= color is used.

CFRAME=*color*
specifies the color of the inset frame. The default color is specified by the ContrastColor attribute of the GraphBorderLines style element in the current ODS style.

CHEADER=*color*
specifies the color of the header text. If you do not specify this option, the CTEXT= color is used.

CSHADOW=*color*

CS=*color*
specifies the color of the drop shadow. See Output 16.11.1 for an example. If you do not specify this option, a drop shadow is not displayed.

CTEXT=*color*

CT=*color*

specifies the color of the text. The default *color* is specified by the Color attribute of the GraphValueText style element in the current ODS style.

DATA

specifies that data coordinates be used in positioning the inset with the POSITION= option. You can specify this option only when you specify POSITION=(*x*,*y*), and you must include it immediately after the coordinates (*x*,*y*). For more information, see the section "Using Coordinates to Position Insets" on page 1123. See Figure 16.11 for an example.

FONT=*font*

specifies the font of the text. The default font is determined by the FontFamily, FontStyle, and FontWeight attributes of the GraphValueText style element in the current ODS style.

HEIGHT=*value*

specifies the height of the text in the inset. The default value is specified by the FontSize attribute of the GraphValueText style element in the current ODS style.

REFPOINT=BR | BL | TR | TL

RP=BR | BL | TR | TL

specifies the reference point for an inset that is positioned by a pair of coordinates (*x*,*y*), which are specified in the POSITION= option. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (*x*,*y*). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 16.12 for an example. By default, REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, this option is ignored. For more information, see "Using Coordinates to Position Insets" on page 1123.

VBAR Statement

VBAR (*variable-list* **)**< / *options*> **;**

The VBAR statement creates a Pareto chart in which vertical bars represent the frequencies of problems in a process or operation. A vertical Pareto chart has a horizontal category axis. The frequency axis is oriented vertically on the left side of the chart and measures the lengths of the bars on the chart. The cumulative percentage axis is on the right of the chart and measures the cumulative percentage curve.

The VBAR statement produces three types of output for Pareto charts:

- It produces ODS Graphics output if ODS Graphics is enabled (for example, by specifying the ODS GRAPHICS ON statement prior to the PROC statement).
- Otherwise, it produces traditional graphics by default if SAS/GRAPH is licensed.
- It produces legacy line printer charts when you specify the LINEPRINTER option in the PROC statement.

For more information about producing these different types of graphs, see Chapter 4, "SAS/QC Graphics."

The *variable-list* specifies the process variables to be analyzed. A chart is created for each variable, and the values of each variable determine the Pareto categories for that chart. If *variable-list* contains only one process variable, you do not need to enclose it in parentheses.

The variables can be numeric or character, and the maximum length of a character variable is 64. Formatted values are used to determine the categories and are displayed in labels and legends. The maximum format length is 64.

Table 16.6 lists the VBAR statement options by function. For complete descriptions, see the section "Dictionary of HBAR and VBAR Statement Options" on page 1095.

Table 16.6 VBAR Statement Options

Option	Description
Data Processing Options	
FREQ=	Specifies the frequency variable
MISSING	Requests that missing values of the process variable be treated as a Pareto category
MISSING1	Requests that missing values of the first CLASS= variable be analyzed as a level
MISSING2	Requests that missing values of the second CLASS= variable be analyzed as a level
OUT=	Creates an output data set that saves the information that is displayed in the Pareto chart
WEIGHT=	Specifies weight variables that are used to weight frequencies
Options for Restricting the Number of Categories	
LOTHER=	Specifies a label for the OTHER= bar
MAXCMPCT=	Displays only the categories whose cumulative percentage is less than the specified percentage
MAXNCAT=	Displays only the categories that have the *n* highest values
MINPCT=	Displays only the categories that have percentages greater than the specified percentage
OTHER=	Merges all categories that are not displayed
OTHERCVAL=	Specifies an OUT= data set character variable value for the OTHER= category
OTHERNVAL=	Specifies an OUT= data set numeric variable value for the OTHER= category
Options for Displaying Bars	
BARLABEL=	Displays labels for bars
BARS=	Specifies a variable that groups bars for a display by using ODS style colors
CHIGH(*n*)	Specifies the color for bars that have the *n* highest values
CLOW(*n*)	Specifies the color for bars that have the *n* lowest values
LABOTHER=	Specifies a label for the OTHER= category
LAST=	Specifies the bottommost category

Table 16.6 (*continued*)

Option	Description
Options for the Cumulative Percent Curve	
ANCHOR=	Specifies the corner of the topmost bar to which the curve is anchored
CMPCTLABEL	Labels curve points with their values
NOCURVE	Suppresses the cumulative percentage curve
NOCUMLABEL	Suppresses the cumulative percentage axis label
NOCUMTICK	Suppresses cumulative percentage axis tick marks and tick mark labels
Options for Comparative Pareto Charts	
CLASS=	Specifies classification variables
CLASSKEY=	Specifies the key cell
CPROP	Requests proportion-of-frequency bars
INTERTILE=	Specifies the distance in screen percentage units between tiles
MISSING1	Requests that missing values of the first CLASS= variable be analyzed as a level
MISSING2	Requests that missing values of the second CLASS= variable be analyzed as a level
NCOLS=	Specifies the number of columns
NOKEYMOVE	Suppresses the placement of the key cell in the top left corner
NROWS=	Specifies the number of rows
ORDER1=	Specifies the order in which values of the first CLASS= variable are displayed
ORDER2=	Specifies the order in which values of the second CLASS= variable are displayed
Options for Controlling Axes	
AXISFACTOR=	Specifies the distance factor between the longest bar and the top frame
FREQAXIS=	Specifies tick mark values for the frequency axis
FREQAXISLABEL=	Labels the frequency axis
CUMAXIS=	Specifies tick mark values for the cumulative percentage axis
CUMAXISLABEL=	Specifies a label for the cumulative percentage axis
FREQOFFSET=	Specifies the frequency axis offset in screen percentage units
GRID	Adds a grid that corresponds to the frequency axis
GRID2	Adds a grid that corresponds to the cumulative percentage axis
NOCHART	Suppresses the Pareto chart
NOFREQLABEL	Suppresses the frequency axis label
NOCUMLABEL	Suppresses the cumulative percentage axis label
NOFREQTICK	Suppresses tick marks and tick mark labels for the frequency axis
NOCUMTICK	Suppresses tick marks and tick mark labels for the cumulative percentage axis
NOCATLABEL	Suppresses the category axis label
SCALE=	Specifies units in which the frequency axis is scaled

Table 16.6 (*continued*)

Option	Description
CATOFFSET=	Specifies the category axis offset in screen percentage units
Options for Reference Lines	
CATREF=	Requests reference lines perpendicular to the category axis
CATREFLABELS=	Specifies labels for CATREF= lines
CUMREF=	Requests reference lines perpendicular to the cumulative percentage axis
CUMREFLABELS=	Specifies labels for CUMREF= lines
FREQREF=	Requests reference lines perpendicular to the frequency axis
FREQREFLABELS=	Specifies labels for FREQREF= lines
HREFLABPOS=	Specifies the position of FREQREFLABELS= and CUMREFLABELS= labels
VREFLABPOS=	Specifies the position of CATREFLABELS= labels
Options for Displaying Legends	
BARLEGEND=	Displays a legend for the BARS=, CBARS=, or PBARS= options
BARLEGLABEL=	Displays a label for BARLEGEND= legend
CATLEGLABEL=	Specifies a label for the Pareto categories legend
CFRAMENLEG	Frames the sample size legend
HLLEGLABEL=	Displays a label for the legend that describes colors and patterns of the highest and lowest bars
NLEGEND=	Requests a sample size legend
NOHLLEG	Suppresses the legend that describes colors and patterns of the highest and lowest bars
Options for ODS Graphics Output	
CATLEGEND=	Controls the display of the Pareto categories legend
CHARTTYPE=	Specifies the type of Pareto chart produced
MARKERS	Requests markers on the cumulative percentage curve
ODSFOOTNOTE=	Specifies a footnote to be displayed on the chart
ODSFOOTNOTE2=	Specifies a secondary footnote to be displayed on the chart
ODSTITLE=	Specifies a title to be displayed on the chart
ODSTITLE2=	Specifies a secondary title to be displayed on the chart
URL=	Specifies a variable whose values are URLs to be associated with bars
Options for Traditional Graphics	
ANGLE=	Rotates the category axis tick mark labels
ANNOKEY	Applies annotation only to the key cell
ANNOTATE=	Specifies an annotation data set that uses frequency axis data units
ANNOTATE2=	Specifies an annotation data set that uses cumulative percentage axis data units
BARLABPOS=	Specifies the position of the BARLABEL= labels

Table 16.6 (*continued*)

Option	Description
BARWIDTH=	Specifies the width (horizontal dimension) of the bars in screen percentage units
CAXIS=	Specifies the axis color
CAXIS2=	Specifies the color for the cumulative percentage axis and tick marks
CBARLINE=	Specifies the color for bar outlines
CBARS=	Specifies the color for bars
CCATREF=	Specifies the color for CATREF= lines
CCONNECT=	Specifies the color for the curve
CCUMREF=	Specifies the color for CUMREF= lines
CFRAME=	Specifies the color for the area that is enclosed by axes and frame
CFRAMESIDE=	Specifies the frame color for row labels
CFRAMETOP=	Specifies the frame color for column labels
CFREQREF=	Specifies the color for FREQREF= lines
CGRID=	Specifies the color for the frequency axis grid lines
CGRID2=	Specifies the color for the cumulative percentage axis grid lines
CLIPREF	Draws reference lines behind bars
COTHER=	Specifies the color for the OTHER= bar
CTEXT=	Specifies the color for text
CTEXTSIDE=	Specifies the color for row labels
CTEXTTOP=	Specifies the color for column labels
CTILES=	Specifies the colors for tile backgrounds
DESCRIPTION=	Specifies a description of the Pareto chart's GRSEG catalog entry
FONT=	Specifies the text font
FRONTREF	Draws reference lines in front of bars
HEIGHT=	Specifies the text height in screen percentage units
HTML=	Specifies a variable whose values create links that are associated with bars in traditional graphics output
INFONT=	Specifies the font for text inside frame
INHEIGHT=	Specifies the text height in screen percentage units for text inside the frame
INTERBAR=	Specifies the distance between bars in screen percentage units
LCATREF=	Specifies the line type for the CATREF= lines
LCUMREF=	Specifies the line type for the CUMREF= lines
LFREQREF=	Specifies the line type for the FREQREF= lines
LGRID=	Specifies the line type for the frequency axis grid lines
LGRID2=	Specifies the line type for the cumulative percentage axis grid lines
NAME=	Specifies the name of the Pareto chart's GRSEG catalog entry
NOFRAME	Suppresses the axis frame
PBARS=	Specifies a pattern for the bars
PHIGH(*n*)=	Specifies the pattern for the bars that have the *n* highest values

Table 16.6 (*continued*)

Option	Description
PLOW(*n*)=	Specifies the pattern for the bars that have the *n* lowest values
POTHER=	Specifies the pattern for the OTHER= bar
TILELEGEND=	Specifies a legend for the CTILES= colors
TILELEGLABEL=	Specifies the label for the TILELEGEND= legend
TURNVLABEL	Turns and strings vertically the characters in the frequency and cumulative percentage axis labels
WAXIS=	Specifies the width in pixels for the axes and frame
WBARLINE=	Specifies the width for bar outlines
WGRID=	Specifies the width of frequency axis grid lines
WGRID2=	Specifies the width of cumulative percentage axis grid lines
Options for Legacy Line Printer Charts	
CONNECTCHAR=	Specifies the plot character for the cumulative percentage curve segments
HREFCHAR=	Specifies the plot character for category reference lines
VREFCHAR=	Specifies the plot character for frequency and cumulative percentage reference lines
SYMBOLCHAR=	Specifies the plot character for points on the cumulative percentage curve

Dictionary of HBAR and VBAR Statement Options

This section provides detailed descriptions of *options* you can specify after the slash (/) in the HBAR and VBAR statements. For example, to request that the frequency axis of a vertical Pareto chart be scaled by counts, use the SCALE= option as follows:

```
proc pareto data=failure;
   vbar cause / scale = count;
run;
```

This section consists of the following subsections:

- The section "General Options" on page 1096 contains descriptions of general Pareto chart options.
- The section "Options for Traditional Graphics" on page 1110 describes options that apply only when traditional graphics output is produced, as when ODS Graphics is disabled.
- The section "Options for Legacy Line Printer Charts" on page 1117 contains descriptions of options that apply only to legacy line printer charts, which are produced by VBAR statements when you specify the LINEPRINTER option in the PROC PARETO statement.

NOTE: The terminology used in the option descriptions describes vertical Pareto charts. For example, the "tallest" bar is the one that extends farthest along the frequency axis, whether it is oriented vertically or horizontally.

General Options

You can specify the following general options:

ANCHOR=*keyword*
specifies where the Pareto curve is anchored to the first bar on the chart. Table 16.7 describes the position keywords available in the HBAR and VBAR statements.

Table 16.7 ANCHOR= Option Keywords

HBAR Keyword	Anchoring Position
BR	Bottom right corner (default)
LC	Left center
RC	Right center
TL	Top left corner

VBAR Keyword	Anchoring Position
BC	Bottom center
BL	Bottom left corner
TC	Top center
TR	Top right corner (default)

See Output 16.2.1 for an illustration.

AXISFACTOR=*value*
specifies a factor used in scaling the frequency axis. This factor determines (approximately) the ratio of the length of the axis to the length of the tallest bar, and it is used to provide space for the cumulative percentage curve. The *value* must be greater than or equal to 1.

By default, the factor is chosen so that the curve is anchored at the top right corner of the first bar (see also the ANCHOR= option). However, if anchoring to the top of the first bar causes the bars to be flattened excessively, a smaller default factor is used.

This option is not applicable if the cumulative percentage curve is suppressed by the NOCURVE option.

BARLABEL=CMPCT | COUNT | VALUE | (*variable-list***)**
requests that a label be displayed for each bar. You can specify the following values:

CMPCT specifies that the label indicates the cumulative percentage that is associated with that bar. An alternative to BARLABEL=CMPCT is the CMPCTLABEL option, which labels points on the cumulative percentage curve with their values.

COUNT specifies that the label displays the count for the bar, regardless of the SCALE= option setting.

VALUE specifies that the label indicates the height of the bar in the units used by the frequency axis. The units are determined by the SCALE= option setting. See Example 16.8 for an illustration.

(*variable-list***)** specifies that the label displays the values of one or more variables from the input data set. If a format is associated with a variable, then the formatted value is displayed. Values can be up to 32 characters long. The variable values must be consistent within observations that correspond to a particular Pareto category. The variables are saved in the OUT= data set. If you specify more than one process variable in the chart statement, you can specify more than one variable in *variable-list*. The BARLABEL= and process variables are matched by their positions in their respective variable lists.

The space in horizontal Pareto charts might be insufficient to display long bar labels. You can specify the AXISFACTOR= option to increase the available space beyond the bars. If you are producing traditional graphics, you can use the BARLABPOS= option to specify how labels are positioned relative to the bars.

BARLEGEND=(*variable-list***)**

requests that a legend be added to the chart to explain colors for bars that are specified in the BARS= or CBARS= option, or patterns for bars that are specified in the PBARS= option. The *variable-list* must be enclosed in parentheses even if only one *variable* is specified. See Output 16.4.1 for an illustration.

The values of the variables in *variable-list* provide the explanatory labels used in the legend. If a format is associated with the variable, then the formatted value is displayed. Values can be up to 32 characters long.

This option is not applicable unless you specify one or more of the BARS=, CBARS=, or PBARS= options. In the DATA= data set, the values of the BARLEGEND= variable must be identical in observations for which the value of the BARS=, CBARS=, or PBARS= variable (or the combination of the CBARS= and PBARS= values) is the same. This ensures that the legend derived from the BARLEGEND= variable is consistent.

If you specify more than one process variable in a chart statement and a corresponding list of BARS=, CBARS=, or PBARS= variables, you can specify a list of BARLEGEND= variables. The number of variables in *variable-list* should be less than or equal to the number of process variables. The lists of variables are matched so that the first variable in *variable-list* is applied to the first process variable and the first BARS=, CBARS=, or PBARS= variable; the second variable in *variable-list* is applied to the second process variable and the second BARS=, CBARS=, or PBARS= variable; and so forth. If the process variable list is longer than *variable-list*, the charts for the extra process variables do not display a bar legend.

BARLEGLABEL='*label***'**

specifies the *label* to be displayed to the left of the legend that is created by the BARLEGEND= option. See Output 16.4.1 for an illustration.

The BARLEGLABEL= option is applicable only in conjunction with BARS=, CBARS=, or PBARS= variables. The *label* can be up to 16 characters and must be enclosed in quotation marks.

If you do not specify a *label*, the BARLEGEND= variable label is displayed (unless the label is longer than 16 characters, in which case the variable name is displayed). If you do not specify the BARLEGLABEL= option and no label is associated with the BARLEGEND= variable, no legend label is displayed.

BARS=(*variable-list*)
uses different colors to group bars of the Pareto chart for display Bars that correspond to the same value of a variable in *variable-list* are assigned the same color from the ODS style. You cannot specify the BARS= option in conjunction with the CHIGH(*n*) or CLOW(*n*) options.

If you specify more than one process variable, you can specify more than one variable in *variable-list*. The number of variables in *variable-list* should be less than or equal to the number of process variables. The two lists of variables are paired in order of their specification. If a BARS= *variable* is not provided for a process variable, the bars for that chart are filled with the default color from the ODS style.

CATLEGEND=AUTO | OFF | ON
specifies whether a category legend is created for ODS Graphics output. You can specify the following values:

AUTO	creates a category legend only when the labels would be too crowded on the category axis.
OFF	suppresses the category legend.
ON	creates a category legend.

By default, CATLEGEND=AUTO. This option is ignored if ODS Graphics is not enabled.

CATLEGLABEL='*label*'
specifies a label for the category legend. A category legend is created when there is insufficient space to label the categories along the category axis or when requested in the CATLEGEND= option. The *label* can be up to 16 characters and must be enclosed in quotation marks. The default label is "Categories:". See Example 16.3 for an illustration. This option is ignored when no category legend is produced.

CATOFFSET=*value*
specifies the length of the offset at both ends of the category axis (in screen percentage units). You can eliminate the offset by specifying CATOFFSET=0.

CATREF='*value-list*'
specifies where reference lines perpendicular to the Pareto category axis are to appear on the chart. Character values can be up to 64 characters and must be enclosed in quotation marks. The values must be values of the process variable regardless of whether the bars are numbered and a category legend is introduced.

CATREFLABELS='*label1*'...'*labeln*'
specifies *labels* for the lines that are requested in the CATREF= option. The number of labels must equal the number of lines requested. Labels can be up to 16 characters and must be enclosed in quotation marks.

CFRAMENLEG

CFRAMENLEG=EMPTY

CFRAMENLEG=*color*
displays a frame around the sample size legend that is requested in the NLEGEND option. You can specify this option in the following ways:

(no argument) fills the frame with the background color that is specified by the Color attribute of the GraphBackground style element in the current ODS style.

EMPTY produces a frame that has a transparent background.

color produces a frame whose background is *color* when you are producing traditional graphics.

CHARTTYPE=CUMULATIVE | INTERVALS<(*interval-options*)> | STANDARD

specifies the type of Pareto chart to be produced. This option is supported only for ODS Graphics output. You can specify the following options:

CUMULATIVE creates a cumulative Pareto bar chart.

INTERVALS<(*interval-options*)**>** creates a Pareto dot plot that includes acceptance intervals, which are computed using simulation. You can specify the following *interval-options* for computing acceptance intervals:

ALPHA=*value*
specifies the significance level for the acceptance intervals. By default, ALPHA=0.05.

NSAMPLES=*n*
specifies the number of random samples used in the simulation. By default, NSAMPLES=2000.

SEED=*n*
specifies the seed value for the random number generator that is used in the simulation. By default, or when you specify $n \leq 0$, a seed value is generated by using the system clock.

STANDARD creates a traditional Pareto chart.

By default, CHARTTYPE=STANDARD.

Wilkinson (2006) describes the advantages of the cumulative Pareto bar chart and the Pareto dot plot that includes acceptance intervals. See Example 16.9 for examples of these alternative Pareto charts.

CHIGH(*n*)

CHIGH(*n*)=*color*

highlights the bars that have the *n* highest frequencies by filling them with a contrasting color from the ODS style. When producing traditional graphics output, you can specify CHIGH(*n*)=*color* to select a specific color. You cannot use the CHIGH(*n*) option in conjunction with a BARS= or CBARS= variable, but you can use it together with the CLOW(*n*) and CBARS=*color* options. See Output 16.3.1 for an illustration.

CLASS=*variable*

CLASS=(*variable1 variable2***)**

creates a comparative Pareto chart by using the levels of the *variables*. If you specify two *variables*, then you must enclose in parentheses. See Example 16.1 and Example 16.2.

If you specify a single *variable*, the observations in the input data set are classified by the formatted values (levels) of the *variable*. A Pareto chart is created for the process variable values in each level,

and these component charts (referred to as cells) are arranged in an array. The cells are labeled with the class levels, and uniform horizontal and vertical axes are used to facilitate comparisons.

If you specify two *variables*, the observations in the input data set are cross-classified by the values (levels) of the *variables*. A Pareto chart is created for the process variable values in each cell of the cross-classification, and these charts are arranged in a matrix. The levels of the first *variable* label the rows, and the levels of the second *variable* label the columns. Uniform horizontal and vertical axes are used to facilitate comparisons.

The *variables* can be numeric or character. The maximum length of a character *variable* is 32. If a format is associated with a *variable*, the formatted values determine the levels. Only the first 32 characters of the formatted values are used to determine the levels. You can specify whether missing values are treated as a level by using the MISSING1 and MISSING2 options.

In traditional graphics output, only the level values are displayed in row and column headers. If a label is associated with a *variable*, the label is displayed in a second header that spans the row or column headers.

CLASSKEY='*value*'

CLASSKEY=('*value1***' '***value2***')**

specifies the key cell in a comparative Pareto chart, which is created when you specify the CLASS= option. The *key cell* is defined as the cell in which the Pareto bars are arranged in decreasing order. This order then determines the uniform category axis used for all the cells.

If you specify CLASS=*variable*, you can specify CLASSKEY='*value*' to identify the key cell as the level for which the variable is equal to *value*. The *value* can have up to 32 characters, and you must specify a formatted *value*. By default, the levels are sorted as specified by the ORDER1= option, and the key cell is the level that occurs first in this order. The cells are displayed in this order from top to bottom (or left to right, depending on the NCOLS= and NROWS= values), and consequently the key cell is displayed at the top or at the left. The cell you specify in the CLASSKEY= option is displayed at the top or at the left unless you also specify the NOKEYMOVE option.

If you specify CLASS=(*variable1 variable2*), you can specify CLASSKEY=('*value1*' '*value2*') to identify the key cell as the level for which *variable1* is equal to *value1* and *variable2* is equal to *value2*. Here, *value1* and *value2* must be formatted values, and they must be enclosed in quotation marks. By default, the levels of *variable1* are sorted in the order determined by the ORDER1= option, and then within each of these levels, the levels of *variable2* are sorted in the order determined by the ORDER2= option. The default key cell is the combination of levels of *variable1* and *variable2* that occurs first in this order. The cells are displayed in order of *variable1* from top to bottom and in order of *variable2* from left to right. Consequently, the default key cell is displayed in the upper left corner. The cell you specify in the CLASSKEY= option is displayed in the upper left corner unless you also specify the NOKEYMOVE option.

For an example of the use of the CLASSKEY= option, see Output 16.1.3.

CLOW(*n***)**

CLOW(*n***)=***color*

highlights the bars that have the *n* lowest frequencies by filling them with a contrasting color from the ODS style. When producing traditional graphics output, you can specify CLOW(*n*)=*color* to select a specific color. You cannot use the CLOW(*n*)= option in conjunction with a CBARS= variable, but you can use it together with the CBARS=*color* and CHIGH(*n*) options.

CMPCTLABEL
labels points on the cumulative percentage curve with their values. By default, the points are not labeled.

CPROP

CPROP=EMPTY

CPROP=*color*
requests that a proportion-of-frequency bar of the specified color be displayed horizontally across the top of each tile in a comparative Pareto chart. You can specify the following values:

(no argument)	creates bars that are filled with a color from the ODS style.
EMPTY	produces empty bars in traditional graphics output.
color	produces bars that are filled with *color* in traditional graphics output.

The length of the bar relative to the width of the tile indicates the proportion of the total frequency count in the chart that is represented by the tile. You can use the bars to visualize the distribution of frequency count by tile. See Output 16.1.4 for an illustration.

The CPROP= option provides a graphical alternative to the NLEGEND option, which displays the actual count. The CPROP= option is applicable only with comparative Pareto charts.

CUMAXIS=*value-list*
specifies tick mark values for the cumulative percentage axis. The values must be equally spaced and in increasing order, and the first value must be 0. You must scale the values in percentage units, and the last value must be greater than or equal to 100.

CUMAXISLABEL='*label***'**
specifies a *label*, up to 40 characters, for the cumulative percentage axis. The default *label* is "Cumulative Percent" or "Cm Pct," depending on the space available.

CUMREF=*value-list*
requests reference lines perpendicular to the cumulative percentage axis at the specified *values*. You must specify the values in cumulative percentage units.

CUMREFLABELS='*label1***'...'***labeln***'**
specifies labels for the lines that are requested in the CUMREF= option. The number of labels must equal the number of lines requested. Enclose the labels in quotation marks. Labels can be up to 16 characters.

FREQ=*variable*
specifies a frequency *variable* whose values provide the counts (numbers of occurrences) of the values of the process variable. Specifying a frequency *variable* is equivalent to replicating the observations in the input data set. The *variable* must be a numeric variable that has nonnegative integer values. See "Creating a Pareto Chart from Frequency Data" on page 1073 for an illustration. If you specify more than one process variable in the chart statement, the *variable* values are used with each process variable. If you do not specify this option, each value of the process variable is counted exactly once.

FREQAXIS=*value-list*
specifies tick mark values for the frequency axis. The values must be equally spaced and in increasing order, and the first value must be 0. You must scale the values in the same units as the bars (see the SCALE= option), and the last value must be greater than or equal to the height of the largest bar.

FREQAXISLABEL='*label*'
specifies a label, up to 40 characters, for the frequency axis. If a WEIGHT= variable is specified, its label is the default frequency axis label. Otherwise, the default label depends on the value of the SCALE= option.

FREQOFFSET=*value*
specifies the length in screen percentage units of the offset at the upper end of the frequency axis.

FREQREF=*value-list*
specifies where reference lines perpendicular to the frequency axis are to appear on the chart. You must specify the values in the same units that are used to scale the frequency axis. By default, the frequency axis is scaled in percentage units, but you can specify other units in the SCALE= option. See Output 16.2.3 for an illustration.

FREQREFLABELS='*label1*'...'*labeln*'
specifies labels for the lines that are requested in the FREQREF= option. The number of labels must equal the number of lines requested. Enclose the labels in quotation marks. Labels can be up to 16 characters.

GRID
adds a grid that corresponds to the frequency axis to the Pareto chart. Grid lines are positioned at tick marks on the frequency axis. The lines are useful for comparing the heights of the bars.

GRID2
adds a grid that corresponds to the cumulative percentage axis to the Pareto chart. Grid lines are positioned at tick marks on the cumulative percentage axis. The lines are useful for reading the cumulative percentage curve.

HLLEGLABEL='*label*'
specifies a label for the legend that is automatically created when you use a combination of the CHIGH(*n*), CLOW(*n*), PHIGH(*n*), and PLOW(*n*) options. See Output 16.3.1 for an illustration. The *label* can be up to 16 characters and must be enclosed in quotation marks. The default label is "Bars:".

HREFLABPOS=*n*
specifies the vertical position of labels for reference lines that are associated with horizontal axes, which are specified in the FREQREF= and CUMREF= options in an HBAR statement or the CATREF= option in a VBAR statement. The available positions are described in the following table.

n	**Position**
1	Along top of chart
2	Staggered from top to bottom of chart
3	Along bottom of chart
4	Staggered from bottom to top of chart

By default, HREFLABPOS=1. **NOTE:** HREFLABPOS=2 and HREFLABPOS=4 are not supported for ODS Graphics output.

INTERTILE=*value*
specifies the distance in horizontal screen percentage units between tiles (cells) in a comparative Pareto chart. When ODS Graphics is enabled, the default value is 2%. In traditional graphics, the tiles are contiguous by default. See Output 16.1.3 for an illustration.

LABOTHER= '*other-label*'
is used in conjunction with the BARLABEL=(*variable*) option and specifies a label for the 'other' category that is optionally specified in the OTHER= option.

LAST='*category*'
requests that the bar that corresponds to *category* be displayed last (at the bottom of a horizontal chart or the right end of a vertical chart) regardless of the frequency that is associated with this category. The category must be a formatted value of the process variable and must be enclosed in quotation marks. The *category* can be up to 64 characters. See Figure 16.6 for an illustration.

LOTHER='*label*'
specifies a label for the bar that is defined in the OTHER= option. This label appears in the legend that is specified in the BARLEGEND= option. The *label* must be enclosed in quotation marks and can be up to 32 characters. The default is the value that is specified in the OTHER= option. The LOTHER= option is applicable only when a BARLEGEND= variable is specified.

MARKERS
requests that the points on the cumulative percentage curve be plotted with markers in ODS Graphics output. You can use a SYMBOL statement to plot the points in traditional graphics output.

MAXCMPCT=*percent*
requests that only the Pareto categories that have the highest frequency counts be displayed, where the sum of their corresponding percentages is less than or equal to *percent*. For example, if you specify the following statements, the chart displays only the most frequently occurring categories that account for no more than 90% of the total frequency:

```
proc pareto data=failure;
   vbar cause / maxcmpct = 90;
```

You can use the OTHER= option in conjunction with the MAXCMPCT= option to create and display a new category that combines categories that are not selected by the MAXCMPCT= option. For example, if you specify the following statements, the chart displays the categories that account for no more than 90% of the total frequency, together with a category labeled "Others" that merges the remaining categories:

```
proc pareto data=failure;
   vbar cause / maxcmpct = 90
                other    = 'Others';
```

The MAXCMPCT= option is an alternative to the MINPCT= and MAXNCAT= options.

MAXNCAT=*n*

requests that only the Pareto categories with the *n* highest frequencies be displayed. For example, if you specify the following statements, the chart displays only the categories that have the 20 highest frequencies:

```
proc pareto data=failure;
   vbar cause / maxncat = 20;
```

If the total number of categories is less than 20, all the categories are displayed.

You can use the OTHER= option in conjunction with the MAXNCAT= option to create and display a new category that combines categories that are not selected by the MAXNCAT= option. For example, if you specify the following statements, the chart displays the categories that have the 19 highest frequencies, together with a category labeled "Others" that merges the remaining categories:

```
proc pareto data=failure;
   vbar cause / maxncat = 20
                other= 'Others';
```

See Figure 16.6 for another illustration.

The MAXNCAT= option is an alternative to the MINPCT= and MAXCMPCT= options.

MINPCT=*percent*

requests that only the Pareto categories whose frequency percentages are greater than or equal to *percent* be displayed. For example, if you specify the following statements, the chart displays only categories that have at least 5% of the total frequency:

```
proc pareto data=failure;
   vbar cause / minpct = 5;
```

You can use the OTHER= option in conjunction with the MINPCT= option to create and display a new category that combines categories that are not selected by the MINPCT= option. The merged category that is created by the OTHER= option is displayed even if its total percentage is less than *percent*. For example, if you specify the following statements, the chart displays the categories whose percentages are greater than or equal to 5%, together with a category labeled "Others" that merges the remaining categories:

```
proc pareto data=failure;
   vbar cause / minpct = 5
                other  = 'Others';
```

The MINPCT= option is an alternative to the MAXNCAT= and MAXCMPCT= options.

MISSING

requests that missing values of the process variable be treated as a Pareto category that is represented with a bar on the chart. If the process variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the process variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify this option, missing values are excluded from the analysis.

MISSING1

requests that missing values of the first CLASS= variable be treated as a level of the CLASS= variable. If the first CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the first CLASS= variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify this option, observations in the DATA= data set for which the first CLASS= variable is missing are excluded from the analysis.

MISSING2

requests that missing values of the second CLASS= variable be treated as a level of the CLASS= variable. If the second CLASS= variable is a character variable, a missing value is defined as a blank internal (unformatted) value. If the second CLASS= variable is numeric, a missing value is defined as any of the SAS missing values. If you do not specify this option, observations in the DATA= data set for which the second CLASS= variable is missing are excluded from the analysis.

NCOLS=*n*

NCOL=*n*

specifies the number of columns in a comparative Pareto chart. You can use this option in conjunction with the NROWS= option. See Output 16.2.3 and Output 16.2.4 for an illustration. By default, NCOLS=1 and NROWS=2 if one CLASS= variable is specified, and NCOLS=2 and NROWS=2 if two CLASS= variables are specified.

NLEGEND

NLEGEND='*label*'

NLEGEND=(*variable***)**

requests a sample size legend and specifies its form. You can specify the following values:

(no argument) requests a sample size legend and specifies its form as *N*=*n*, where *n* is the total count for the Pareto categories. In a comparative Pareto chart, a legend is displayed in each tile, and *n* is the total count for that particular cell. See Output 16.2.1 for an illustration.

'*label*' requests a sample size legend and specifies its form as *label*=*n*, where *n* is the total count for the Pareto categories. The *label* can be up to 32 characters and must be enclosed in quotation marks. For an illustration, see Figure 16.4 or Output 16.1.4.

(*variable***)** requests a sample size legend that is the value of *variable* from the DATA= data set. The formatted length of *variable* cannot exceed 32. If a format is associated with *variable*, then the formatted value is displayed. This option is intended for use with comparative Pareto charts and enables you to display a customized legend inside each tile (this legend does not need to provide a total count). It is assumed that the values of *variable* are identical for all observations in a particular class.

By default, the legend is placed in the upper left corner of the chart. If you specify the NOCURVE option, the legend is placed in the upper right corner of the chart. You can use the CFRAMENLEG= option to frame the sample size legend. No sample size legend is displayed if you do not specify an NLEGEND option.

NOCATLABEL
suppresses the category axis label. This option is useful for avoiding clutter where the meaning of the category axis is apparent from the labels for the Pareto categories. See Output 16.2.2 for an illustration.

NOCHART
suppresses the creation of a Pareto chart. This option is useful when you are simply creating an output data set.

NOCUMLABEL
suppresses the cumulative percentage axis label. This option is useful for avoiding clutter on comparative Pareto charts.

NOCUMTICK
suppresses the cumulative percentage axis label, tick marks, and tick mark labels.

NOCURVE
suppresses the cumulative percentage curve and the cumulative percentage axis. Compare Output 16.2.1 and Output 16.2.2 for an illustration.

NOFREQLABEL
suppresses the frequency axis label.

NOFREQTICK
suppresses the frequency axis label, tick marks, and tick mark labels.

NOHLLEG
suppresses the legend that is generated by the CHIGH(*n*)=, CLOW(*n*)=, PHIGH(*n*)=, and PLOW(*n*)= options.

NOKEYMOVE
suppresses the rearrangement of cells within a comparative Pareto chart that occurs when you use the CLASSKEY= option. By default, the key cell appears in the top left corner of a comparative Pareto chart.

NROWS=*n*
NROW=*n*
specifies the number of rows in a comparative Pareto chart. You can use the NROWS= option in conjunction with the NCOLS= option. See Output 16.2.3 and Output 16.2.4 for an illustration. By default, NROWS=2.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string***'**
adds a footnote to ODS Graphics output. You can specify the following values:

FOOTNOTE (or **FOOTNOTE1**) uses the value of the SAS FOOTNOTE statement as the graph footnote.

'*string*' uses *string* as the footnote. The quoted *string* can contain either of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the process variable has no label).

ODSFOOTNOTE2=FOOTNOTE2 | '*string***'**

adds a secondary footnote to ODS Graphics output. You can specify the following values:

FOOTNOTE2 uses the value of the SAS FOOTNOTE2 statement as the secondary graph footnote.

'*string*' uses *string* as the secondary footnote. The quoted *string* can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the process variable has no label).

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | LABELFMT | '*string***'**

specifies a title for ODS Graphics output. You can specify the following values:

TITLE (or **TITLE1**) uses the value of the SAS TITLE statement as the graph title.

NONE suppresses all titles from the graph.

DEFAULT uses the default ODS Graphics title (a descriptive title that consists of the plot type and the process variable name).

LABELFMT uses the default ODS Graphics title, but substitutes the process variable label for the process variable name.

'*string*' uses *string* as the graph title. The quoted *string* can contain the following escaped characters, which are replaced with the appropriate values from the analysis:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the process variable has no label).

ODSTITLE2=TITLE2 | '*string***'**

specifies a secondary title for ODS Graphics output. You can specify the following values:

TITLE2 uses the value of the SAS TITLE2 statement as the secondary graph title.

'*string*' uses *string* as the graph title. The quoted *string* can contain the following escaped characters, which are replaced with the appropriate values from the analysis:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the process variable has no label).

ORDER1=DATA | FORMATTED | FREQ | INTERNAL

specifies the display order for the values of the first CLASS= variable. The levels of the first CLASS= variable are always constructed using the formatted values of the variable, and the formatted values are always used to label the rows (columns) of a comparative Pareto chart. You can specify the following values:

DATA displays the rows (columns) from top to bottom (left to right) in the order in which the values of the first CLASS= variable first appear in the input data set.

FORMATTED displays the rows (columns) from top to bottom (left to right) in increasing order of the formatted values of the first CLASS= variable. For example, suppose you use a numeric CLASS= variable called Day (with values 1, 2, and 3) to create a one-way comparative Pareto chart. Also suppose you use the FORMAT procedure to associate the formatted values 1 = 'Wednesday', 2 = 'Thursday', and 3 = 'Friday' with Day. If you specify ORDER1=FORMATTED, the rows appear in alphabetical order (`'Friday'`, `'Thursday'`, `'Wednesday'`) from top to bottom.

FREQ displays the rows (columns) from top to bottom (left to right) in order of decreasing frequency count. If two or more classes have the same frequency count, the order is determined by the formatted values.

INTERNAL displays the rows (columns) from top to bottom (left to right) in increasing order of the internal (unformatted) values of the first CLASS= variable. If there are two or more distinct internal values that have the same formatted value, the order is determined by the internal value that occurs first in the input data set. In the previous example with variable Day, if you specify ORDER1=INTERNAL, the rows of the comparative chart appear in chronological order (`'Wednesday'`, `'Thursday'`, `'Friday'`) from top to bottom.

By default, ORDER1=INTERNAL.

ORDER2=INTERNAL | FORMATTED | DATA | FREQ

specifies the display order for the values of the second CLASS= variable. The levels of the second CLASS= variable are always constructed using the formatted values of the variable, and the formatted values are always used to label the columns of a two-way comparative Pareto chart.

The PARETO procedure determines the layout of a two-way comparative Pareto chart by first using the ORDER1= option to obtain the order of the rows from top to bottom (recall that ORDER1=INTERNAL by default). Then the ORDER2= option is applied to the observations that correspond to the first row to obtain the order of the columns from left to right. If any columns remain unordered (that is, the categories are unbalanced), the ORDER2= option is applied to the observations in the second row, and so on until all the columns have been ordered.

The values of the ORDER2= option are interpreted as described for the ORDER1= option. By default, ORDER2=INTERNAL.

OTHER='*category*'

specifies a new category that merges all categories that are not selected in the MAXNCAT=, MINPCT=, or MAXCMPCT= options. See the section "Restricting the Number of Pareto Categories" on page 1074 for an illustration.

The *category* should be specified as a formatted value of the process variable. The *category* can be up to 32 characters and must be enclosed in quotation marks. If you specify an OUT= data set, you

should also specify an internal value that corresponds to *category* by specifying the OTHERCVAL= option or the OTHERNVAL= option.

The OTHER= option is not applicable unless you specify the MAXNCAT=, MINPCT=, or MAXCMPCT= option. You can use the COTHER=, LOTHER=, POTHER=, OTHERCVAL=, and OTHERNVAL= options with the OTHER= option.

OTHERCVAL='*value*'
specifies the internal (unformatted) *value* for a character process variable in the OUT= data set that corresponds to the category that is specified in the OTHER= option. The *value* can be up to 64 characters and must be enclosed in quotation marks.

The OTHERCVAL= option is not applicable unless you specify the OTHER= and OUT= options. If you specify the OTHER= option but not the OTHERCVAL= option, the value specified in the OTHER= option is written to the OUT= data set.

OTHERNVAL=*value*
specifies the internal (unformatted) *value* for a numeric process variable in the OUT= data set that corresponds to the category that is specified in the OTHER= option. The OTHERNVAL= option is not applicable unless you specify the OTHER= and OUT= options. If you specify the OTHER= option but not the OTHERNVAL= option, a missing value is written to the OUT= data set.

OUT=*SAS-data-set*
creates an output data set that contains the information that is displayed in the Pareto chart. This data set is useful if you want to create a report to accompany your chart. See Example 16.8 for an illustration.

SCALE=COUNT | FREQUENCY | PERCENT | WEIGHT
specifies the scale for the frequency axis. You can specify the following values:

COUNT or **FREQUENCY** specifies that the scale is counts. See Output 16.1.4 for an illustration. This option is ignored if you specify the WEIGHT= option.

PERCENT specifies that the scale is the percentage of the total frequency or, if you specify the WEIGHT= option, the percentage of the total weight.

WEIGHT scales the vertical axis in the same units as the variable you specify in the WEIGHT= option. This option applies only if you specify the WEIGHT= option.

By default, SCALE=PERCENT. See Output 16.8.1 for an example.

NOTE: Regardless of the value you specify for the SCALE= option, the cumulative percentage axis is scaled in cumulative percentage units.

URL=*variable*
specifies URLs as values of the specified character *variable* (or formatted values of a numeric *variable*). These URLs are associated with bars on the Pareto chart when ODS Graphics output is directed into HTML. The value of *variable* should be the same for each observation that has a particular value of the process variable. The URL= option is not supported for traditional graphics output.

VREFLABPOS=*n*

specifies the vertical positioning of the labels for reference lines that are associated with vertical axes, which are specified in the CATREF= option in an HBAR statement or in the FREQREF= and CUMREF= options in a VBAR statement. If you specify VREFLABPOS=1, the labels are positioned at the left of the chart; if you specify VREFLABPOS=2, the labels are positioned at the right. By default, VREFLABPOS=1.

WEIGHT=*variable-list*

specifies weight variables that are used to construct weighted Pareto charts. Variables in the *variable-list* are paired with the process variables in order of specification. The WEIGHT= variables must be numeric, and their values must be nonnegative (noninteger values are permitted). If a WEIGHT= variable is not provided for a process variable, the weights applied to that process variable are assumed to be 1. See "Weighted Pareto Charts" on page 1118 for computational details.

A WEIGHT= variable is particularly useful for carrying out a Pareto analysis based on cost rather than frequency of occurrence. See Example 16.8 for an illustration.

Options for Traditional Graphics

You can specify the following options only when traditional graphics are produced. The PARETO procedure produces traditional graphics when ODS Graphics is disabled and SAS/GRAPH is licensed.

ANGLE=*value*

specifies an angle in degrees for rotating the labels on the category axis. The *value* is the angle between the baseline of the label and the category axis. See Output 16.1.1 and Output 16.1.2 for an illustration. The *value* must be greater than or equal to –90 and less than 90. The default value is 0.

ANNOKEY

applies the annotation requested in the ANNOTATE= and ANNOTATE2= options only to the key cell in a comparative Pareto chart. By default, annotation is applied to all of the cells.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains annotation variables as described in *SAS/GRAPH: Reference*. You can use the *SAS-data-set* to customize the Pareto charts that are produced by a single HBAR or VBAR statement. (A data set that is specified in the ANNOTATE= option in the PROC PARETO statement customizes charts that are produced by *all* HBAR and VBAR charts.) The *SAS-data-set* is associated with the frequency axis. If the annotation is based on data coordinates, you must use the same units as the frequency axis.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an input data set that contains annotation variables as described in *SAS/GRAPH: Reference*. You can use the *SAS-data-set* to customize the Pareto charts that are produced by a single HBAR or VBAR statement. (A data set that is specified in the ANNOTATE2= option in the PROC PARETO statement customizes charts that are produced by *all* HBAR and VBAR charts.) The *SAS-data-set* is associated with the cumulative percentage axis. If the annotation is based on data coordinates, you must use the same units as the cumulative percentage axis.

BARLABPOS=*keyword*

specifies the position for labels that are requested in the BARLABEL= option.

You can specify the following *keywords* in an HBAR statement:

HBAR displays the label right-justified on the bar. If the label is longer than the bar, it is left-justified at the base of the bar.

HFIT right-justifies the label on the bar. If the label is longer than the bar, the label is displayed to the right of the bar.

HLJUST left-justifies the label at the base of the bar.

HRIGHT displays the label to the right of the bar. If there is insufficient space for the label to the right of the bar, the label is right-justified at the right edge of the frame.

HRJUST right-justifies the label at the right edge of the frame.

The default for an HBAR statement is BARLABPOS=HRIGHT.

You can specify the following *keywords* in a VBAR statement:

HCENTER centers the label horizontally above the bar. If the centered label would extend outside the frame, the label is left-justified or right-justified at the edge of the frame.

HLJUST left-justifies the label horizontally above the bar. The label is truncated if necessary.

VBAR displays the label vertically on the bar. If the label is longer than the bar, it extends above the bar.

VFIT displays the label vertically on or above the bar, depending on the available space. If the label is longer than the bar, it is displayed just below the top edge of the frame.

The default for a VBAR statement is to center the labels horizontally above the bars, with a reduction in text height if necessary. Reduction is not applied when the BARLABPOS= option is specified.

BARWIDTH=*value*

specifies the width of the bars in screen percentage units. By default, the bars are made as wide as possible.

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color for the axis lines and tick marks. The default color is specified by the ContrastColor attribute of the GraphAxisLines style element in the current ODS style. If the NOGSTYLE option is in effect, *color* is also used for bar outlines and grid lines, unless overridden by the CBARLINE=, CGRID=, or GRID2= option.

CAXIS2=*color*

specifies the color for the tick mark labels and axis label that are associated with the cumulative percentage axis. By default, the color specified in the CTEXT= option (or its default) is used.

CBARLINE=*color*

specifies the color for bar outlines. The default color is specified by the ContrastColor attribute of the GraphOutlines style element in the current ODS style.

CBARS=*color*

CBARS=(*variable-list***)**

specifies how the bars of the Pareto chart are colored. You can specify the following values:

color uses a single color for all the bars. You can use this option in conjunction with the CHIGH(*n*) and CLOW(*n*) options.

variable-list uses a distinct color for each bar (or combination of bars). The colors are specified as values of variables in the *variable-list*. Each variable must be a character variable. You can use the special value `'EMPTY'` to indicate that a bar is not to be colored. Note that *variable-list* must be enclosed in parentheses. You cannot specify a *variable-list* conjunction with the CHIGH(*n*) or CLOW(*n*) option.

If you specify more than one process variable, you can specify more than one CBARS= variable. The number of CBARS= variables should be less than or equal to the number of process variables. The two lists of variables are paired in order of specification.

If no CBARS= color or variable is specified for a process variable, the bars for its chart are displayed in the default color, which is determined by the Color attribute of the GraphData1 style element in the current ODS style.

If you specify one or more CBARS= variables, you can also use the BARLEGEND= option to add a legend to the chart that explains the significance of each color. Furthermore, you can use the PBARS= option to specify patterns in conjunction with the CBARS= option.

CCATREF=*color*

specifies the color for reference lines that are requested in the CATREF= option. The default color is specified by the ContrastColor attribute of the GraphReference style element in the current ODS style.

CCONNECT=*color*

specifies the color for the line segments that connect the points on the cumulative percentage curve. The default color is determined by the ContrastColor attribute of the GraphDataDefault style element in the current ODS style. You can specify the color for the points on the cumulative percentage curve in SYMBOL statement COLOR= option.

CCUMREF=*color*

specifies the color for reference lines that are requested in the CUMREF= option. The default color is specified by the ContrastColor attribute of the GraphReference style element in the current ODS style.

CFRAME=*color*

specifies the color for filling the area that is enclosed by the axes and the frame. The default color is specified by the Color attribute of the GraphWalls style element in the current ODS style. You cannot use the CFRAME= option in conjunction with the NOFRAME option or the CTILES= option.

CFRAMESIDE=*color*

specifies the color for filling the frame area for the row labels, which are displayed along the left side of a comparative Pareto chart. If a label is associated with the classification variable, *color* is also used to fill the frame area for this label. By default, the frame is transparent.

CFRAMETOP=*color*
specifies the color for filling the frame area for the column labels, which are displayed across the top of a comparative Pareto chart. If a label is associated with the classification variable, *color* is also used to fill the frame area for this label. By default, the frame is transparent.

CFREQREF=*color*
specifies the color for reference lines that are requested in the FREQREF= option. The default color is specified by the ContrastColor attribute of the GraphReference style element in the current ODS style.

CGRID=*color*
specifies the color for frequency axis grid lines. If you specify this option, you do not need to specify the GRID option. The default color is specified by the ContrastColor attribute of the GraphGridLines style element in the current ODS style.

CGRID2=*color*
specifies the color for cumulative percentage axis grid lines. If you specify this option, you do not need to specify the GRID2 option. The default color is specified by the ContrastColor attribute of the GraphGridLines style element in the current ODS style.

CLIPREF
draws reference lines that are requested in the CATREF=, CUMREF=, and FREQREF= options behind the bars on the Pareto chart. When the GSTYLE option is in effect, reference lines are drawn in front of the bars by default.

COTHER=*color*
specifies the color for the bar that is defined by the OTHER= option. By default the CFRAME= color is used. The COTHER= option is not applicable unless a BARS= or CBARS= variable is specified.

CTEXT=*color*

CT=*color*
specifies the color for text, such as tick mark labels, axis labels, and legends. The default color is specified by the Color attribute of a style element in the current ODS style. Axis labels use the GraphLabelText style element, and all other text uses the GraphValueText style element.

CTEXTSIDE=*color*
specifies the color for row labels, which are displayed along the left side of a comparative Pareto chart. If you do not specify a *color*, the color specified in the CTEXT= option is used. If neither option is specified, the color is determined by the Color attribute of the GraphValueText style element in the current ODS style.

CTEXTTOP=*color*
specifies the color for column labels, which are displayed across the top of a comparative Pareto chart. If you do not specify a *color*, the color specified in the CTEXT= option is used. If neither option is specified, the color is determined by the Color attribute of the GraphValueText style element in the current ODS style.

CTILES=(*variable***)**
specifies a character variable whose values are the fill colors for the tiles in a comparative Pareto chart. This option generalizes the CFRAME= option, which provides a single color for all of the tiles. The *variable* must be enclosed in parentheses. The values of the *variable* must be identical for all

observations that have the same level of the CLASS= variables. You can use the same color to fill more than one tile. You can use the special value `'EMPTY'` to indicate that a tile is not to be filled.

You cannot use the CTILES= option in conjunction with the NOFRAME or CFRAME= options. You can use the TILELEGEND= option in conjunction with the CTILES= option to add an explanatory legend for the CTILES= colors at the bottom of the chart. See Output 16.5.1 for an illustration.

DESCRIPTION='*string*'

DES='*string*'

specifies a description, up to 256 characters long, for the GRSEG catalog entry for a traditional graphics chart.

FONT=*font*

specifies a font for text that is used in labels and legends. The default font is determined by the FontFamily, FontStyle, and FontWeight attributes of a style element in the current ODS style; axis labels use the GraphLabelText style element and all other text uses the GraphValueText style element.

FRONTREF

draws reference lines that are requested in the CATREF=, FREQREF=, and CUMREF= options in front of the bars on the Pareto chart. When the NOGSTYLE option is in effect, reference lines are drawn behind the bars by default and can be obscured by them.

HEIGHT=*value*

specifies the height in screen percentage units of text for labels and legends. This option takes precedence over the GOPTONS HTEXT= option. The default value is specified by the FontSize attribute of the a style element in the current ODS style; axis labels use the GraphLabelText style element and all other text uses the GraphValueText style element.

HTML=*variable*

specifies a variable whose values create links that are associated with Pareto bars when traditional graphics output is directed into HTML. You can specify a character variable or a formatted numeric variable. The value of the HTML= variable should be the same for each observation that has a particular value of the process variable.

INFONT=*font*

specifies a font for bar labels, cumulative percentage curve labels, and sample size legends. This option takes precedence over the FONT= option and the FTEXT= option in the GOPTIONS statement. The default font is determined by the FontFamily, FontStyle, and FontWeight attributes of the GraphValueText style element in the current ODS style.

INHEIGHT=*value*

specifies the height in screen percentage units of bar labels, cumulative percentage curve labels, and sample size legends. This option takes precedence over the HEIGHT= option and the HTEXT= option in a GOPTIONS statement. The default value is specified by the FontSize attribute of the GraphValueText style element in the current ODS style.

INTERBAR=*value*

specifies the distance in screen percentage units between bars on the chart. By default, the bars are contiguous.

LCATREF=*line-type*
specifies the line type for reference lines that are requested in the CATREF= option. The default line type is specified by the LineStyle attribute of the GraphReference style element in the current ODS style.

LCUMREF=*line-type*
specifies the line type for reference lines that are requested in the CUMREF= option. The default line type is specified by the LineStyle attribute of the GraphReference style element in the current ODS style.

LFREQREF=*line-type*
specifies the line type for lines that are requested in the FREQREF= option. The default line type is specified by the LineStyle attribute of the GraphReference style element in the current ODS style.

LGRID=*line-type*
specifies the line type for frequency axis grid lines. If you specify this option, you do not need to specify the GRID option. The default line type is specified by the LineStyle attribute of the GraphGridLines style element in the current ODS style.

LGRID2=*line-type*
specifies the line type for cumulative percentage axis grid lines. If you specify this option, you do not need to specify the GRID2 option. The default line type is specified by the LineStyle attribute of the GraphGridLines style element in the current ODS style.

NAME='*string*'
specifies the name of the GRSEG catalog entry for a traditional graphics chart, and the name of the graphics output file if one is created. The name can be up to 256 characters long, but the GRSEG name is truncated to eight characters. The default name is "PARETO".

NOFRAME
suppresses the frame that is drawn around the chart by default. You cannot specify the NOFRAME option in conjunction with the CFRAME= or TILES= options.

PBARS=*pattern*
PBARS=(*variable-list***)**
specifies pattern fills for the bars. You can specify the following values:

pattern uses a single pattern for all the bars. You can use this approach in conjunction with the PHIGH(*n*)= and PLOW(*n*)= options.

variable-list uses a distinct pattern for *each* bar (or combination of bars). You provide the patterns as values of variables in the *variable-list*. For example, you might use the solid pattern ('S') to indicate severe problems and the empty pattern ('E') for all other problems. Each variable must be a character variable of length eight, and the *variable-list* must be enclosed in parentheses. You cannot specify a *variable-list* in conjunction with the PHIGH(*n*)= and PLOW(*n*)= options.

If you specify more than one process variable in the chart statement, you can provide more than one variable in the *variable-list*. The number of variables in the *variable-list* should be less than or equal to the number of process variables. The two lists of variables are paired in order of specification. If a variable is not

provided in the *variable-list* for a process variable, the bars for that chart are not filled.

If you specify a *variable-list*, you can also use the BARLEGEND= option to add a legend to the chart that explains the significance of each pattern.

You can use the CBARS= option to specify colors in conjunction with the PBARS= option.

PHIGH(*n*)=*pattern*
specifies the pattern for the bars that have the *n* highest values. You cannot specify this option in conjunction with a PBARS=*variable-list*, but you can specify this option together with the PLOW(*n*)= and PBARS=*pattern* options.

PLOW(*n*)=*pattern*
specifies the pattern for the bars that have the *n* lowest values. You cannot specify this option in conjunction with a PBARS=*variable-list*, but you can use this option together with the PHIGH(*n*)= and PBARS=*pattern* options.

POTHER=*pattern*
specifies the pattern for the bar that is defined by the OTHER= option. This option applies only if you specify a PBARS=*variable-list*.

TILELEGEND=(*variable*)
specifies a *variable* that is used to add a legend for CTILES= colors. The variable can have a formatted length less than or equal to 32. If a format is associated with the variable, then the formatted value is displayed. You must specify the TILELEGEND= option in conjunction with the CTILES= option. If you specify the CTILES= option but do not specify the TILELEGEND= option, a color legend is not displayed.

The values of the CTILES= and TILELEGEND= variables should be consistent for all observations that have the same level of the CLASS= variables. The value of the TILELEGEND= variable is used to identify the corresponding color value of the CTILES= variable in the legend. See Output 16.5.1 for an illustration.

TILELEGLABEL='*label*'
specifies a label for the legend that is created when you specify a TILELEGEND= variable. The *label* can be up to 16 characters and must be enclosed in quotation marks. The default is "Tiles:". See Output 16.5.1 for an illustration.

TURNVLABEL
TURNVLABELS
turns and strings out vertically the characters in the labels for the frequency and cumulative percentage axes. The TURNVLABELS option is valid only in a VBAR statement.

WAXIS=*n*
specifies the line thickness (in pixels) for the axes and frame. This thickness is also used for bar outlines and grid lines, unless overridden by the WBARLINE=, WGRID=, or WGRID2= option. The default line thickness is specified by the LineThickness attribute of the GraphAxisLines style element in the current ODS style.

WBARLINE=*n*
specifies the width for bar outlines. The default outline thickness is specified by the LineThickness attribute of the GraphOutlines style element in the current ODS style.

WGRID=*n*
specifies the width of the frequency axis grid lines. If you specify this option, the GRID option is not required. The default line thickness is specified by the LineThickness attribute of the GraphGridLines style element in the current ODS style.

WGRID2=*n*
specifies the width of the cumulative percentage axis grid lines. If you specify this option, the GRID2 option is not required. The default line thickness is specified by the LineThickness attribute of the GraphGridLines style element in the current ODS style.

Options for Legacy Line Printer Charts

NOTE: The HBAR statement does not produce legacy line printer charts, so the following *options* apply only to the VBAR statement.

CONNECTCHAR='*character*'
CCHAR='*character*'
specifies the plot character for line segments that connect points on the cumulative percentage curve. The default character is a plus sign (+).

HREFCHAR='*character*'
specifies the plot character used to form the lines that are requested in the CATREF= option. The default character is a vertical bar (|).

SYMBOLCHAR='*character*'
specifies the plot character for points on the cumulative percentage curve. The default character is an asterisk (*).

VREFCHAR='*character*'
specifies the character to be used to form the lines that are requested in the FREQREF= and CUMREF= options. The default character is a dash (-).

Details: PARETO Procedure

Terminology

Basic Pareto Charts

A basic Pareto chart (see Figure 16.1) analyzes the unique values of a *process variable*. These values are called *Pareto categories* or *levels*, and they usually represent problems that are encountered during some phase of a manufacturing or service activity.

A basic vertical Pareto chart (as produced by the PARETO procedure's VBAR statement) has one horizontal axis and two vertical axes:

- The *category* axis is displayed horizontally at the bottom of the chart and lists the Pareto categories.

- The *frequency axis* (or *primary vertical axis*) is displayed on the left. The relative frequency of each Pareto category is represented by a vertical bar whose height is measured on the frequency axis. You can use the SCALE= option to scale this axis in percentage, count, or weight units.
- The *cumulative percentage axis* (or *secondary vertical axis*) is displayed on the right. This axis is scaled in cumulative percentage units and is used to read the *cumulative percentage curve*. The height of each point on the curve represents the percentage of the total frequency that is accounted for by the Pareto categories to the left of the point.

A horizontal Pareto chart (as produced by the HBAR statement), is essentially a vertical Pareto chart rotated 90 degrees clockwise. The category axis is displayed vertically on the left. Categories appear in order of decreasing relative frequency from top to bottom. The frequency axis appears at the top of the chart and the cumulative percentage axis appears at the bottom. The relative frequencies of the Pareto categories are represented by horizontal bars. A point on the cumulative percentage curve represents the percentage of the total frequency that is accounted for by the Pareto categories above that point.

NOTE: For the sake of brevity, in this chapter the term *height* refers to the size of a bar as measured along the frequency axis, whether the Pareto chart is oriented vertically or horizontally.

Restricted Pareto Charts

A *restricted Pareto chart* (see Figure 16.6) displays only the n most frequently occurring categories in a data set that contains N categories, where $N > n$. The remaining $N - n$ categories are dropped or are merged into a single "other" category that is created when you specify the OTHER= option. The MAXCMPCT=, MAXNCAT=, and MINPCT= options provide alternative methods for specifying n. See the entries for these options in the section "Dictionary of HBAR and VBAR Statement Options" on page 1095.

Weighted Pareto Charts

A *weighted Pareto chart* (see Example 16.8) displays bars whose heights represent the weighted frequencies of the categories. Typical weights are the cost of repair or the loss incurred by the customer.

The weight W_i for the ith Pareto category is computed as

$$W_i = \sum_{u \in C_i} w(u) f(u)$$

where C_i is the set of observations that make up the ith category, $w(u)$ is the value of the weight variable in the uth observation, and $f(u)$ is the value of the frequency variable in the uth observation (taking $f(u) \equiv 1$ if a FREQ= variable is not specified). If SCALE=WEIGHT is specified, the height of the bar for the ith category is W_i. If SCALE=PERCENT is specified, the height of this bar is

$$\frac{100 W_i}{\sum_{j=1}^{N} W_j}$$

where N is the total number of categories.

Comparative Pareto Charts

A *comparative Pareto chart* combines two or more Pareto charts for the same process variable. The component charts are displayed with uniform axes to facilitate comparison. The observations that are represented by a component chart are called a *cell*. The framed areas for the component charts are called *tiles*.

In a *one-way comparative Pareto chart*, each component chart corresponds to a different level of a single classification variable, which is specified in the CLASS= option. The component charts are arranged in a stack or a row, as illustrated in Output 16.1.3, Output 16.1.4, Output 16.2.2, and Output 16.2.3. In a *two-way comparative Pareto chart*, each component chart corresponds to a different combination of levels of two classification variables, which are specified in the CLASS= option. The component charts are arranged in a matrix, as illustrated in Output 16.2.4.

Every comparative Pareto chart has a *key cell*, in which the bars are in decreasing order and whose order is imposed on all the other cells to achieve a uniform category axis. By default, the key cell is the cell in the upper left corner, but you can use the CLASSKEY= option to designate any other cell as the key cell. If you designate another cell as the key cell, the rows and columns of the comparative chart are rearranged so that the key cell appears in the upper left. However, if you require the rows and columns in a particular order, you can specify the NOKEYMOVE option in conjunction with the CLASSKEY= option to suppress the rearrangement.

You can use the NROWS= and NCOLS= options to specify the numbers of rows and columns in a comparative Pareto chart. By default, NROWS=2 and NCOLS=1 for a one-way comparison and NROWS=2 and NCOLS=2 for a two-way comparison. There is no upper limit to the number of rows or columns that you can specify, but in practice the limit is determined by the area of the graphical display. If the numbers of classification variable levels exceed the NROWS= and NCOLS= values, the chart is created on multiple panels or pages.

If the same set of Pareto categories does not occur in each cell of a comparative Pareto chart, the categories are said to be *unbalanced*. In this case, PROC PARETO uses the following convention to construct the uniform category axis. First, the categories that occur in the key cell are arranged on the category axis from left to right (top to bottom for a horizontal chart) and sorted in decreasing order of frequency, with tied levels arranged in order of their formatted values. The categories not in the key cell are assigned frequencies of 0 in the key cell, and they are arranged at the right (bottom) of the category axis, where they are ordered by their formatted values. This arrangement is simply a convention of the PARETO procedure and should not be interpreted to mean that one category is more important than another.

Whether the categories in the input data set are balanced or not, the categories in the OUT= data set are always balanced. PROC PARETO balances this data set by assigning values of 0 to the _COUNT_ and _PCT_ variables as necessary.

Unbalanced categories present a special problem when the MAXNCAT= option is used to restrict the number of categories that are displayed on the chart. For example, suppose that you specify MAXNCAT=12 and there are 15 categories in all, 10 of which occur in the key cell. Because there is no unambiguous method for selecting two of the remaining five categories to complete the restricted list, the PARETO procedure reduces the restricted list to the categories that occur in the key cell and displays only those 10 categories. A warning message is issued in the SAS log.

Labels for Chart Features

Table 16.8 summarizes the methods for labeling the features of Pareto charts.

Table 16.8 Labeling Features of Pareto Charts

Feature	Method for Specifying Label
Titles	TITLE*n* statements, ODSTITLE= option, ODSTITLE2= option
Footnotes	FOOTNOTE*n* statements, ODSFOOTNOTE= option, ODSFOOTNOTE2= option
Category axis	Process variable label
Frequency axis	FREQAXISLABEL= option
Cumulative percentage axis	CUMAXISLABEL= option
Bars	BARLABEL= option
Points on cumulative percentage curve	CMPCTLABEL= option
Rows and columns	CLASS= variable labels
Cells	NLEGEND option or NLEGEND= variable
Category legend	CATLEGLABEL= option
High/low bar legend	HLLEGLABEL= option
Bar color legend	BARLEGLABEL= option
Tile legend	TILELEGLABEL= option
Annotation	ANNOTATE= and ANNOTATE2= data sets

Scaling the Cumulative Percentage Curve

Pareto charts shown in textbooks usually scale the cumulative percentage curve so that it is anchored at the top right corner of the leftmost bar. The upper end of the frequency axis is then extended to accommodate the curve. For an illustration, see Figure 16.1. By default, the PARETO procedure uses the top right corner as the anchor position on a vertical chart and the bottom right corner of the topmost bar as the anchor position on a horizontal chart. You can override the default by specifying the ANCHOR= option.

This method of scaling is not feasible if the number of categories is very large and if the Pareto distribution is uniform. In this case, the bars are excessively compressed relative to the curve. Conversely, this method excessively compresses the curve relative to the bars when you use a count scale for the frequency axis in a comparative Pareto chart and the tallest bar does not occur in the key cell. In either situation, PROC PARETO overrides the textbook scaling method and balances the scales of the bars and the curve.

You can use the AXISFACTOR= option to specify the extent to which the frequency axis should be extended. Alternatively, you can extend the frequency axis by using the FREQAXIS= option to specify the tick mark values for the axis.

Another scaling anomaly is illustrated by the comparative Pareto chart in Output 16.1.4. There, the cumulative percentage curve in the bottom chart is not anchored because a uniform count scale is combined with different sample sizes in the two cells.

Positioning Insets

This section provides details about three different methods of positioning insets using the POSITION= option. You can use the POSITION= option to specify the following:

- compass points
- keywords for margin positions
- coordinates in data units or percentage axis units

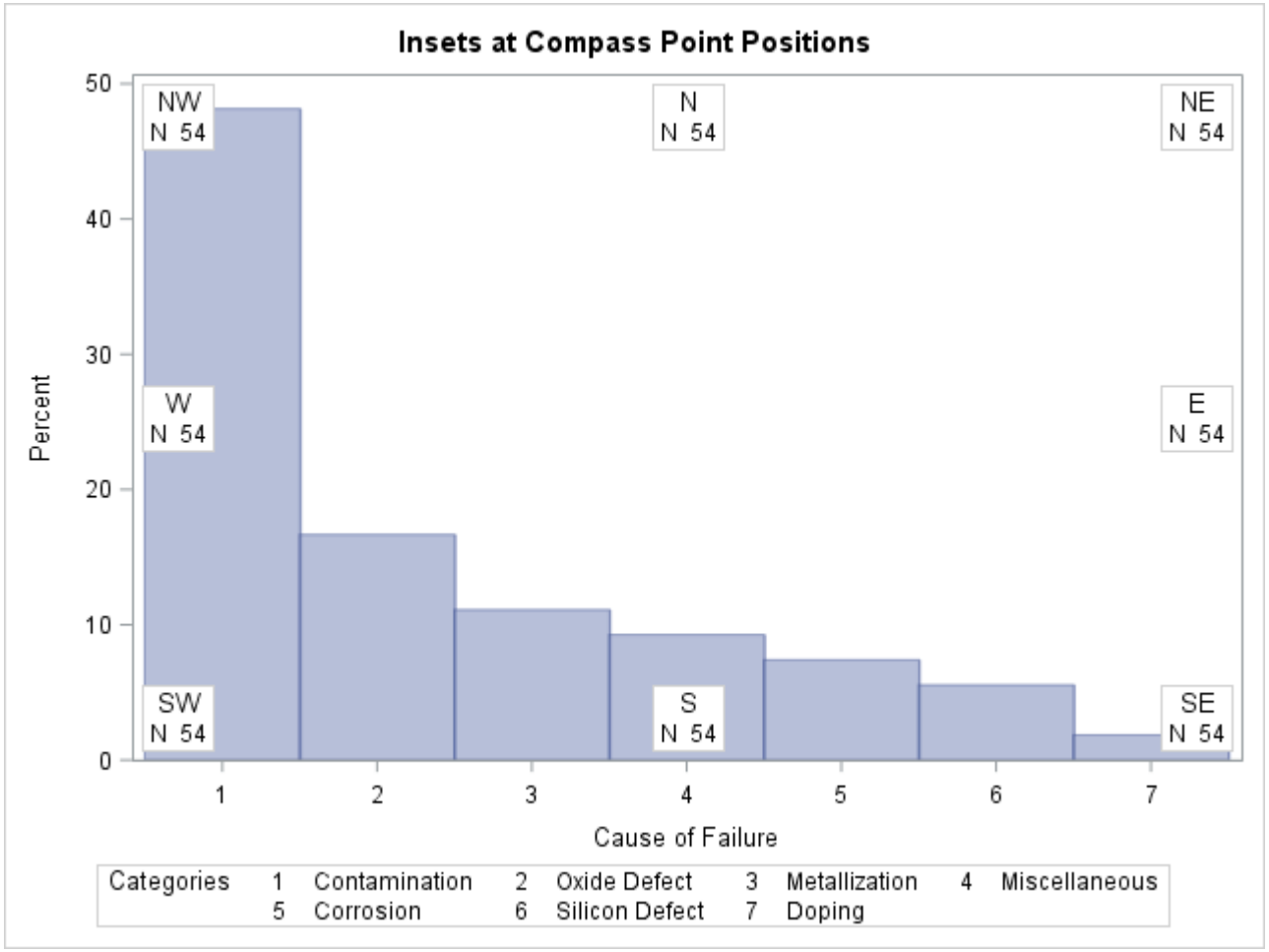
Using Compass Points to Position Insets

NOTE: See *Positioning Insets in Pareto Charts* in the SAS/QC Sample Library.

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 16.9, which demonstrates all eight compass positions. The default is NW.

```
proc pareto data=Failure3;
   vbar Cause / freq     = Counts
                odstitle = "Insets at Compass Point Positions"
                nocurve
                ;
   inset n / cfill header='NW' pos=nw;
   inset n / cfill header='N ' pos=n ;
   inset n / cfill header='NE' pos=ne;
   inset n / cfill header='E ' pos=e ;
   inset n / cfill header='SE' pos=se;
   inset n / cfill header='S ' pos=s ;
   inset n / cfill header='SW' pos=sw;
   inset n / cfill header='W ' pos=w ;
run;
```

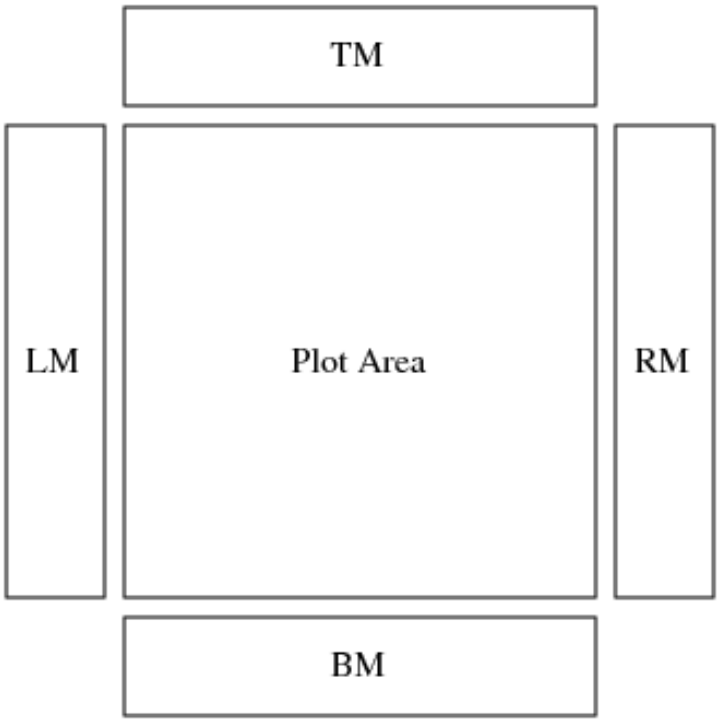
Figure 16.9 Using Compass Points to Position Insets



Positioning Insets in the Margins

You can also use the margin keywords LM, RM, TM, or BM in the INSET statement to position an inset in one of the four margins that surround the plot area, as illustrated in Figure 16.10.

Figure 16.10 Positioning Insets in the Margins



For an example of an inset placed in the right margin, see Output 16.11.1. You might want to place an inset in a margin if it contains a large number of entries (for example the contents of a data set that is specified in the DATA= keyword). If you attempt to display a lengthy inset in the interior of the plot, the inset is likely to collide with the data display.

Insets that are associated with a comparative Pareto chart cannot be positioned in the margins.

Using Coordinates to Position Insets

When you produce traditional graphics, you can also specify the position of the inset with coordinates by specifying POSITION=(x, y). The coordinates can be specified in axis percentage units (the default) or in axis data units.

Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. Data units along the category axis are based on category numbers. Categories are numbered from left to right (VBAR chart) or top to bottom (HBAR chart), starting with 1.

NOTE: See *Positioning Insets in Pareto Charts* in the SAS/QC Sample Library.

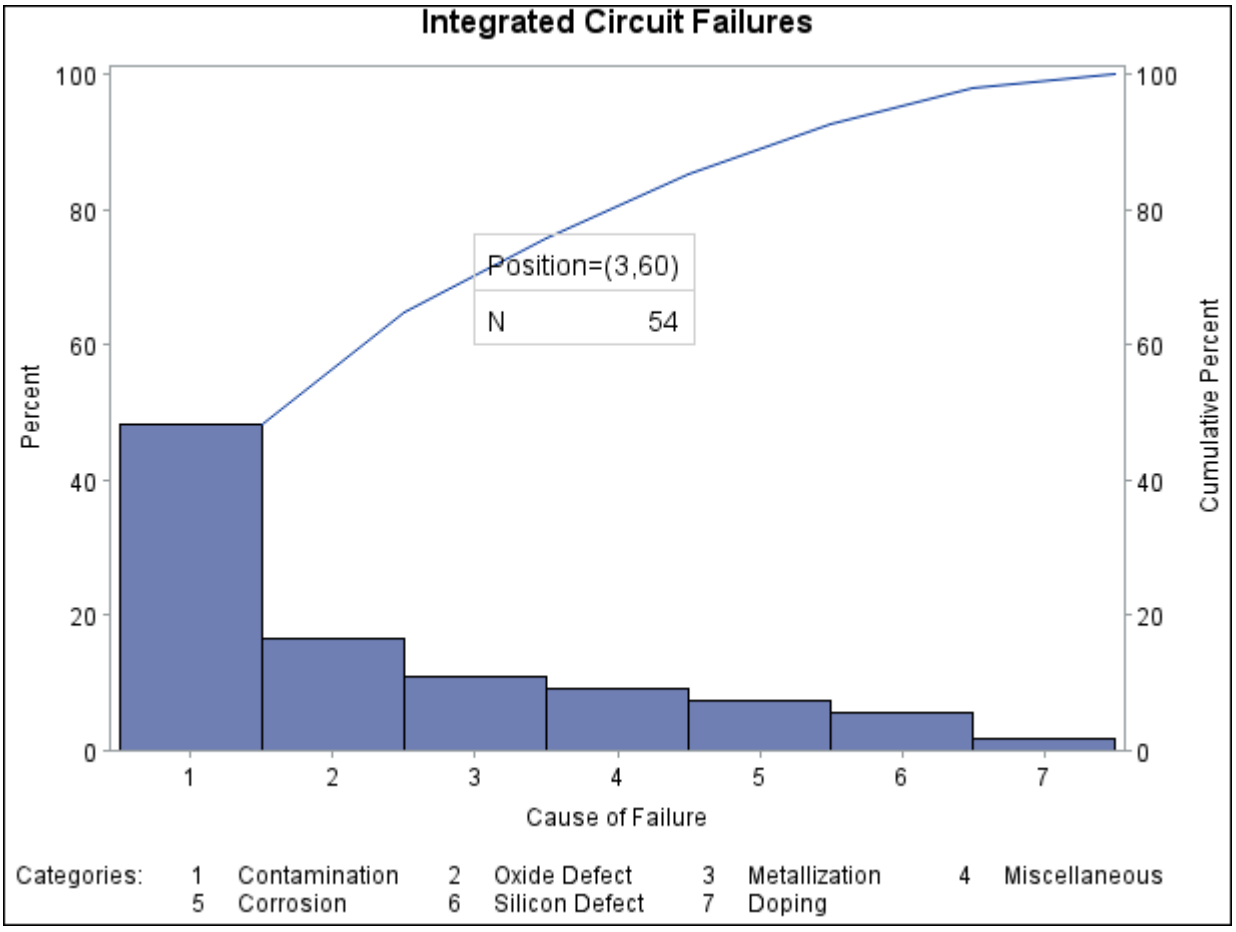
For example, the following statements produce the Pareto chart that is displayed in Figure 16.11:

```
ods graphics off;
title 'Integrated Circuit Failures';
proc pareto data=Failure3;
   vbar Cause / freq = Counts;
   inset n / header   = 'Position=(3,60)'
             position = (3,60) data
             height   = 3;
run;
```

The HEIGHT= option in the INSET statement specifies the text height that is used to display the statistics in the inset.

The bottom left corner of the inset is lined up with the tick mark for the third category on the horizontal axis and at 60 on the vertical axis. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point by specifying the REFPOINT= option, as shown in the next section.

Figure 16.11 Inset Positioned Using Data Unit Coordinates



Axis Percentage Unit Coordinates

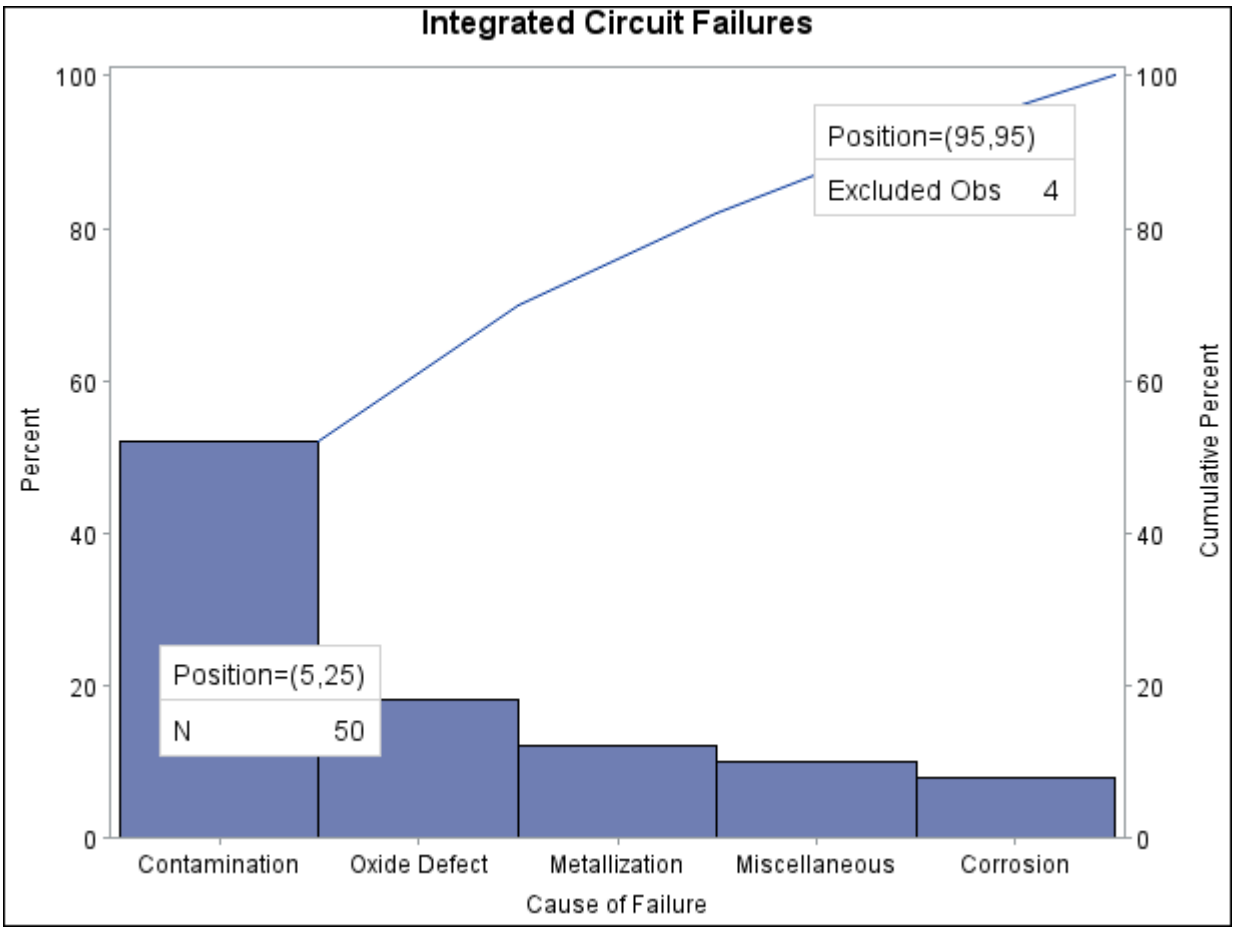
NOTE: See *Positioning Insets in Pareto Charts* in the SAS/QC Sample Library.

If you do not use the DATA option, the inset is positioned using axis percentage units. The coordinates of the bottom left corner of the display are (0, 0), and the coordinates of the upper right corner are (100, 100). For example, the following statements create a Pareto chart that has two insets, both positioned using coordinates in axis percentage units.

```
proc pareto data=Failure3;
   vbar Cause / freq     = Counts
                maxncat  = 5;
   inset n / position = (5,25)
             header   = 'Position=(5,25)'
             height   = 3
             cfill    = blank
             refpoint = tl;
   inset nexcl / position = (95,95)
                 header   = 'Position=(95,95)'
                 height   = 3
                 cfill    = blank
                 refpoint = tr;
run;
```

The chart is shown in Figure 16.12. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates that are specified in the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Coordinates in axis percentage units must be between 0 and 100.

Figure 16.12 Inset Positioned Using Axis Percentage Unit Coordinates



Creating Output Data Sets

The OUT= data set saves the information that is displayed on a Pareto chart. If you specify CLASS= variables, the OUT= data set contains one block of observations for each combination of levels of the CLASS= variables, and each block contains an observation for each Pareto category. The observations are sorted in the order in which the categories are displayed on the chart. The following variables from a DATA= data set are saved in an OUT= data set:

- process variables
- CLASS= variables
- BY variables
- WEIGHT= variables
- the CTILES= variable
- the TILELEGEND= variable

- the NLEGEND= variable
- BARS= or CBARS= variables
- PBARS= variables
- BARLEGEND= variables

In addition, the OUT= data set contains the following variables that are created during the analysis:

- _COUNT_, which saves the frequency count for each Pareto category
- _WCOUNT_, which saves the weighted count for each category. This variable is created only when you specify the WEIGHT= option.
- _PCT_, which saves the percentage of the total count for each category. If you specify the WEIGHT= option, the variable _PCT_ saves the percentage of the total weighted count.
- _CMPCT_, which saves the cumulative percentage for each category

See Output 16.8.2 for an example of an OUT= data set.

If you specify the MAXNCAT=, MAXCMPCT=, or MINPCT= option, the OUT= data set saves only the categories that are displayed on the chart. If you create an OTHER= category that merges the remaining categories, an additional observation is saved with the new category. Because the OTHER= value is defined as a formatted value of the process variable, you should also specify a corresponding internal value, as follows:

- If the process variable is a character variable, specify the internal value in the OTHERCVAL= option. If you do not specify this value, the OTHER= value is saved as the internal value.
- If the process variable is a numeric variable, specify the internal value in the OTHERNVAL= option. If you do not specify this value, an internal missing value is saved.

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of a graph that ODS Graphics produces is determined by the style that is associated with the ODS destination where the graph is produced. HBAR and VBAR statement options that control the appearance of traditional graphics (listed in the section "Options for Traditional Graphics" on page 1110) are ignored for ODS Graphics output.

When ODS Graphics is in effect, the PARETO procedure assigns a name to graphs it creates. You can use this name to refer to the graph when using ODS. The name is listed in Table 16.9.

Table 16.9 ODS Graphics Produced by the PARETO Procedure

ODS Graph Name	Plot Description
ParetoChart	Pareto chart

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Constructing Effective Pareto Charts

The following are recommendations for improving the visual clarity of Pareto charts:

- Decide carefully how the bars should be scaled. The default percentage scale is not always the best choice. For example, a count scale might be more appropriate in a comparative Pareto chart where the total count per cell varies widely from cell to cell and where you want to compare Pareto distributions on an absolute scale rather than a relative scale. You can request a count scale by specifying SCALE=COUNT. In other situations, it might be more appropriate to use a weighted percentage scale or a weighted count scale (specify a WEIGHT= variable and either SCALE=PERCENT or SCALE=WEIGHT).
- Use a weight variable if the counts are dependent on a factor (such as exposure or opportunity) that varies from one category to another. For example, suppose you are creating a Pareto chart for the number of medical claims that are categorized by the job titles of company employees who submit them. The counts can be weighted to adjust for the fact that there are more individuals in some jobs than in others and for the fact that some jobs might be associated with greater health risks than others.
- Use the NOCURVE option to eliminate the cumulative percentage curve in situations where the curve reveals little information about the data. In general, the bars should be more prominent than the curve.
- Maximize the space used for the bars by eliminating unnecessary labels and visual clutter. This is particularly important for comparative Pareto charts. The NOCATLABEL, NOFREQLABEL, and NOCUMLABEL options are useful for this purpose. You can also use the NOFREQTICK and NOCUMTICK options to eliminate tick marks and tick mark labels on the frequency and cumulative percentage axes.
- Make legends more informative by specifying legend labels.
- Avoid filling bars with multiple types of cross-hatched patterns; solid color fills are less distracting. Use color sparingly to emphasize important features (such as the "vital few" categories), and choose bar colors that provide good visual discrimination.
- If you are working with a large data set that involves many categories, limit the number of categories that are displayed to achieve visual clarity.
- If your application involves classification effects, construct more than one Pareto chart for the data by using various combinations of classification variables. (This approach is illustrated in Example 16.2).
- Provide reference lines on comparative Pareto charts to aid visual comparison.

See to Chapter 2 of Cleveland (1985) for a general discussion of the principles of statistical graphics.

Missing Values

By default, observations that have missing values of a process variable are not processed. If you specify the MISSING option, then missing values are treated as a Pareto category.

Likewise, observations that have missing values of the CLASS= variables are not processed by default. Missing values of the first CLASS= variable are treated as a level if the MISSING1 option is specified, and missing values of the second CLASS= variable are treated as a level if the MISSING2 option is specified.

Role of Variable Formats

The categories of a Pareto chart are always determined using formatted values of the process variable, and the format is used to label the categories.

On the chart, the categories are displayed in decreasing order of frequency. If multiple categories have the same count, the tied categories are displayed in order of their formatted values.

When you create a comparative Pareto chart, the formatted levels of the CLASS= variables are used to group the data into cells. There is a cell for each level of the CLASS= variable in a one-way comparative chart, and there is a cell for each combination of levels of the CLASS= variables in a two-way comparative chart.

You can specify the order of the rows and columns that correspond to the classification levels by specifying the ORDER1= and ORDER2= options. The default value of these options is INTERNAL, which means that the order is determined by the internal values of the CLASS= variables. It is possible for a particular formatted value to correspond to more than one internal value. To resolve this ambiguity, the internal value that determines the position of the row or column is the value that occurs first in the input data set.

Other values that you can specify for the ORDER1= and ORDER2= options are FORMATTED, FREQ, and DATA.

Large Data Sets

Although there is no limit to the number of observations that can be read from an input data set, the maximum number of Pareto categories that can be read is 32,767. This limit is a practical issue only if you are creating a restricted Pareto chart from a large data set, because the number of categories that can be displayed is limited by the resolution of your graphical display. The number of categories that can be read is limited by the amount of memory available, because the levels are stored in memory. If you run out of memory, you should first reduce the data by using the FREQ procedure.

Examples: PARETO Procedure

Example 16.1: Creating Before-and-After Pareto Charts

NOTE: See *Before & After Pareto Charts Using a BY Variable* in the SAS/QC Sample Library.

During the manufacture of a metal-oxide semiconductor (MOS) capacitor, causes of failures were recorded before and after a tube in the diffusion furnace was cleaned. This information was saved in a SAS data set named Failure3:

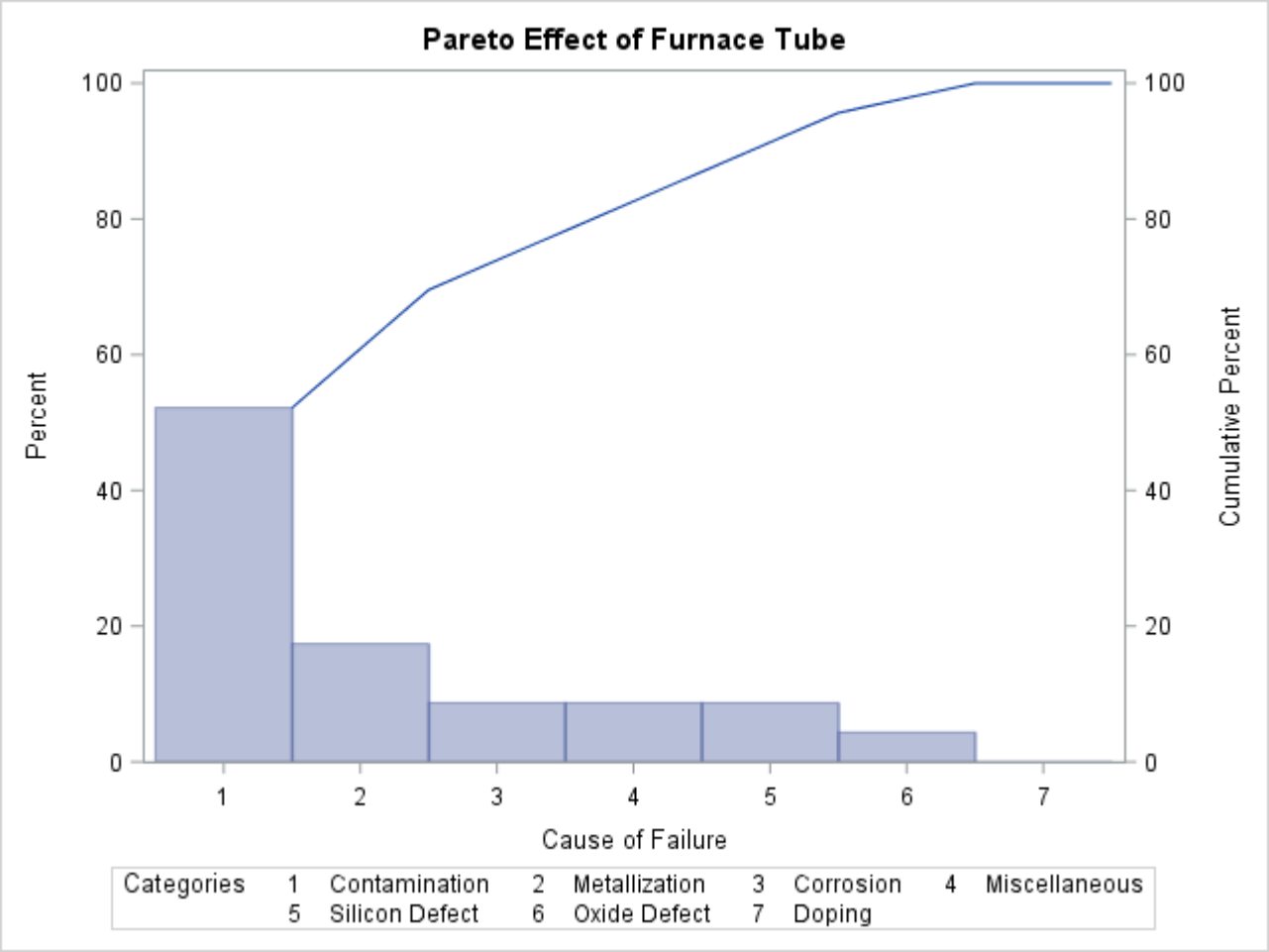
```
data Failure3;
   length Cause $ 16 Stage $ 16;
   label  Cause = 'Cause of Failure';
   input  Stage & $ Cause & $ Counts;
datalines;
Before Cleaning   Contamination    14
Before Cleaning   Corrosion         2
Before Cleaning   Doping            1
Before Cleaning   Metallization     2
Before Cleaning   Miscellaneous     3
Before Cleaning   Oxide Defect      8
Before Cleaning   Silicon Defect    1
After Cleaning    Doping            0
After Cleaning    Corrosion         2
After Cleaning    Metallization     4
After Cleaning    Miscellaneous     2
After Cleaning    Oxide Defect      1
After Cleaning    Contamination    12
After Cleaning    Silicon Defect    2
;
```

To compare distribution of failures before and after cleaning, you can use the BY statement to create two separate Pareto charts, one for the observations in which Stage is equal to `'Before Cleaning'` and one for the observations in which Stage is equal to `'After Cleaning'`:

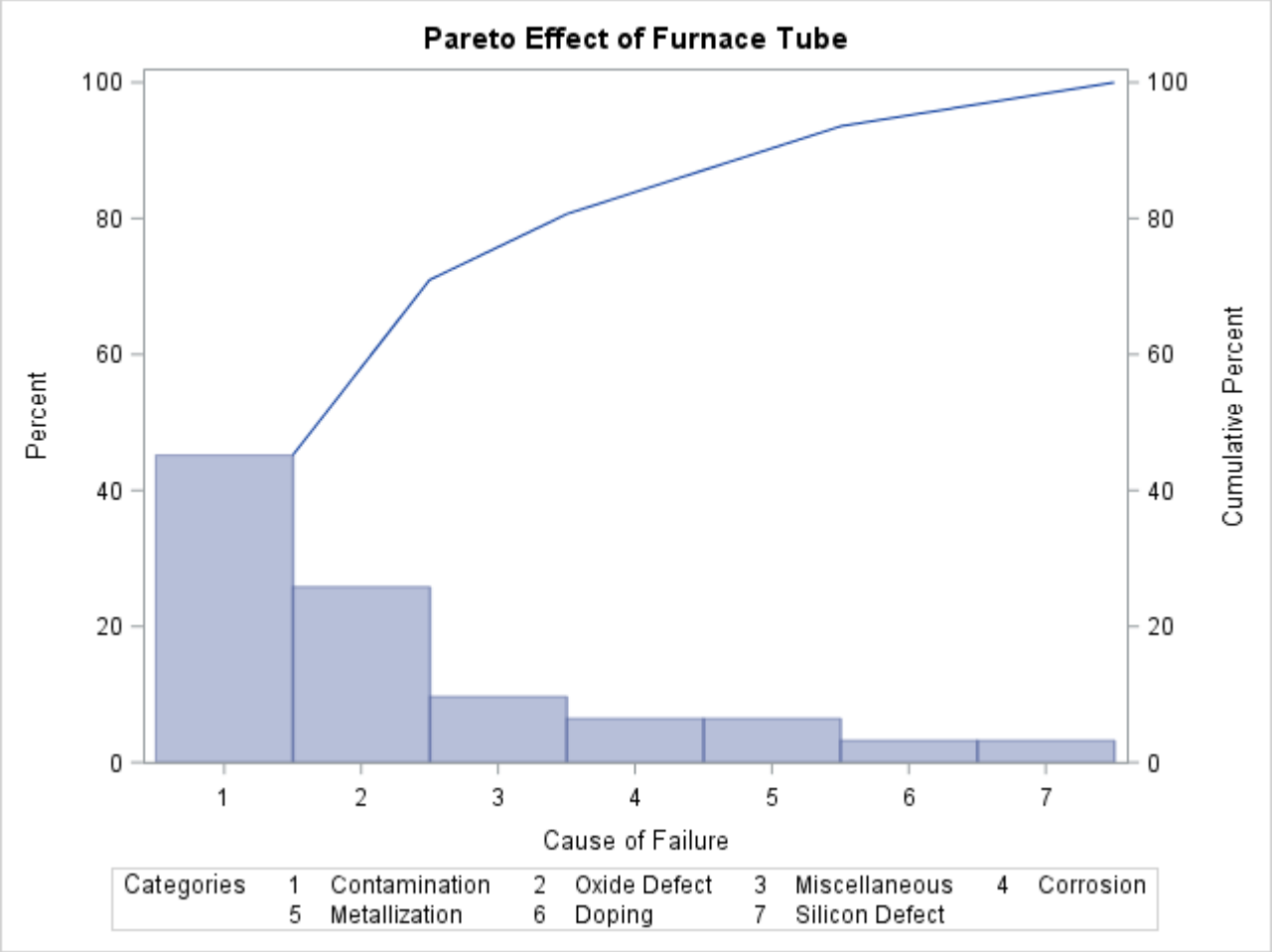
```
proc sort data=Failure3;
   by Stage;
run;

title 'Pareto Effect of Furnace Tube';
proc pareto data=Failure3;
   vbar Cause / freq     = Counts
                odstitle = title;
   by Stage;
run;
```

The SORT procedure sorts the observations in order of the values of Stage. It is not necessary to sort by the values of Cause because this is done by the PARETO procedure. The two charts, displayed in Output 16.1.1 and Output 16.1.2, reveal a reduction in oxide defects after the tube was cleaned. This is a relative reduction, because the frequency axes are scaled in percentage units. Note that the `'After Cleaning'` chart is produced first, based on alphabetical sorting of BY groups.

Output 16.1.1 "After" Analysis Using Stage as a BY Variable



Output 16.1.2 "Before" Analysis Using Stage as a BY Variable

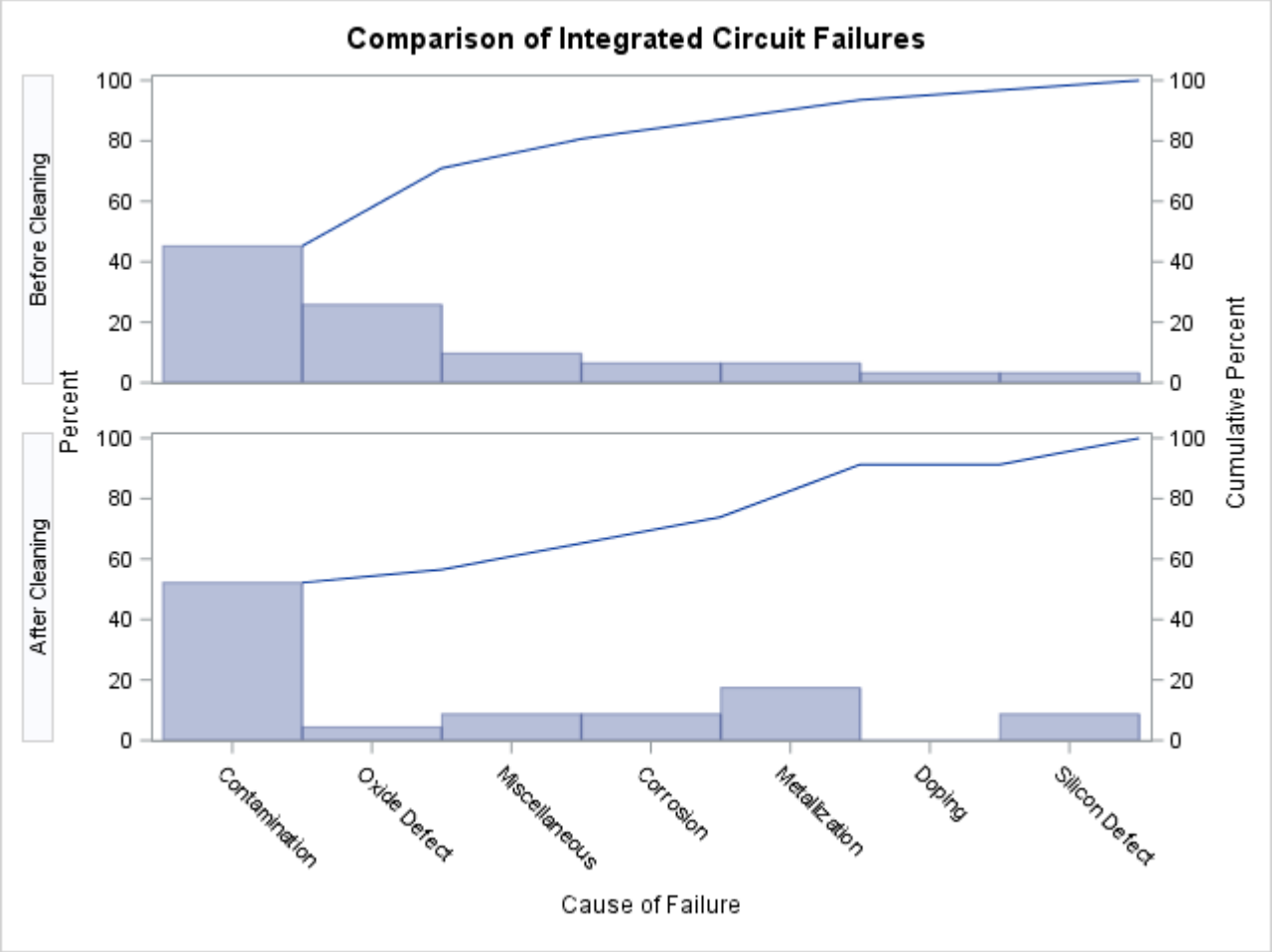


In general, it is difficult to compare Pareto charts that are created by using BY processing because their axes are not necessarily uniform. A better approach is to construct a comparative Pareto chart, as illustrated by the following statements:

```
title 'Comparison of Integrated Circuit Failures';
proc pareto data=Failure3;
   vbar Cause / class      = Stage
                freq       = Counts
                scale      = percent
                intertile  = 5.0
                classkey   = 'Before Cleaning'
                odstitle   = title;
run;
```

The CLASS= option designates Stage as a classification variable, and this directs PROC PARETO to create the one-way comparative Pareto chart shown in Output 16.1.3, which displays a component chart for each level of Stage. The INTERTILE= option separates the cells with an offset of 5 screen percentage units.

Output 16.1.3 Before-and-After Analysis That Uses a Comparative Pareto Chart



In a comparative Pareto chart, there is always one special cell, called the *key cell*, in which the bars are displayed in decreasing order, and whose order determines the uniform category axis that is used for all the cells. The key cell is positioned at the top of the chart. Here, the key cell is the set of observations for which Stage equals `'Before Cleaning'`, as specified by the CLASSKEY= option. By default, the levels are sorted in the order determined by the ORDER1= option, and the key cell is the level that occurs first in this order.

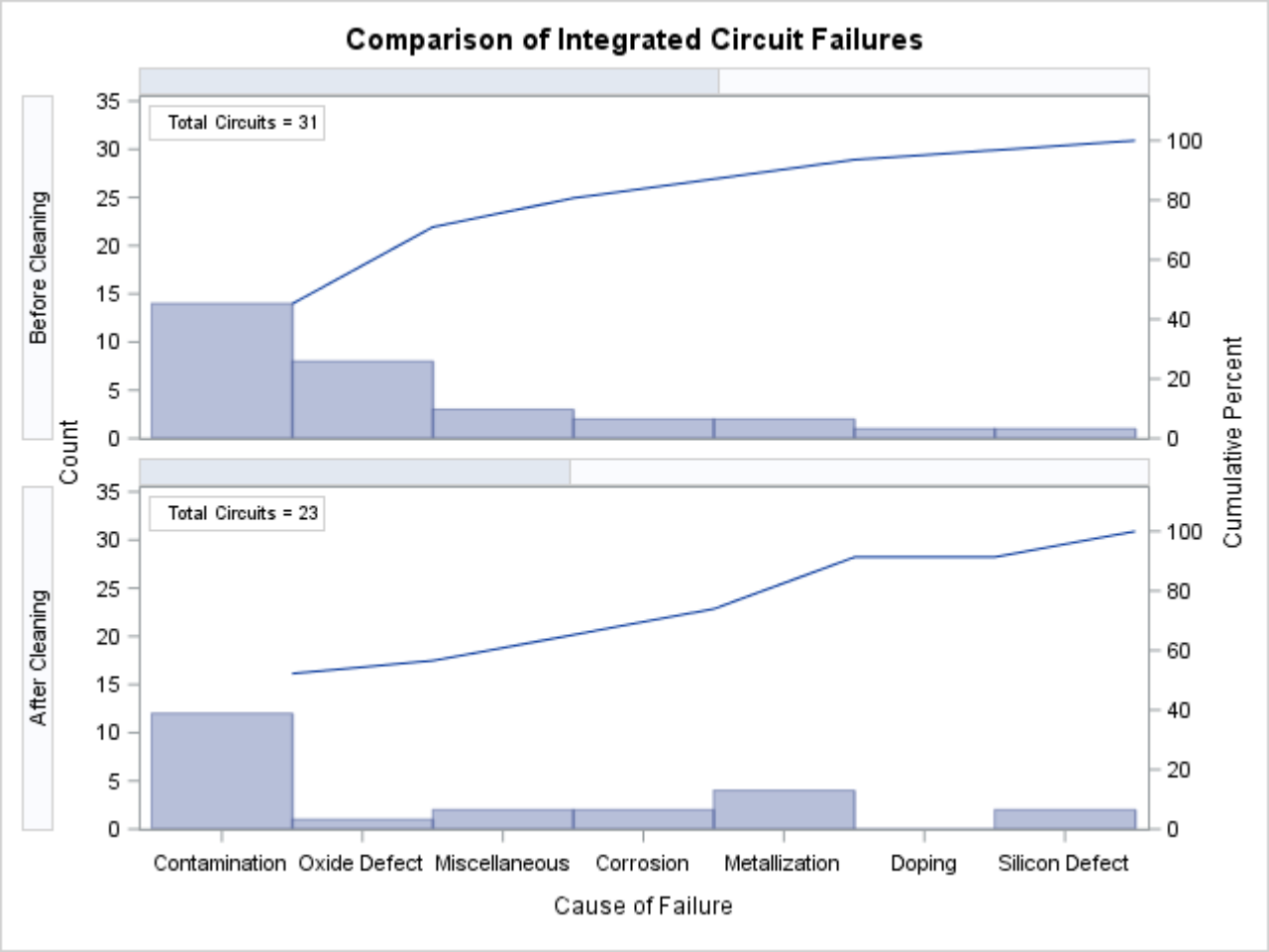
In many applications, it can be more revealing to base comparisons on counts rather than percentages. The following statements construct a chart that has a frequency scale:

```
title 'Comparison of Integrated Circuit Failures';
proc pareto data=Failure3;
   vbar Cause / class     = Stage
                freq      = Counts
                scale     = count
                nlegend   = 'Total Circuits'
                classkey  = 'Before Cleaning'
                odstitle  = title
                cframenleg
                cprop;
run;
```

Specifying SCALE=COUNT scales the frequency axis in count units. The NLEGEND= option adds a sample size legend, and the CFRAMENLEG option frames the legend. The CPROP option adds bars that indicate the proportion of total frequency represented by each cell.

The chart is shown in Output 16.1.4.

Output 16.1.4 Before-and-After Analysis Using Comparative Pareto Chart



Note that the lower cumulative percentage curve in Output 16.1.4 is not anchored to the first bar. This is a consequence of the uniform frequency scale and of the fact that the number of observations in each cell is not the same.

Example 16.2: Creating Two-Way Comparative Pareto Charts

NOTE: See *Basic and Comparative Pareto Charts* in the SAS/QC Sample Library.

During the manufacture of a MOS capacitor, different cleaning processes were used by two manufacturing systems operating in parallel. Process A used a standard cleaning solution, and Process B used a different cleaning mixture that contained less particulate matter. The failure causes that were observed with each process for five consecutive days were recorded and saved in a SAS data set called Failure4:

```
data Failure4;
   length Process $ 9 Cause $ 16;
   label  Cause   = 'Cause of Failure';
   input  Process & $ Day & $ Cause & $ Counts;
   datalines;
Process A   March 1   Contamination    15
Process A   March 1   Corrosion         2
Process A   March 1   Doping           1
Process A   March 1   Metallization    2
Process A   March 1   Miscellaneous    3
Process A   March 1   Oxide Defect     8
Process A   March 1   Silicon Defect   1
Process A   March 2   Contamination    16
Process A   March 2   Corrosion        3
Process A   March 2   Doping           1
Process A   March 2   Metallization    3
Process A   March 2   Miscellaneous    1
Process A   March 2   Oxide Defect     9
Process A   March 2   Silicon Defect   2
Process A   March 3   Contamination    20
Process A   March 3   Corrosion        1
Process A   March 3   Doping           1
Process A   March 3   Metallization    0
Process A   March 3   Miscellaneous    3
Process A   March 3   Oxide Defect     7
Process A   March 3   Silicon Defect   2
Process A   March 4   Contamination    12
Process A   March 4   Corrosion        1
Process A   March 4   Doping           1
Process A   March 4   Metallization    0
Process A   March 4   Miscellaneous    0
Process A   March 4   Oxide Defect     10
Process A   March 4   Silicon Defect   1
Process A   March 5   Contamination    23
Process A   March 5   Corrosion        1
Process A   March 5   Doping           1
Process A   March 5   Metallization    0
Process A   March 5   Miscellaneous    1
Process A   March 5   Oxide Defect     8
Process A   March 5   Silicon Defect   2
Process B   March 1   Contamination    8
Process B   March 1   Corrosion        2
Process B   March 1   Doping           1
Process B   March 1   Metallization    4
Process B   March 1   Miscellaneous    2
Process B   March 1   Oxide Defect     10
Process B   March 1   Silicon Defect   3
Process B   March 2   Contamination    9
Process B   March 2   Corrosion        0
Process B   March 2   Doping           1
Process B   March 2   Metallization    2
Process B   March 2   Miscellaneous    4
Process B   March 2   Oxide Defect     9
```

```
Process B   March 2   Silicon Defect    2
Process B   March 3   Contamination     4
Process B   March 3   Corrosion         1
Process B   March 3   Doping            1
Process B   March 3   Metallization     0
Process B   March 3   Miscellaneous     0
Process B   March 3   Oxide Defect     10
Process B   March 3   Silicon Defect    1
Process B   March 4   Contamination     2
Process B   March 4   Corrosion         2
Process B   March 4   Doping            1
Process B   March 4   Metallization     0
Process B   March 4   Miscellaneous     3
Process B   March 4   Oxide Defect      7
Process B   March 4   Silicon Defect    1
Process B   March 5   Contamination     1
Process B   March 5   Corrosion         3
Process B   March 5   Doping            1
Process B   March 5   Metallization     0
Process B   March 5   Miscellaneous     1
Process B   March 5   Oxide Defect      8
Process B   March 5   Silicon Defect    2
;
```

In addition to the process variable Cause, this data set has two classification variables: Process and Day. The variable Counts is a frequency variable.

This example creates a series of displays that progressively use more of the classification information.

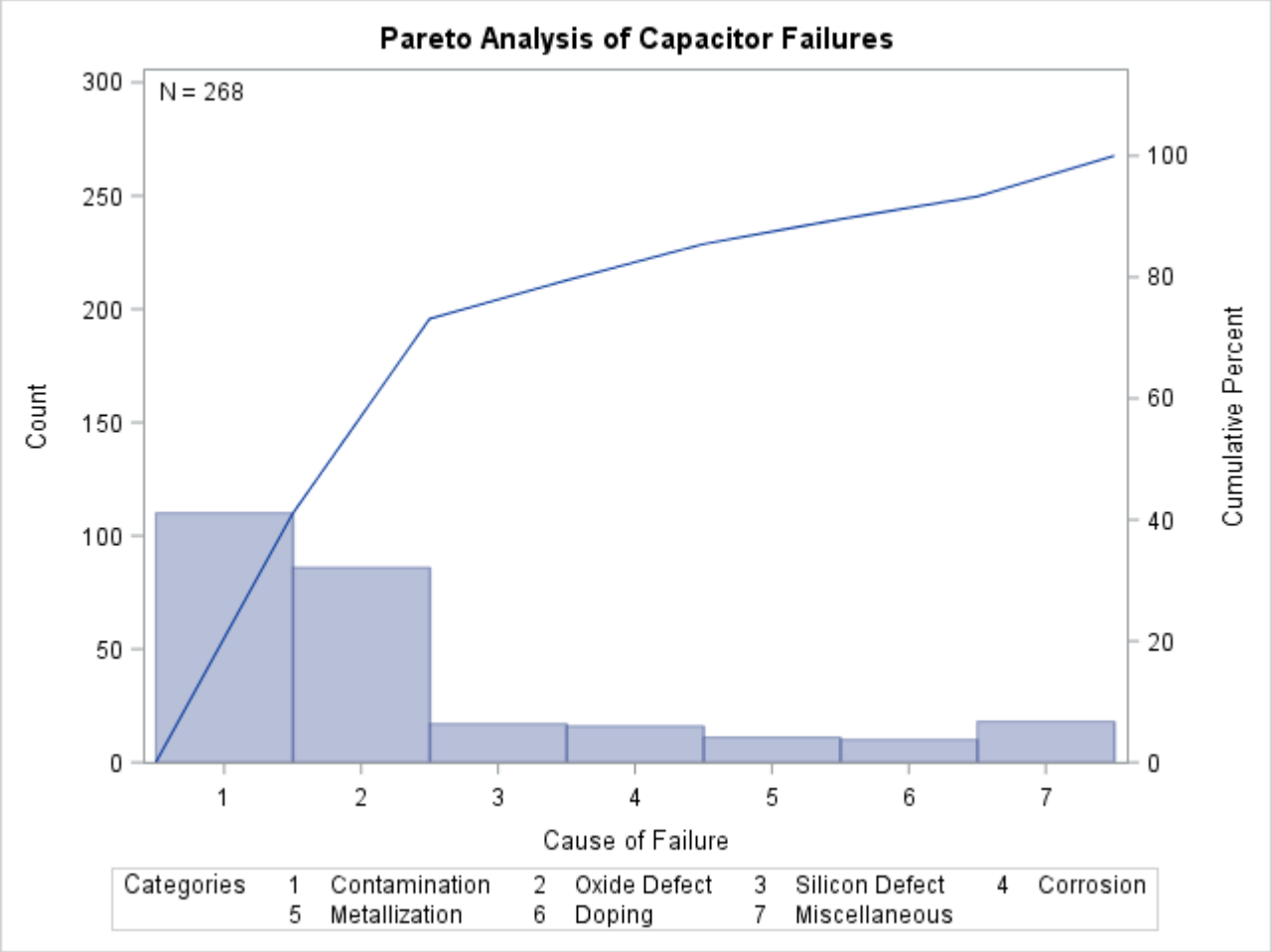
Basic Pareto Chart

The following statements create the first display, which analyzes the process variable without taking into account the classification variables:

```
title 'Pareto Analysis of Capacitor Failures';
proc pareto data=Failure4;
   vbar Cause / freq     = Counts
                last     = 'Miscellaneous'
                scale    = count
                anchor   = bl
                odstitle = title
                nlegend;
run;
```

The chart, shown in Output 16.2.1, indicates that contamination is the most frequently occurring problem.

Output 16.2.1 Pareto Analysis without Classification Variables



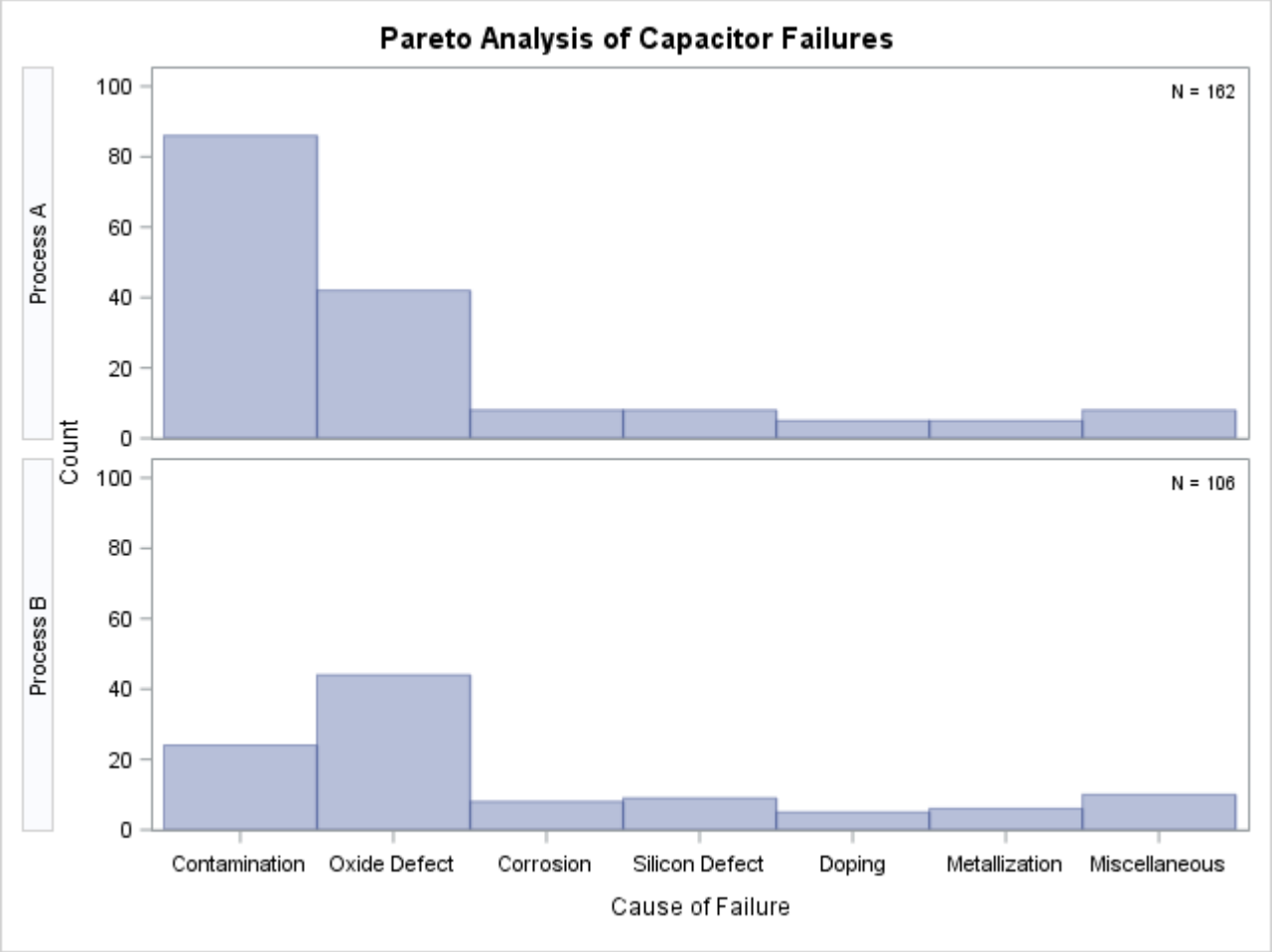
The ANCHOR=BL option anchors the cumulative percentage curve at the bottom left (BL) of the first bar. The NLEGEND option adds a sample size legend.

One-Way Comparative Pareto Chart for Process

The following statements specify Process as a classification variable to create a comparative Pareto chart, which is displayed in Output 16.2.2:

```
proc pareto data=Failure4;
   vbar Cause / class       = Process
                freq        = Counts
                last        = 'Miscellaneous'
                scale       = count
                odstitle    = title
                nocurve
                nlegend;
run;
```

Output 16.2.2 One-Way Comparative Pareto Analysis with CLASS=Process



Each cell corresponds to a level of the CLASS= variable (Process). By default, the cells are arranged from top to bottom in alphabetical order of the formatted values of Process, and the key cell is the top cell. The main difference in the two cells is a decrease in contamination when Process B is used.

The NOCURVE option suppresses the cumulative percentage curve, along with the cumulative percentage axis.

One-Way Comparative Pareto Chart for Day

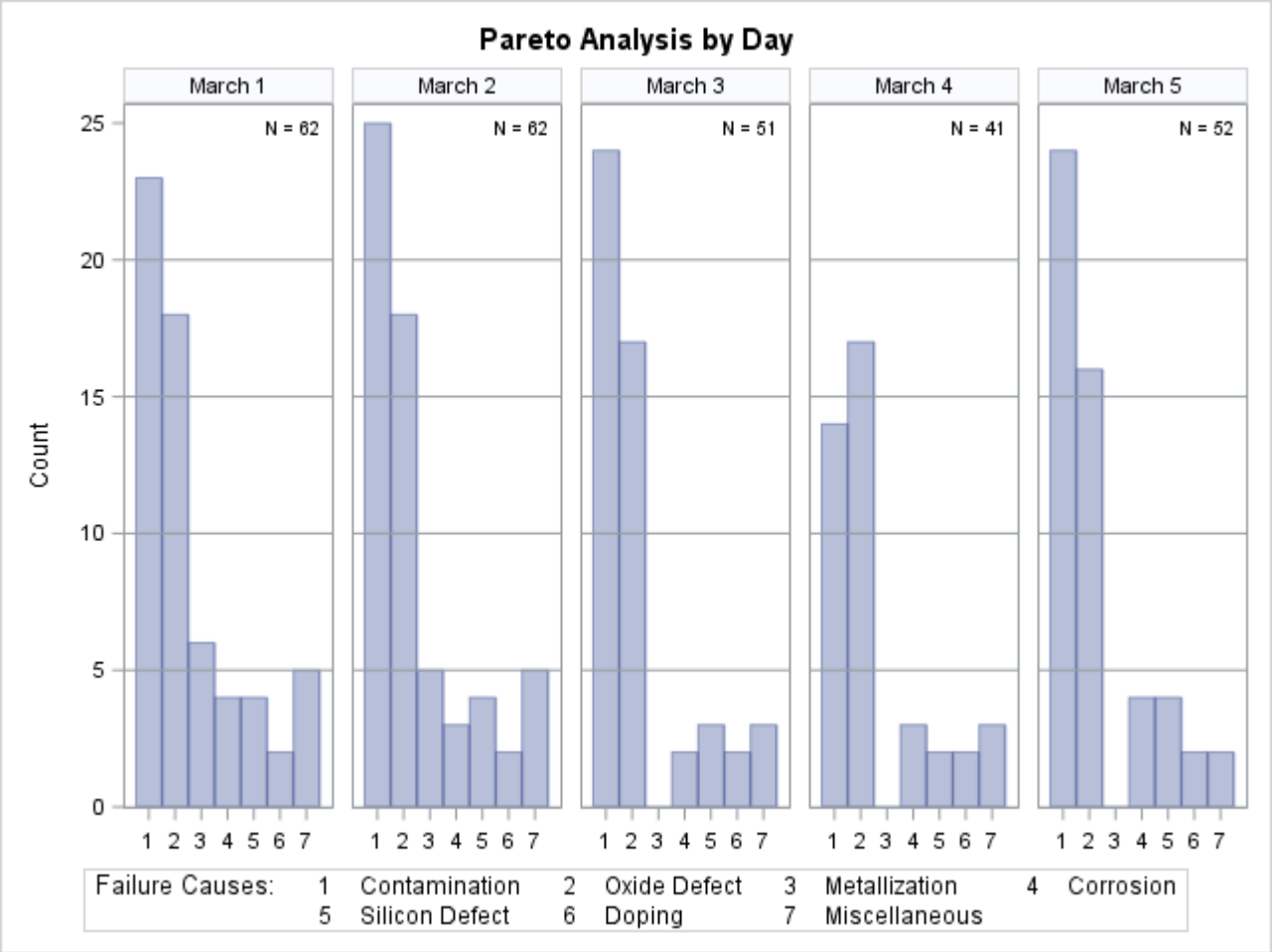
The following statements specify Day as a classification variable:

```
title 'Pareto Analysis by Day';
proc pareto data=Failure4;
   vbar Cause / class       = Day
                freq        = Counts
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                odstitle    = title
                nrows       = 1
```

```
                    ncols       = 5
                    freqref     = 5 10 15 20
                    nocatlabel
                    nocurve
                    nlegend;
run;
```

The NROWS= and NCOLS= options display the cells in a side-by-side arrangement. The FREQREF= option adds reference lines perpendicular to the frequency axis. The NOCATLABEL option suppresses the category axis labels, and the CATLEGLABEL= option incorporates that information into the category legend label. The chart is displayed in Output 16.2.3.

Output 16.2.3 One-Way Comparative Pareto Analysis with CLASS=Day



By default, the key cell is the leftmost cell. There were no failures due to metallization starting on March 3 (in fact, process controls to reduce this problem were introduced on this day).

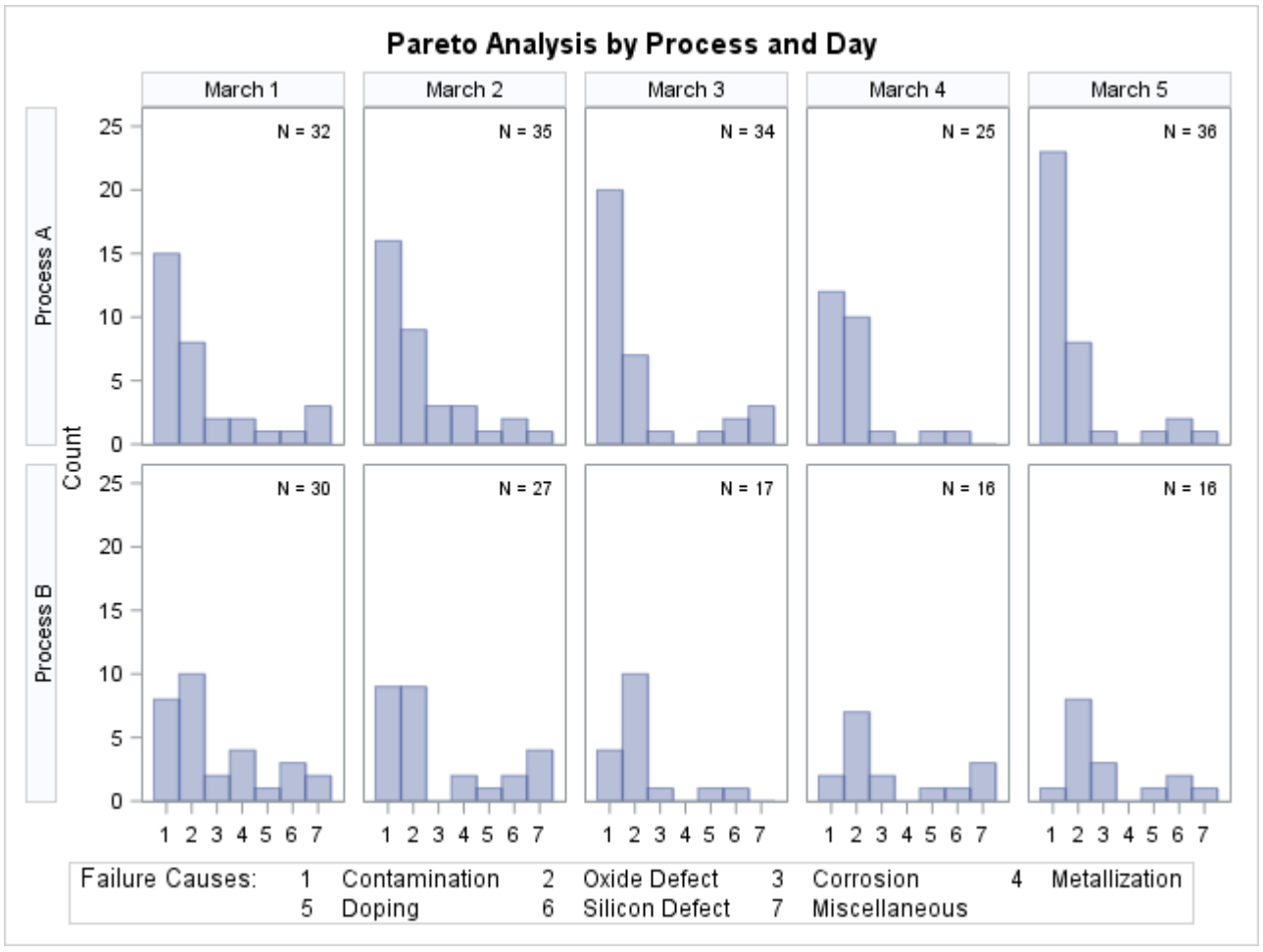
Two-Way Comparative Pareto Chart for Process and Day

The following statements specify both Process and Day as CLASS= variables to create a two-way comparative Pareto chart:

```
title 'Pareto Analysis by Process and Day';
proc pareto data=Failure4;
   vbar Cause / class       = ( Process Day )
                freq        = Counts
                nrows       = 2
                ncols       = 5
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                odstitle    = title
                nocatlabel
                nocurve
                nlegend;
run;
```

The chart is displayed in Output 16.2.4.

Output 16.2.4 Two-Way Comparative Pareto Analysis for Process and Day



The cells are arranged in a matrix whose rows correspond to levels of the first CLASS= variable (Process) and whose columns correspond to levels of the second CLASS= variable (Day). The dimensions of the matrix are specified in the NROWS= and NCOLS= options. The key cell is in the upper left corner.

The chart reveals continuous improvement when Process B is used.

Example 16.3: Highlighting the "Vital Few"

NOTE: See *Highlighting the "Vital Few"* in the SAS/QC Sample Library.

This example is a continuation of Example 16.2.

In some applications you might want to use colors and patterns to highlight the bars that correspond to the most frequently occurring categories, which are referred to as the "vital few."

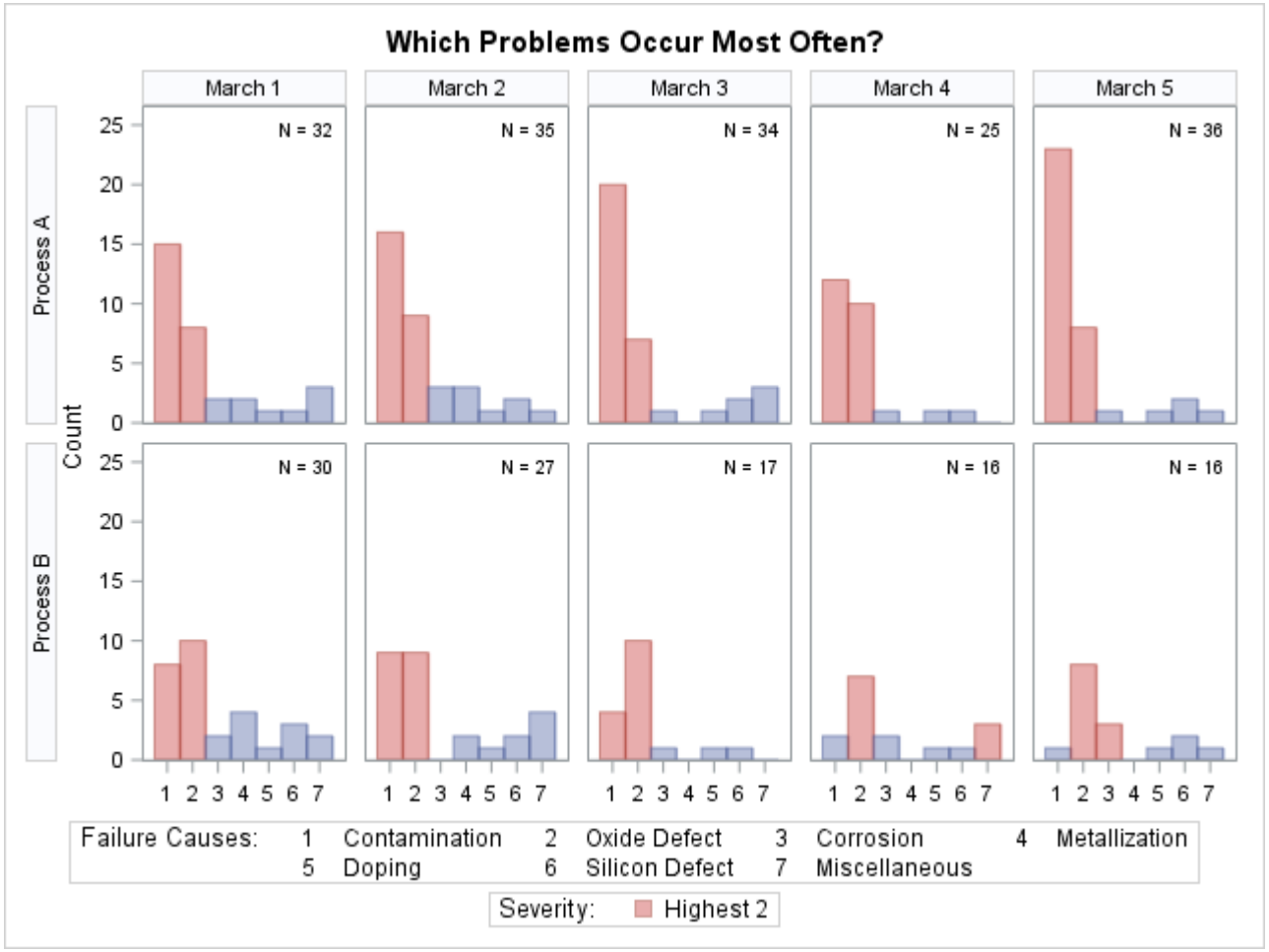
The following statements highlight the two most frequently occurring categories in each cell of the comparative Pareto chart shown in Output 16.2.4:

```
title 'Which Problems Occur Most Often?';
proc pareto data=Failure4;
   vbar Cause / class       = ( Process Day )
                freq        = Counts
                nrows       = 2
                ncols       = 5
                last        = 'Miscellaneous'
                scale       = count
                chigh(2)
                hlleglabel  = 'Severity:'
                catleglabel = 'Failure Causes:'
                odstitle    = title
                nocatlabel
                nocurve
                nlegend;
run;
```

Specifying CHIGH(2) causes the two highest bars in each cell to be filled with a contrasting color from the ODS style. The new chart is displayed in Output 16.3.1. In all but two of the cells, the two vital problems are `'Contamination'` and `'Oxide Defect'`.

You can also highlight the "trivial many" categories (also referred to as the "useful many") with the CLOW(*m*) option. You can use these options in conjunction with the CHIGH(*n*) and BARS= options. For more information, see the entries for these options in the "Dictionary of HBAR and VBAR Statement Options" on page 1095.

Output 16.3.1 Emphasizing the "Vital Few"



Example 16.4: Highlighting Combinations of Categories

NOTE: See *Highlighting Specific Pareto Categories* in the SAS/QC Sample Library.

In some applications, it is useful to classify the categories into groups that are not necessarily related to frequency. This example, which is a continuation of Example 16.2, shows how you use a bar legend to display this classification.

Suppose that contamination and metallization are high-priority problems, oxide defect is a medium-priority problem, and all other categories are low-priority problems. Begin by adding this information to the data set Failure4 as follows:

```
data Failure4;
   length Priority $ 16;
   set Failure4;
   if Cause = 'Contamination' or
      Cause = 'Metallization'
   then
      Priority = 'High';
   else
   if Cause = 'Oxide Defect'
   then
      Priority = 'Medium';
   else
      Priority = 'Low';
run;
```

The variable Priority indicates the priority that is associated with a defect cause.

The following statements specify Priority in both the BARS= and BARLEGEND= options:

```
title 'Which Problems Take Priority?';
proc pareto data=Failure4;
   vbar Cause / class       = ( Process Day )
                freq        = Counts
                nrows       = 2
                ncols       = 5
                last        = 'Miscellaneous'
                scale       = count
                bars        = ( Priority )
                barlegend   = ( Priority )
                barleglabel = 'Priority:'
                catleglabel = 'Failure Causes:'
                odstitle    = title
                nocatlabel
                nocurve
                nlegend;
run;
```

Colors from the ODS style are assigned to the bars based on levels of the BARS= variable. The chart is displayed in Output 16.4.1. The levels of the BARLEGEND= variable are the values that are displayed in the legend labeled “Priority:” at the bottom of the chart.

In general, when you specify BARS= and BARLEGEND= variables, their values must be consistent and unambiguous. Each observation that has a particular value of the process variable should have the same BARS= or BARLEGEND= variable value. For more information, see the entries for the BARS= and BARLEGEND= options in “Dictionary of HBAR and VBAR Statement Options” on page 1095.

Output 16.4.1 Highlighting Selected Subsets of Categories



Example 16.5: Highlighting Combinations of Cells

NOTE: See *Highlighting Tiles in a Comparative Pareto Chart* in the SAS/QC Sample Library.

This example is a continuation of Example 16.4.

In some applications that involve comparative Pareto charts, it is useful to classify the cells into groups. This example shows how you can use traditional graphics to display this type of classification by coloring the cells (also called tiles) and adding a legend.

Suppose you want to enhance Output 16.4.1 by highlighting the two cells for which Process='Process B' and Day='March 4' and 'March 5' to emphasize the improvement displayed in those cells. Begin by adding a tile color variable (Tilecol) and a tile legend variable (Tileleg) to the data set Failure4 as follows:

```
data Failure4;
   length Tilecol $ 8 Tileleg $ 16;
   set Failure4;
   if (Process='Process B') and (Day='March 4' or Day='March 5')
   then do; Tilecol='ywh'; Tileleg = 'Improvement'; end;
   else do; Tilecol='ligr'; Tileleg = 'Status Quo'; end;
run;
```

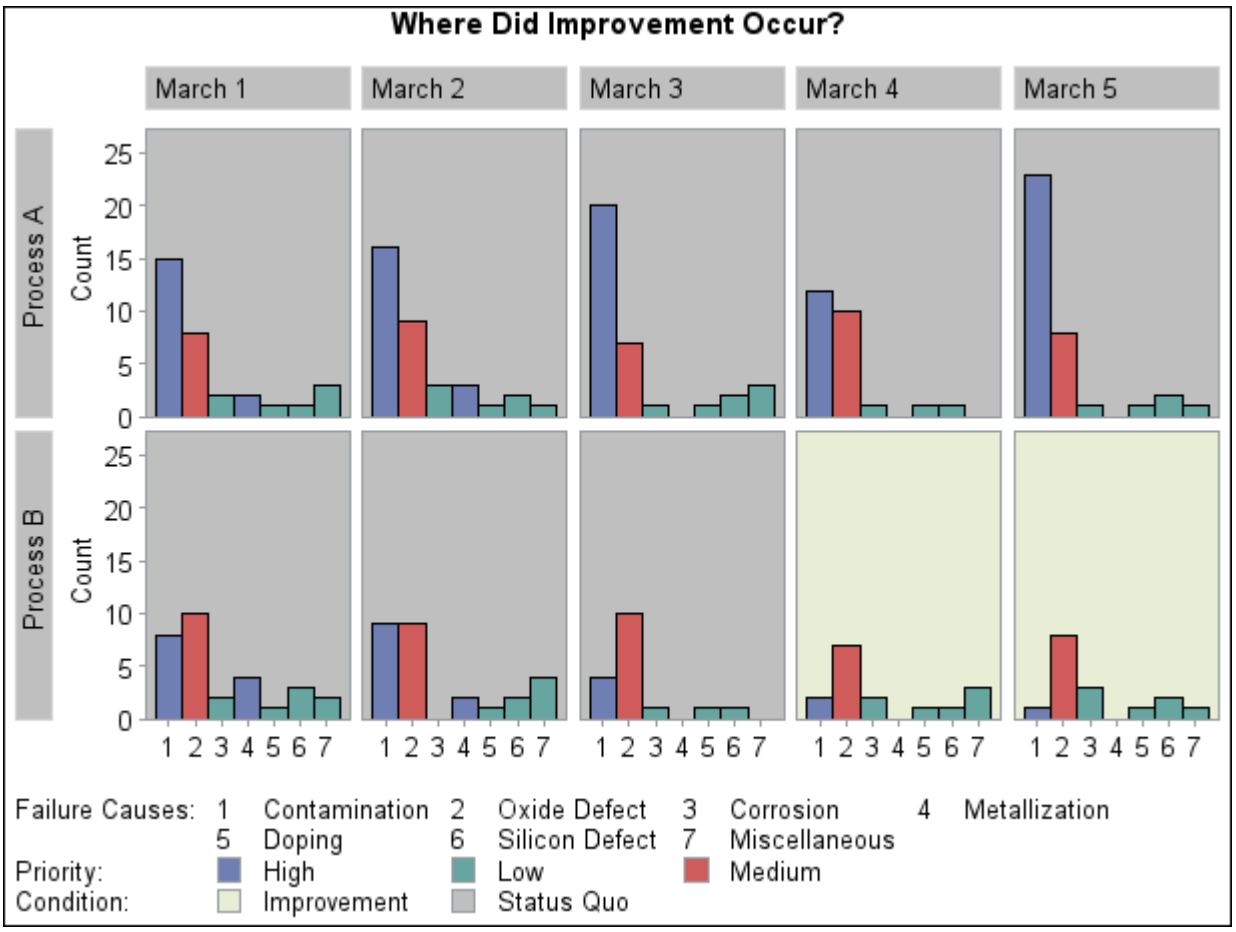
The following statements specify Tilecol as a CTILES= variable and Tileleg as a TILELEGEND= variable. Note that the variable names are enclosed in parentheses.

```
ods graphics off;
title 'Where Did Improvement Occur?';
proc pareto data=Failure4;
   vbar Cause / class        = ( Process Day )
                freq         = Counts
                nrows        = 2
                ncols        = 5
                last         = 'Miscellaneous'
                scale        = count
                catleglabel  = 'Failure Causes:'
     /* options for highlighting bars: */
                bars         = ( Priority )
                barlegend    = ( Priority )
                barleglabel  = 'Priority:'
     /* options for highlighting tiles: */
                ctiles       = ( Tilecol )
                tilelegend   = ( Tileleg )
                tileleglabel = 'Condition:'
                intertile    = 1.0
                cframeside   = ligr
                cframetop    = ligr
                nocatlabel
                nocurve;
run;
```

The ODS GRAPHICS OFF statement before the PROC statement disables ODS Graphics, so the Pareto chart is produced using traditional graphics. The CTILES=, TILELEGEND=, CFRAMESIDE=, and CFRAMETOP= options are valid only for traditional graphics output. See the section "Options for Traditional Graphics" on page 1110 for descriptions of options specific to traditional graphics.

In the chart, shown in Output 16.5.1, the values that are displayed in the legend labeled "Condition:" are the levels of the TILELEGEND= variable.

Output 16.5.1 Highlighting Specific Tiles

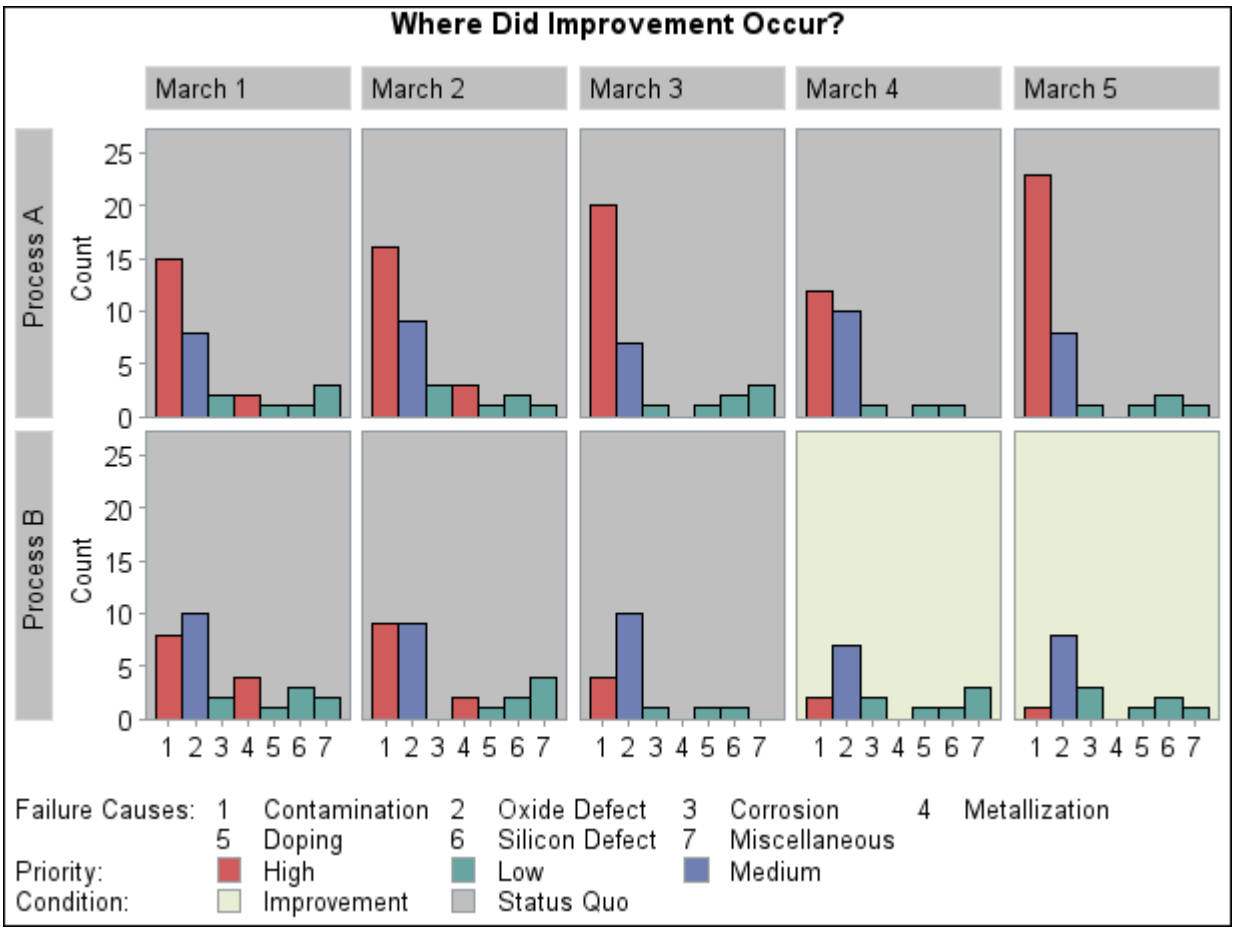


PROC PARETO sequentially assigns colors from a list defined by the ODS style to the levels of the BARS= variable. The first color is associated with the first value of Priority, and so on. When traditional graphics is enabled, you can use the CBARS= option to assign specific colors to Pareto categories. The following statements assign explicit color values to the variable PriorityColor:

```
data Failure4;
   length PriorityColor $ 8;
   set Failure4;
   if Priority = 'High'
   then
      PriorityColor = 'CXD05B5B';
   else
   if Priority = 'Medium'
   then
      PriorityColor = 'CX6F7EB3';
   else
      PriorityColor = 'CX66A5A0';
run;
```

Output 16.5.2 shows the chart that is produced by replacing the BARS= option with CBARS=PriorityColor. The high-priority problems are represented by red bars, and the low-priority problems are represented by green bars.

Output 16.5.2 Assigning Specific Bar Colors



Example 16.6: Ordering Rows and Columns in a Comparative Pareto Chart

NOTE: See *Ordering Rows and Columns in a Comparative Chart* in the SAS/QC Sample Library.

This example illustrates methods for controlling the order of rows and columns in a comparative Pareto chart.

The following statements create a data set named Failure5:

```
proc format;
   value procfmt   1   = 'Process A'
                   2   = 'Process B';
   value dayfmt    1   = 'Monday'
                   2   = 'Tuesday'
                   3   = 'Wednesday'
                   4   = 'Thursday'
                   5   = 'Friday';
run;

data Failure5;
   length Cause $16;
   format Process procfmt. Day dayfmt.;
```

```
   label  Cause   = 'Cause of Failure'
          Process = 'Cleaning Method'
          Day     = 'Day of Manufacture';
   input  Process Day Cause $16. Counts @@;
   datalines;
   1   1   Contamination    15   1   1   Corrosion         2
   1   1   Doping            1   1   1   Metallization     2
   1   1   Miscellaneous     3   1   1   Oxide Defect      8
   1   1   Silicon Defect    1   1   2   Contamination    16
   1   2   Corrosion         3   1   2   Doping            1
   1   2   Metallization     3   1   2   Miscellaneous     1
   1   2   Oxide Defect      9   1   2   Silicon Defect    2
   1   3   Contamination    20   1   3   Corrosion         1
   1   3   Doping            1   1   3   Metallization     0
   1   3   Miscellaneous     3   1   3   Oxide Defect      7
   1   3   Silicon Defect    2   1   4   Contamination    12
   1   4   Corrosion         1   1   4   Doping            1
   1   4   Metallization     0   1   4   Miscellaneous     0
   1   4   Oxide Defect     10   1   4   Silicon Defect    1
   1   5   Contamination    23   1   5   Corrosion         1
   1   5   Doping            1   1   5   Metallization     0
   1   5   Miscellaneous     1   1   5   Oxide Defect      8
   1   5   Silicon Defect    2   2   1   Contamination     8
   2   1   Corrosion         2   2   1   Doping            1
   2   1   Metallization     4   2   1   Miscellaneous     2
   2   1   Oxide Defect     10   2   1   Silicon Defect    3
   2   2   Contamination     9   2   2   Corrosion         0
   2   2   Doping            1   2   2   Metallization     2
   2   2   Miscellaneous     4   2   2   Oxide Defect      9
   2   2   Silicon Defect    2   2   3   Contamination     4
   2   3   Corrosion         1   2   3   Doping            1
   2   3   Metallization     0   2   3   Miscellaneous     0
   2   3   Oxide Defect     10   2   3   Silicon Defect    1
   2   4   Contamination     2   2   4   Corrosion         2
   2   4   Doping            1   2   4   Metallization     0
   2   4   Miscellaneous     3   2   4   Oxide Defect      7
   2   4   Silicon Defect    1   2   5   Contamination     1
   2   5   Corrosion         3   2   5   Doping            1
   2   5   Metallization     0   2   5   Miscellaneous     1
   2   5   Oxide Defect      8   2   5   Silicon Defect    2
;
```

Note that Failure5 is similar to the data set Failure4, which is created in Example 16.2. Here, the classification variables Process and Day are numeric formatted variables, and the formatted values of Day are `'Monday'` through `'Friday'`. In Example 16.2, Process and Day are character variables, and the values of Day are 'March 1' through 'March 5'.

The following statements create a two-way comparative Pareto chart for Cause; in this chart the rows represent levels of Process, and the columns represent levels of Day:

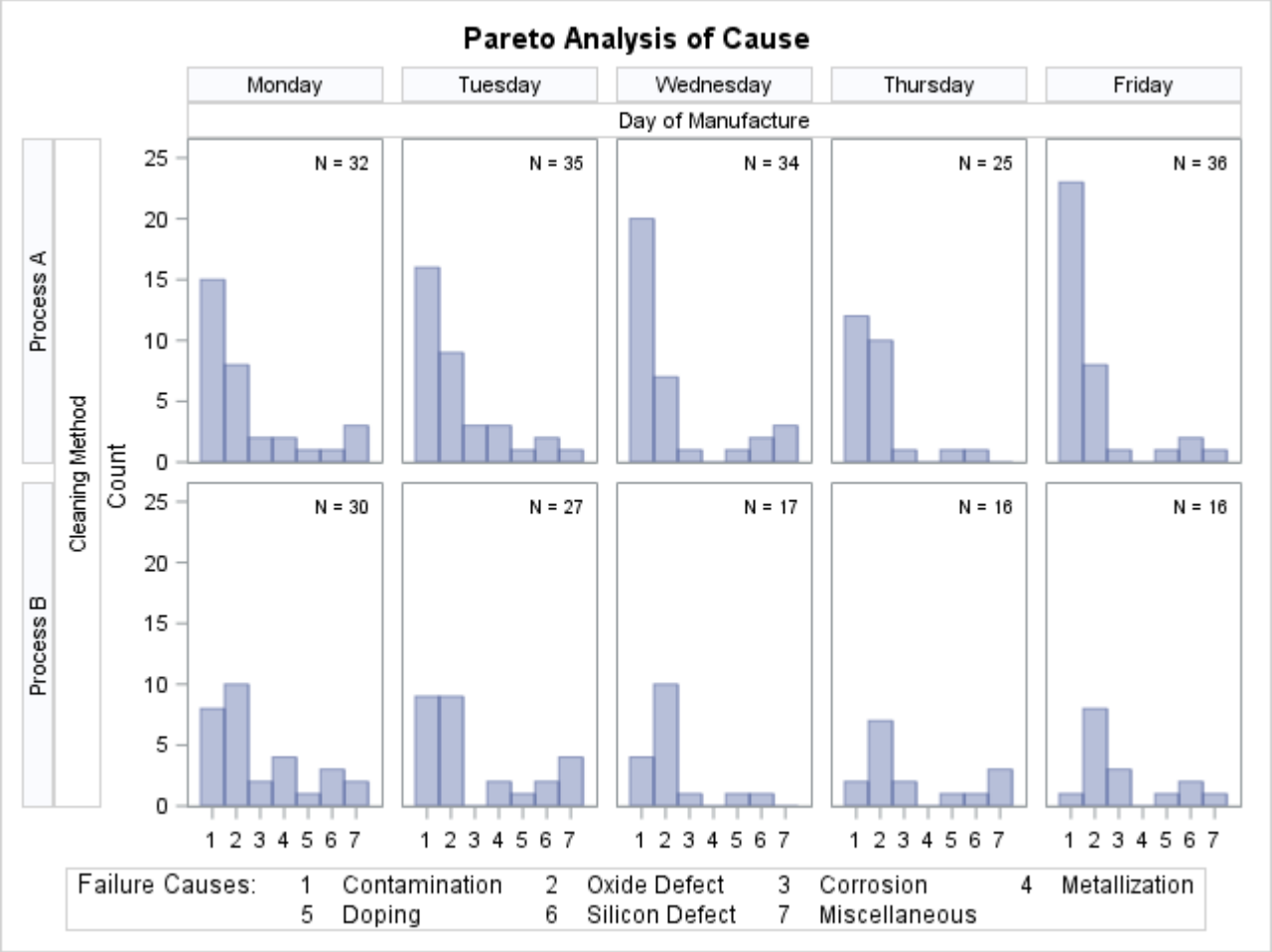
```
title 'Pareto Analysis by Process and Day';
proc pareto data=Failure5;
   vbar Cause / class       = ( Process Day )
                freq        = Counts
                nrows       = 2
                ncols       = 5
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                nocatlabel
                nocurve
                nlegend;
run;
```

The chart is shown in Output 16.6.1. The levels of the classification variables are determined by their formatted values. The default order in which the rows and columns are displayed is determined by the internal values of the classification variables, and consequently the columns appear in the order of the days of the week.

If Day had been defined as a character variable with values `'Monday'` through `'Friday'`, the columns in Output 16.6.1 would have appeared in alphabetical order.

You can override the default order by specifying the ORDER1= or ORDER2= option (or both).

Output 16.6.1 Controlling Row and Column Order



Example 16.7: Merging Columns in a Comparative Pareto Chart

NOTE: See *Merging Columns in a Comparative Pareto Chart* in the SAS/QC Sample Library.

This example is a continuation of Example 16.4 and illustrates a method for merging the columns in a comparative Pareto chart.

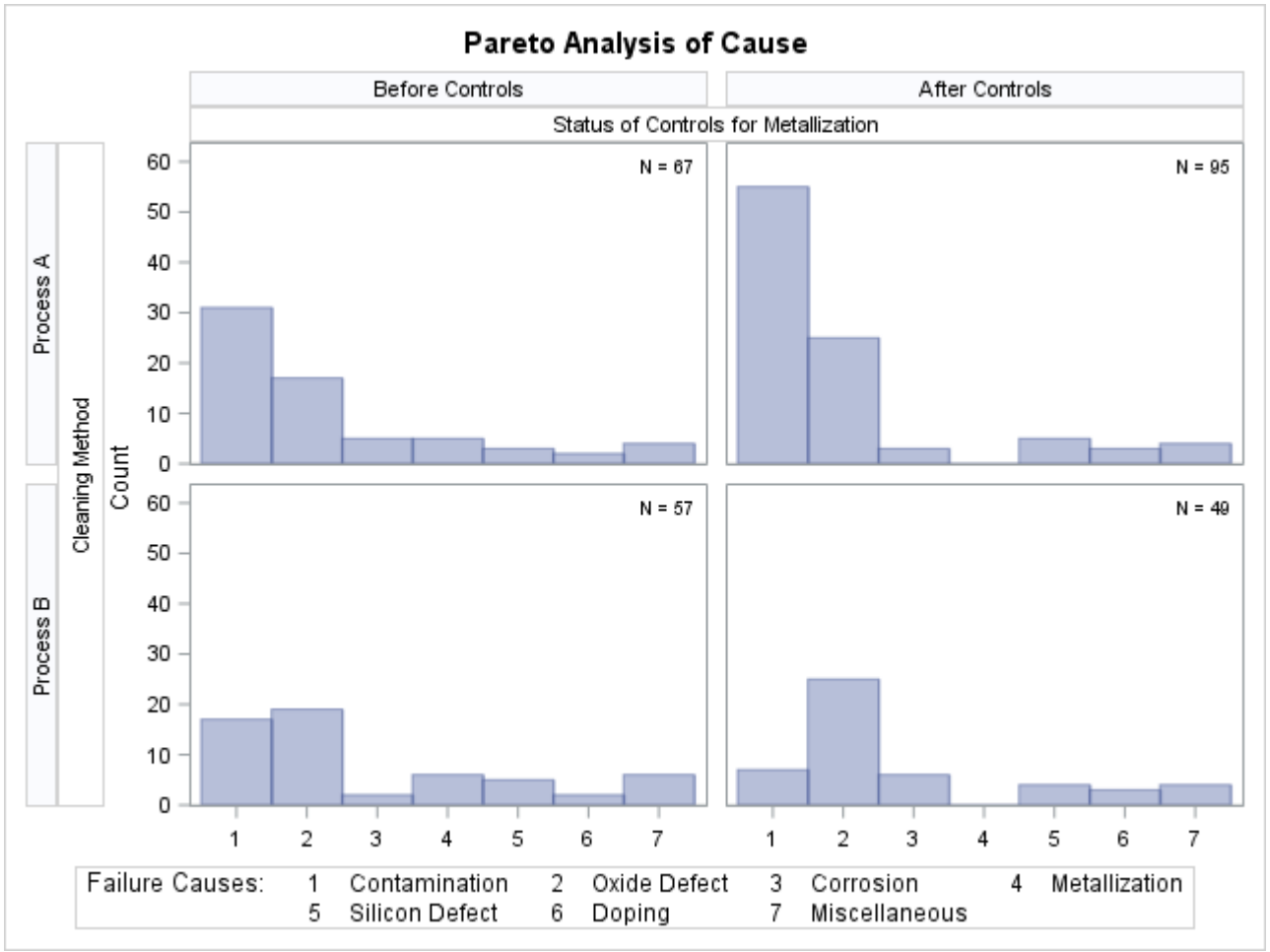
Suppose that controls for metallization were introduced on Wednesday. To show the effect of the controls, the columns for `'Monday'` and `'Tuesday'` are to be merged into a column labeled `'Before Controls'`, and the remaining columns are to be merged into a column labeled `'After Controls'`. The following statements introduce a format named `'cntlfmt'` that merges the levels of Day:

```
proc format;
   value cntlfmt   1-2 = 'Before Controls'
                   3-5 = 'After Controls';
```

The following statements create the chart shown in Output 16.7.1:

```
proc pareto data=Failure5;
   vbar Cause / class       = ( Process Day )
                freq        = Counts
                last        = 'Miscellaneous'
                scale       = count
                catleglabel = 'Failure Causes:'
                nocatlabel
                nocurve
                nlegend;
   format Day cntlfmt.;
   label  Day = 'Status of Controls for Metallization';
run;
```

Output 16.7.1 Merging Classification Levels



The levels of Day are determined by its formatted values, `'Before Controls'` and `'After Controls'`. By default, the order in which the columns are displayed is determined by the internal values. In this example, there are multiple distinct internal values for each level, and PROC PARETO uses the internal value that occurs first in the input data set.

Example 16.8: Creating Weighted Pareto Charts

NOTE: See *Pareto Analysis Based on Cost* in the SAS/QC Sample Library.

In many applications, you can quantify the priority or severity of a problem by using a measure such as the cost of repair or the loss to the customer expressed in man-hours. This example shows how to analyze such data by using a weighted Pareto chart that incorporates the cost.

Suppose that the cost associated with each of the problems in data set Failure5 (see Example 16.6) has been determined and that the costs have been converted to a relative scale. The following statements add the cost information to the data set:

```
data Failure5;
   length Analysis $ 16;
   label Analysis = 'Basis for Analysis';
   set Failure5;
   Analysis = 'Cost';
      if      Cause = 'Contamination'  then Cost = 3.0;
      else if Cause = 'Metallization'  then Cost = 8.5;
      else if Cause = 'Oxide Defect'   then Cost = 9.5;
      else if Cause = 'Corrosion'      then Cost = 2.5;
      else if Cause = 'Doping'         then Cost = 3.6;
      else if Cause = 'Silicon Defect' then Cost = 3.4;
      else                                  Cost = 1.0;
      output;
   Analysis = 'Frequency';
      Cost = 1.0;
      output;
run;
```

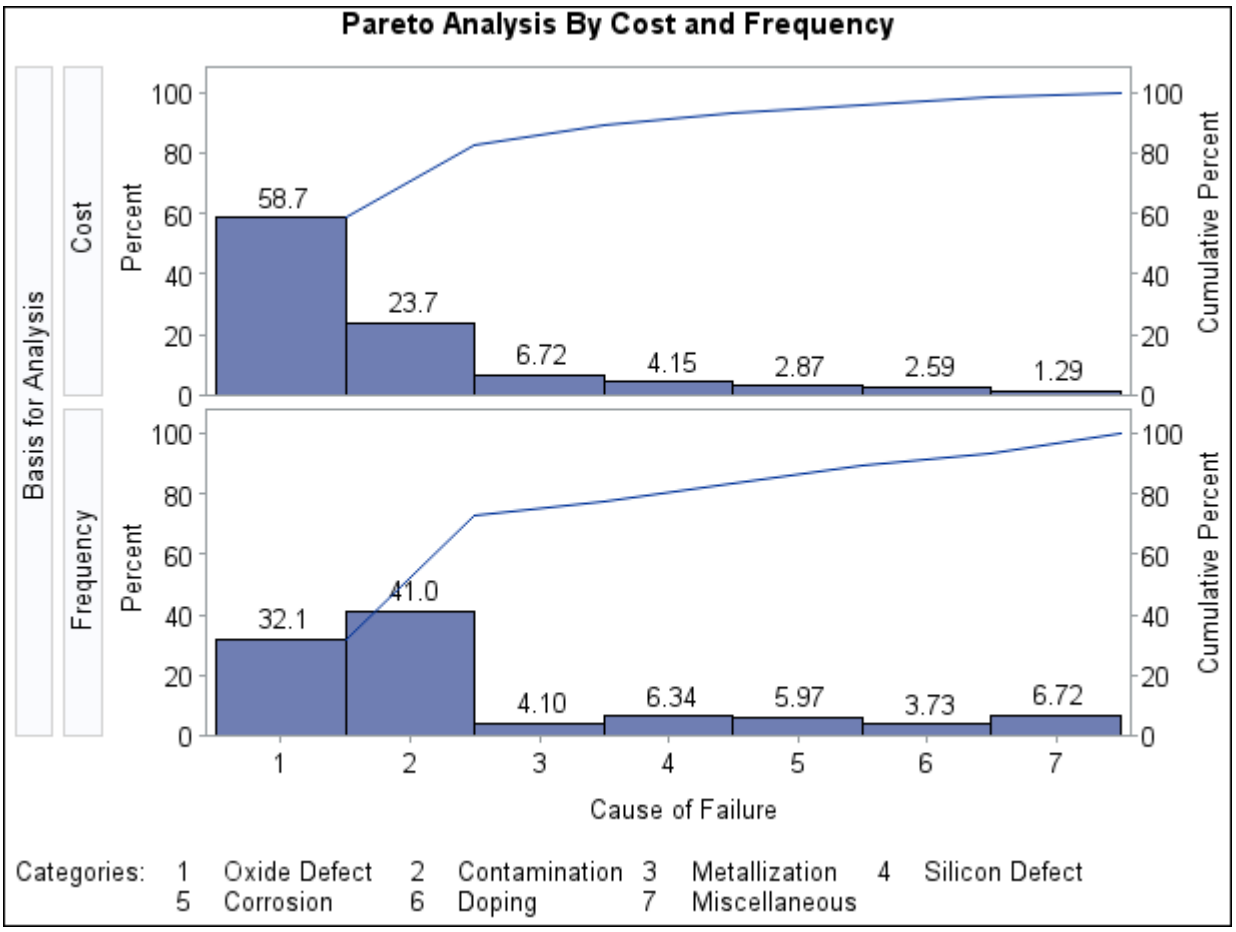
The classification variable Analysis has two levels, `'Cost'` and `'Frequency'`. For Analysis=`'Cost'`, the value of Cost is the relative cost, and for Analysis=`'Frequency'`, the value of Cost is one.

The following statements use Analysis as the classification variable to create a one-way comparative Pareto chart in which the cells are weighted Pareto charts that use Cost as the weight variable:

```
ods graphics off;
goptions vsize=4.25 in htext=2.8 pct htitle=3.2 pct;
title 'Pareto Analysis By Cost and Frequency';
proc pareto data=Failure5;
   vbar Cause / class      = ( Analysis )
                freq       = Counts
                weight     = Cost
                barlabel   = value
                out        = Summary
                intertile  = 1.0;
run;
```

The display is shown in Output 16.8.1.

Output 16.8.1 Taking Cost into Account



Within each cell, the height of a bar is the frequency of the category multiplied by the value of Cost, expressed as a percentage of the total across all categories. Thus, for the cell in which Analysis is equal to `'Frequency'`, the bars simply indicate the frequencies expressed in percentage units. This display shows that the most commonly occurring problem (contamination) is not the most expensive problem (oxide defect). The output data set Summary is listed in Output 16.8.2.

Output 16.8.2 Summary Output Data Set

Pareto Analysis By Cost and Frequency

Obs	Analysis	Cause	Cost	_COUNT_	_WCOUNT_	_PCT_	_CMPCT_
1	Cost	Oxide Defect	9.5	86	817.0	58.6799	58.680
2	Cost	Contamination	3.0	110	330.0	23.7018	82.382
3	Cost	Metallization	8.5	11	93.5	6.7155	89.097
4	Cost	Silicon Defect	3.4	17	57.8	4.1514	93.249
5	Cost	Corrosion	2.5	16	40.0	2.8729	96.122
6	Cost	Doping	3.6	10	36.0	2.5856	98.707
7	Cost	Miscellaneous	1.0	18	18.0	1.2928	100.000
8	Frequency	Oxide Defect	1.0	86	86.0	32.0896	32.090
9	Frequency	Contamination	1.0	110	110.0	41.0448	73.134
10	Frequency	Metallization	1.0	11	11.0	4.1045	77.239
11	Frequency	Silicon Defect	1.0	17	17.0	6.3433	83.582
12	Frequency	Corrosion	1.0	16	16.0	5.9701	89.552
13	Frequency	Doping	1.0	10	10.0	3.7313	93.284
14	Frequency	Miscellaneous	1.0	18	18.0	6.7164	100.000

Example 16.9: Creating Alternative Pareto Charts

NOTE: See *Alternative Pareto Charts* in the SAS/QC Sample Library.

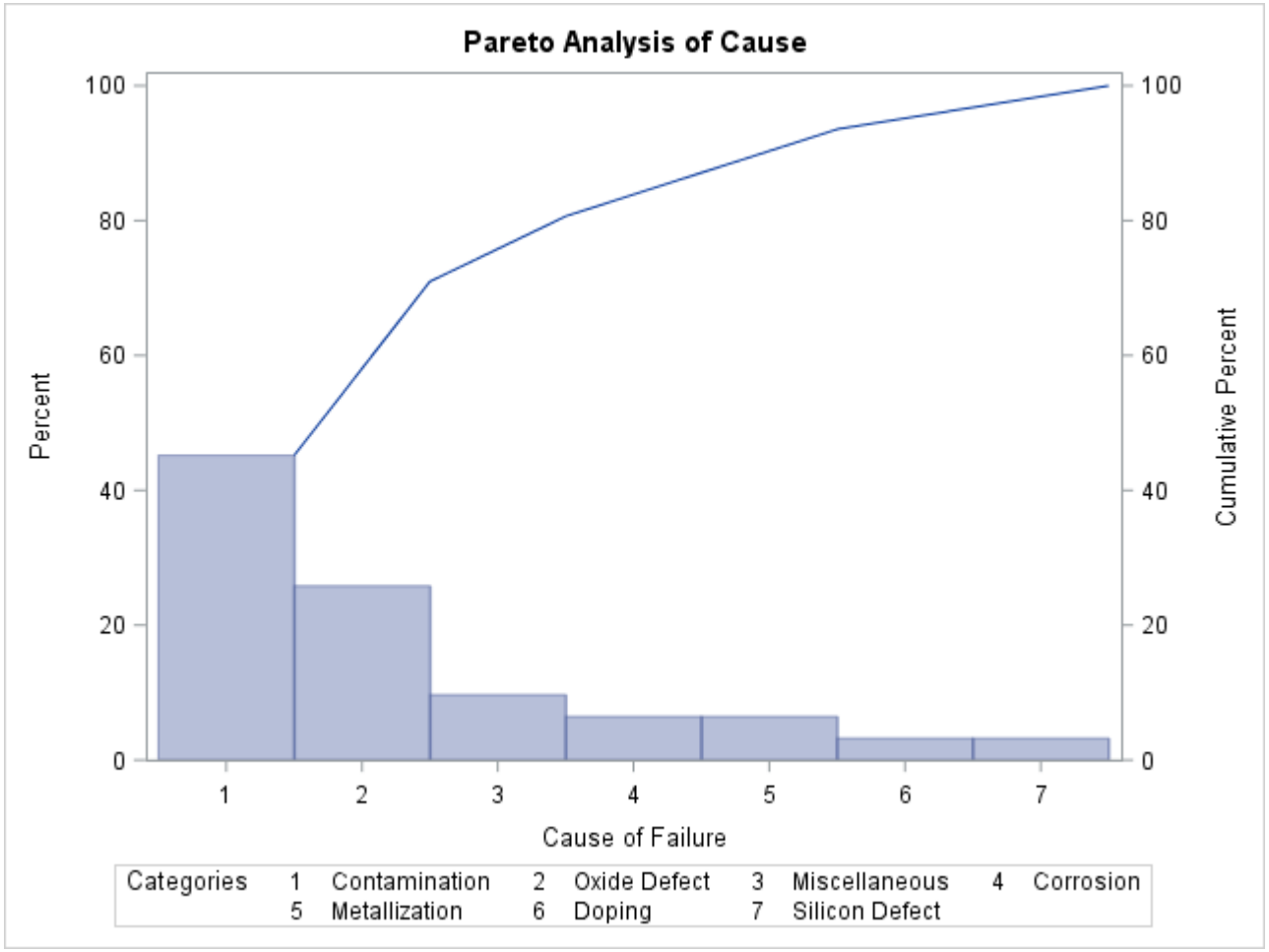
This example uses the Failure1 data set of integrated circuit fabrication failures from the section "Creating a Pareto Chart from Raw Data" on page 1069. The following statements use the CHARTTYPE= option to produce a standard Pareto chart, a cumulative Pareto bar chart, and a Pareto dot plot that includes acceptance intervals for the data:

```
proc pareto data=Failure1;
   vbar Cause;
   vbar Cause / charttype=cumulative;
   vbar Cause / charttype=intervals;
run;
```

NOTE: ODS Graphics must be enabled for you to use the CHARTTYPE= option.

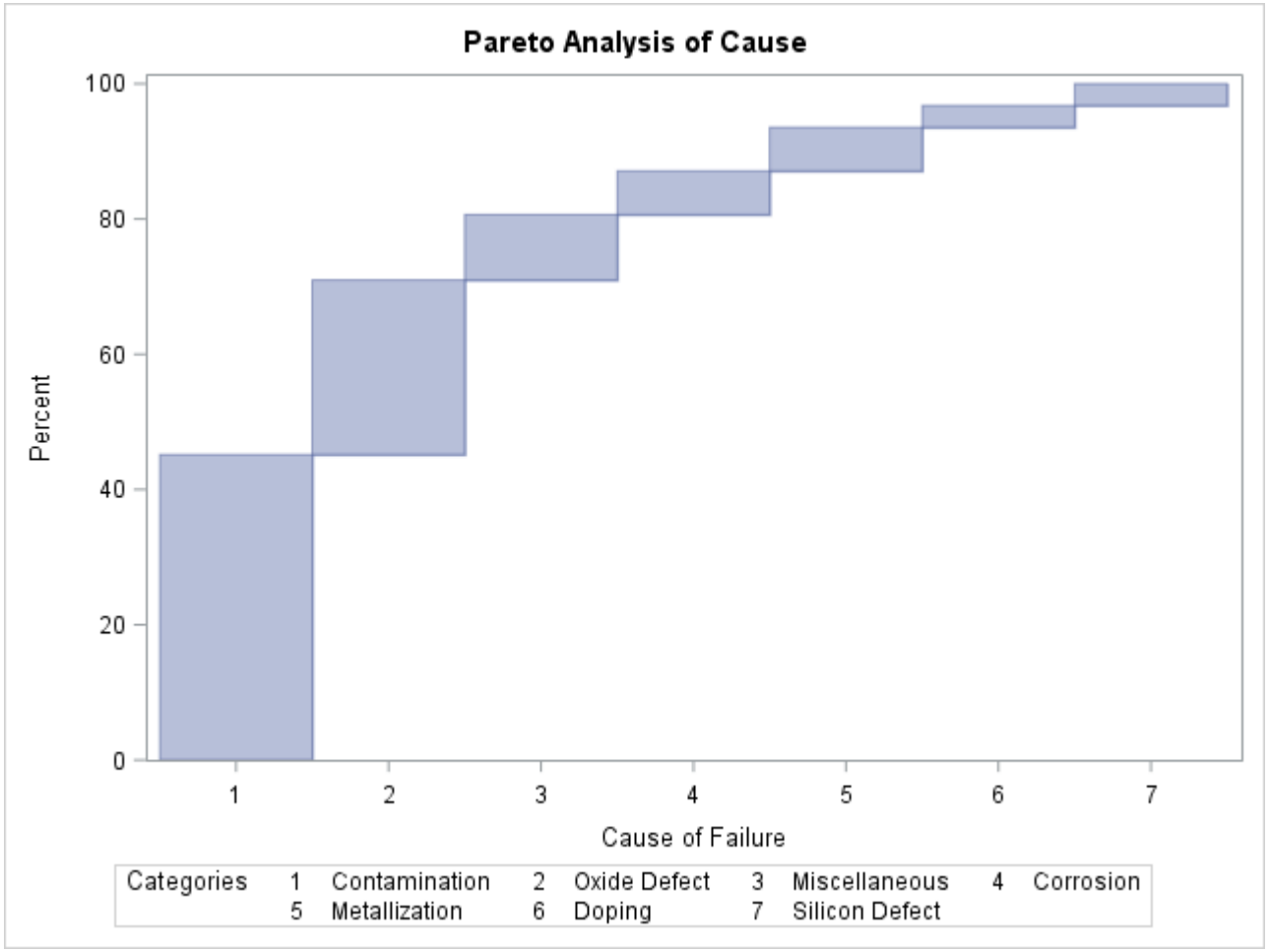
Output 16.9.1 shows the standard Pareto chart that the first VBAR statement produces.

Output 16.9.1 Standard Pareto Chart



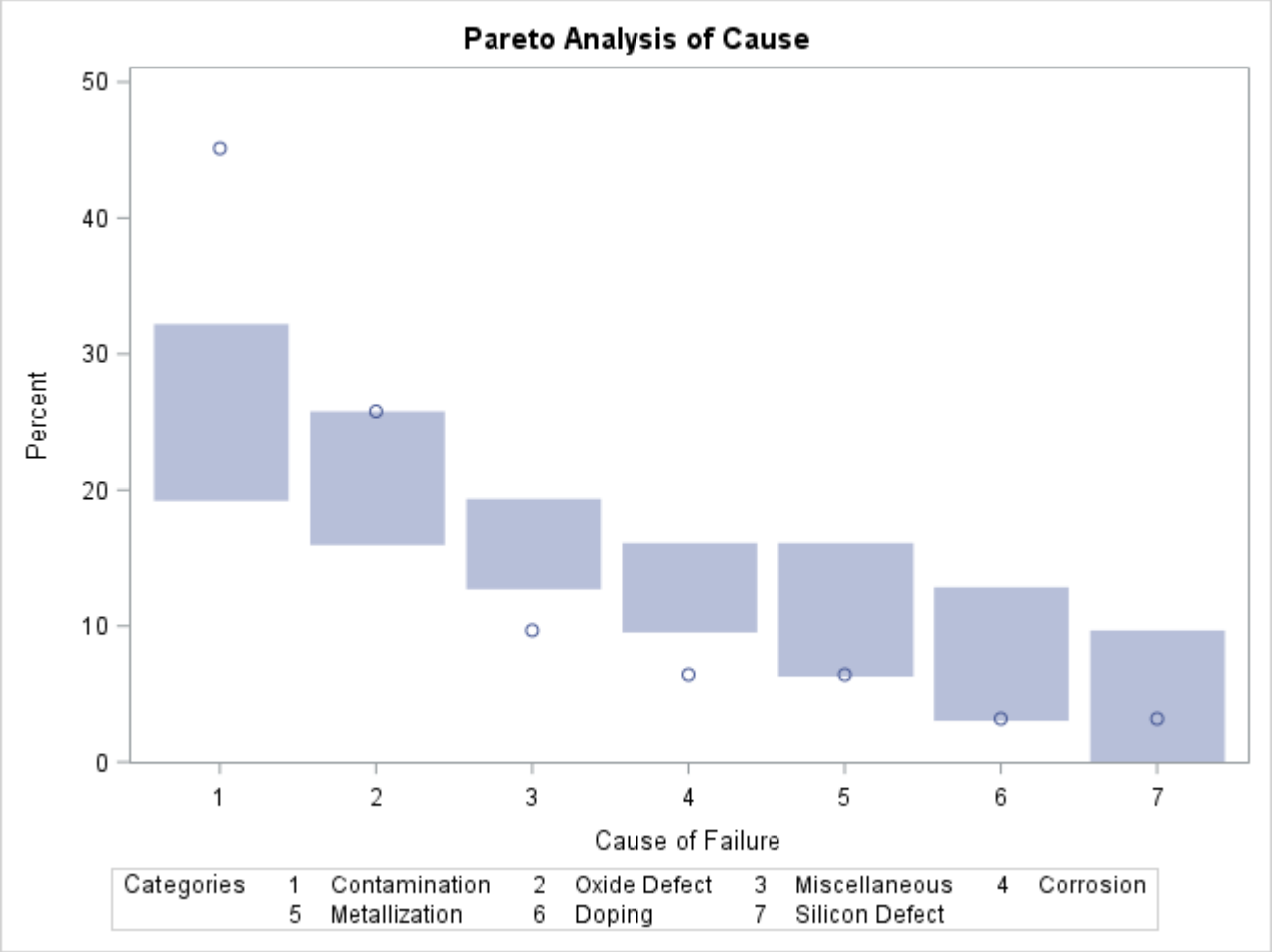
Output 16.9.2 shows the cumulative Pareto bar chart that the second VBAR statement produces.

Output 16.9.2 Cumulative Pareto Bar Chart



Output 16.9.3 shows the Pareto dot plot and acceptance intervals that the third VBAR statement produces.

Output 16.9.3 Pareto Dot Plot and Acceptance Intervals



Output 16.9.3 shows that the most frequently occurring problem, *Contamination*, occurs more frequently than the first-ranked cause from a random sample of seven uniformly distributed causes. This result indicates that addressing contamination problems should be given a high priority.

Example 16.10: Customizing Inset Labels and Formatting Values

NOTE: See *Customizing Inset Labels and Formatting Values* in the SAS/QC Sample Library.

When you add an inset to a Pareto chart, by default each inset statistic is identified by an appropriate label and its value is displayed using an appropriate format. However, you might want to provide your own labels and formats. For example, in Figure 16.7 the default label used for the N statistic is not very descriptive. The following statements produce a comparative Pareto chart whose insets display longer labels for both statistics. A format that uses one decimal place is also specified for each statistic. (These are integer values—the decimals are added only to demonstrate this feature.) Note that a single INSET statement produces an inset in each cell of the comparative Pareto chart.

```
proc pareto data=Failure3;
   vbar Cause /
      class    = Stage
      freq     = Counts
      maxncat  = 5
      classkey = 'Before Cleaning';
   inset n    ='Observations Shown' (4.1)
         nexcl='Observations Excluded' (3.1);
run;
```

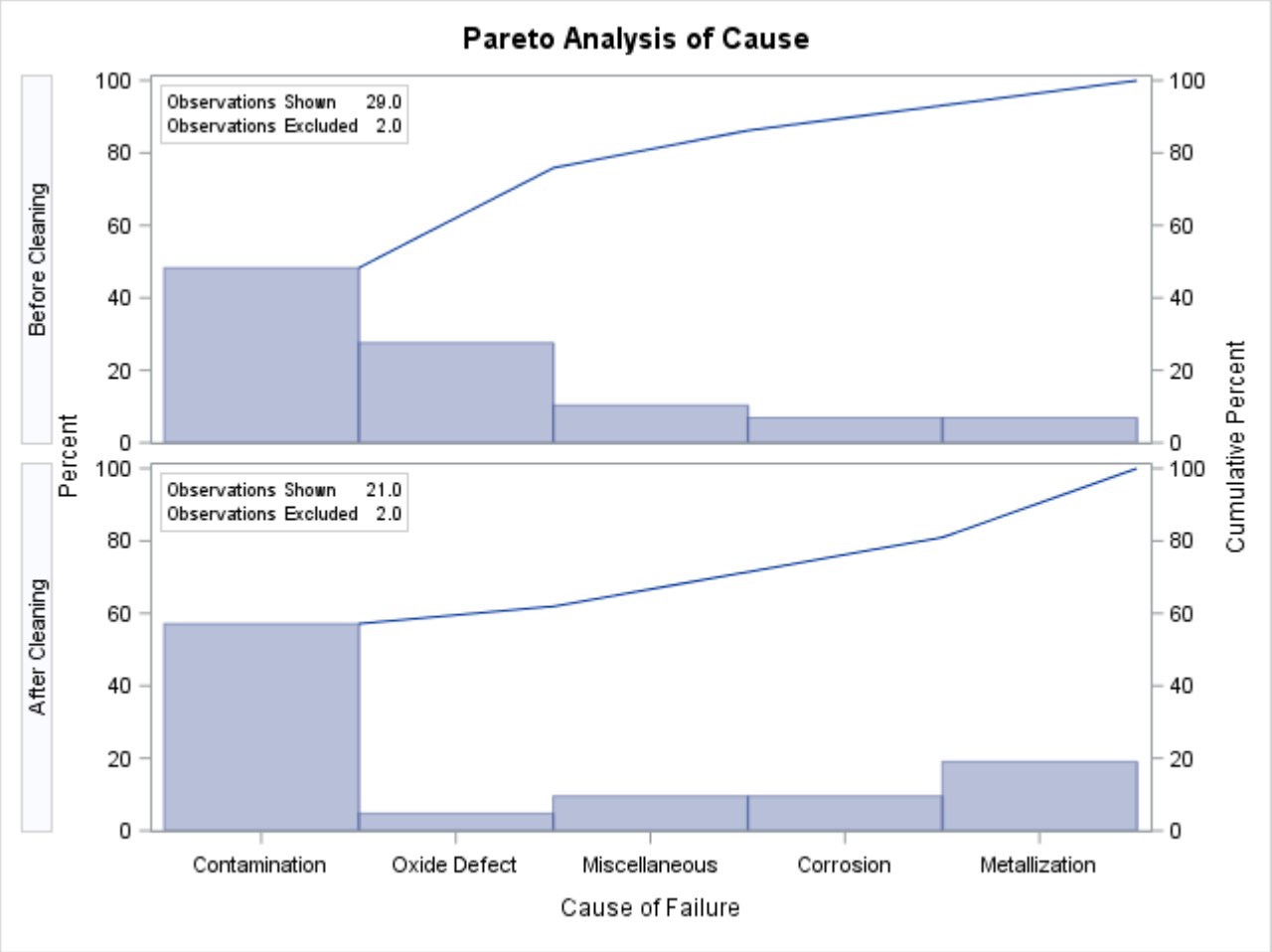
The resulting chart is displayed in Output 16.10.1.

You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotation marks. The label can have up to 24 characters.

The format 4.1 specified in parentheses after the N keyword displays the statistics by using a field width of four and one decimal place. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also use the FORMAT= option to specify a format to be used for all the statistics in the INSET statement. For more information about SAS formats, see *SAS Formats and Informats: Reference*.

NOTE: If you specify both a label and a format for a statistic, the label must appear before the format.

Output 16.10.1 Customizing Labels and Formatting Values in an Inset



Example 16.11: Specifying Inset Headers and Positions

NOTE: See *Specifying Inset Headers and Positions* in the SAS/QC Sample Library.

By default, the first INSET statement that is specified after a chart statement displays an inset in the upper left corner of the chart. You can control the inset position by specifying the POSITION= option. In addition, you can display a header at the top of the inset by specifying the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and produce the horizontal Pareto chart shown in Output 16.11.1:

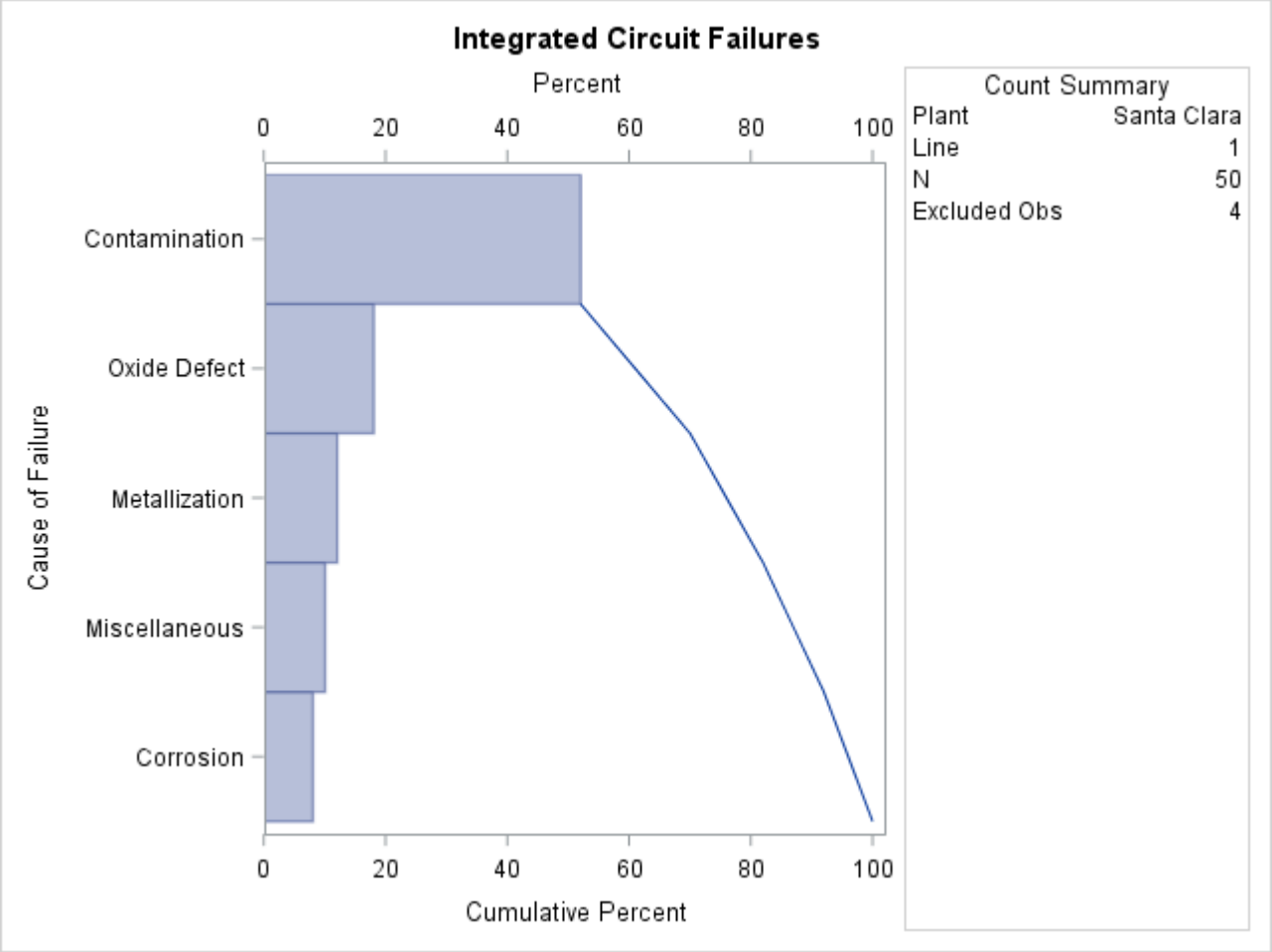
```
data location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
datalines;
Plant     Santa Clara
Line      1
;

title 'Integrated Circuit Failures';
proc pareto data=Failure3;
   hbar Cause /
      freq     = Counts
      maxncat  = 5
      odstitle = title;
   inset data = location n nexcl /
      position = rm
      header   = 'Count Summary';
run;
```

The header (in this case, "Count Summary") can be up to 40 characters. The POSITION=RM option is specified to position the inset in the right margin so that it does not interfere with features of the chart. For more information about positioning, see the section "Positioning Insets" on page 1121.

INSET statement options, such as the POSITION= and HEADER= options, are specified after the slash (/). For more information about INSET statement options, see the section "INSET Statement Options" on page 1087.

Note that the contents of the data set location appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset.

Output 16.11.1 Adding a Header and Repositioning the Inset



Example 16.12: Managing a Large Number of Categories

NOTE: See *Managing a Large Number of Categories* in the SAS/QC Sample Library.

The Centers for Disease Control publish a variety of public health statistics. The numbers of deaths in 2010 in the United States that were caused by various types of cancer are recorded in the SAS data set CancerDeaths2010:

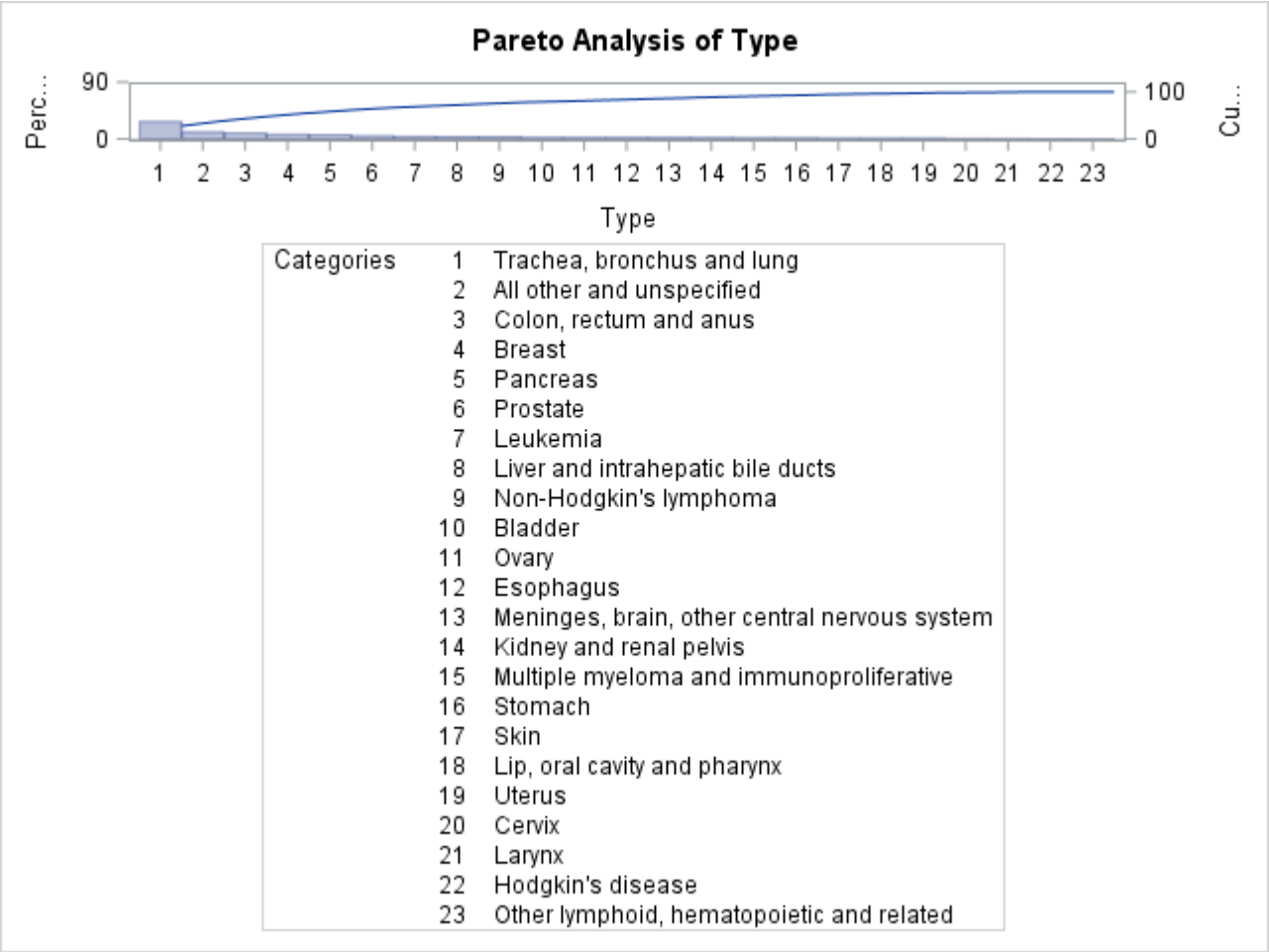
```
data CancerDeaths2010;
   length Type $ 45;
   input Type & @47 Deaths comma7.;
   datalines;
Lip, oral cavity and pharynx                     8,474
Esophagus                                       14,490
Stomach                                         11,390
Colon, rectum and anus                          52,622
Liver and intrahepatic bile ducts               20,305
Pancreas                                        36,888
Larynx                                           3,691
Trachea, bronchus and lung                     158,318
Skin                                             9,154
Breast                                          41,435
Cervix                                           3,939
Uterus                                           8,402
Ovary                                           14,572
Prostate                                        28,561
Kidney and renal pelvis                         13,219
Bladder                                         14,731
Meninges, brain, other central nervous system   14,164
Hodgkin's disease                                1,231
Non-Hodgkin's lymphoma                          20,294
Leukemia                                        22,569
Multiple myeloma and immunoproliferative        11,428
Other lymphoid, hematopoietic and related           68
All other and unspecified                       64,798
;
```

The following statements produce a Pareto chart for the data in CancerDeaths2010:

```
proc pareto data=CancerDeaths2010;
   vbar Type / freq = Deaths;
run;
```

The resulting Pareto chart is shown in Output 16.12.1.

Output 16.12.1 Cancer Deaths Pareto Chart with Default Width

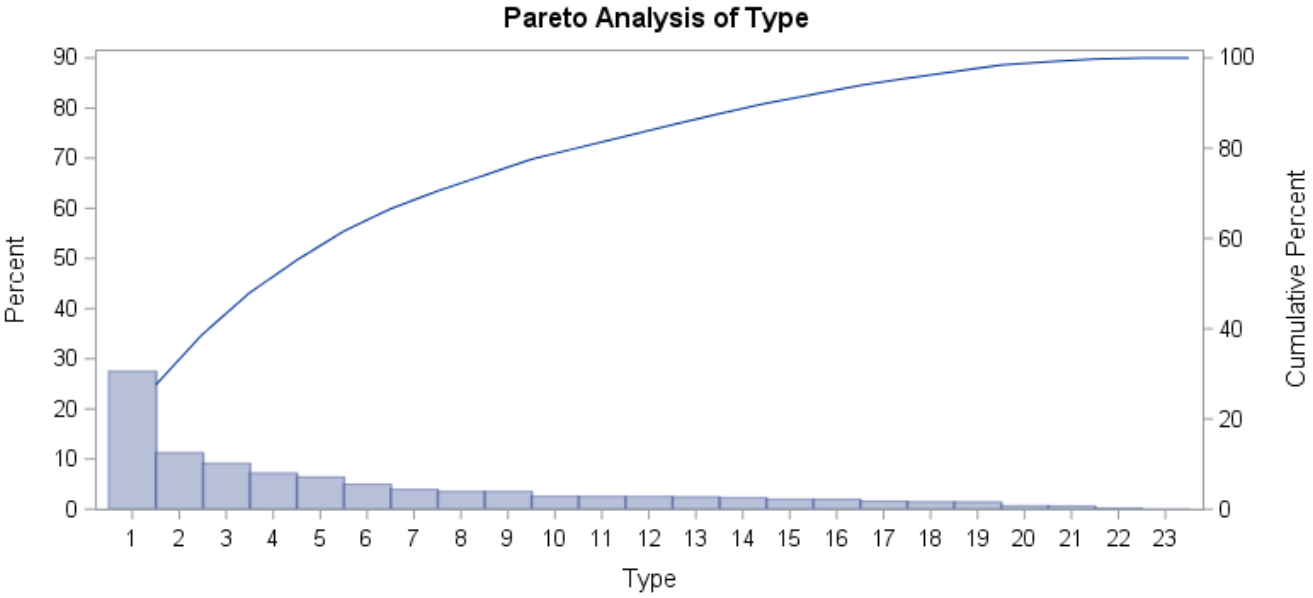


Note that PROC PARETO has labeled the category axis tick marks with numbers and produced a corresponding category legend. This is done by default when there is not enough room to use category names as tick labels on the category axis. Unfortunately, because some of the category names are long, the legend has room for only one column of entries and therefore occupies an inordinate amount of space. Among the alternatives for addressing this problem are the following:

- replacing the original category names with shorter ones
- increasing the space available for the graph

You can implement the second alternative by specifying the WIDTH= option in the ODS GRAPHICS statement prior to invoking the procedure. (The ODS GRAPHICS statement is documented in the *SAS Output Delivery System: User's Guide*.) Output 16.12.2 shows the Pareto chart that is produced after the graph width is increased.

Output 16.12.2 Cancer Deaths Pareto Chart with Increased Width



Categories					
	1	Trachea, bronchus and lung	2	All other and unspecified
	3	Colon, rectum and anus	4	Breast
	5	Pancreas	6	Prostate
	7	Leukemia	8	Liver and intrahepatic bile ducts
	9	Non-Hodgkin's lymphoma	10	Bladder
	11	Ovary	12	Esophagus
	13	Meninges, brain, other central nervous system	14	Kidney and renal pelvis
	15	Multiple myeloma and immunoproliferative	16	Stomach
	17	Skin	18	Lip, oral cavity and pharynx
	19	Uterus	20	Cervix
	21	Larynx	22	Hodgkin's disease
	23	Other lymphoid, hematopoietic and related		

In a standard Pareto chart, the cumulative percentage curve is anchored at the top of the first category bar. In Output 16.12.2 PROC PARETO has automatically relaxed that rule to avoid excessive compression of the bars. You can use the FREQAXIS= option to specify that the frequency axis extend to 100%, which restores the anchoring of the curve. (For more information about scaling the frequency and cumulative percentage axes, see the section "Scaling the Cumulative Percentage Curve" on page 1120.)

Note also in Output 16.12.2 that the category `'All other and unspecified'` has the second highest frequency. To better indicate the specific types of cancer responsible for the most deaths, you can use the LAST= option to display the `'All other and unspecified'` category last.

The following statements incorporate these changes and add other enhancements to the chart:

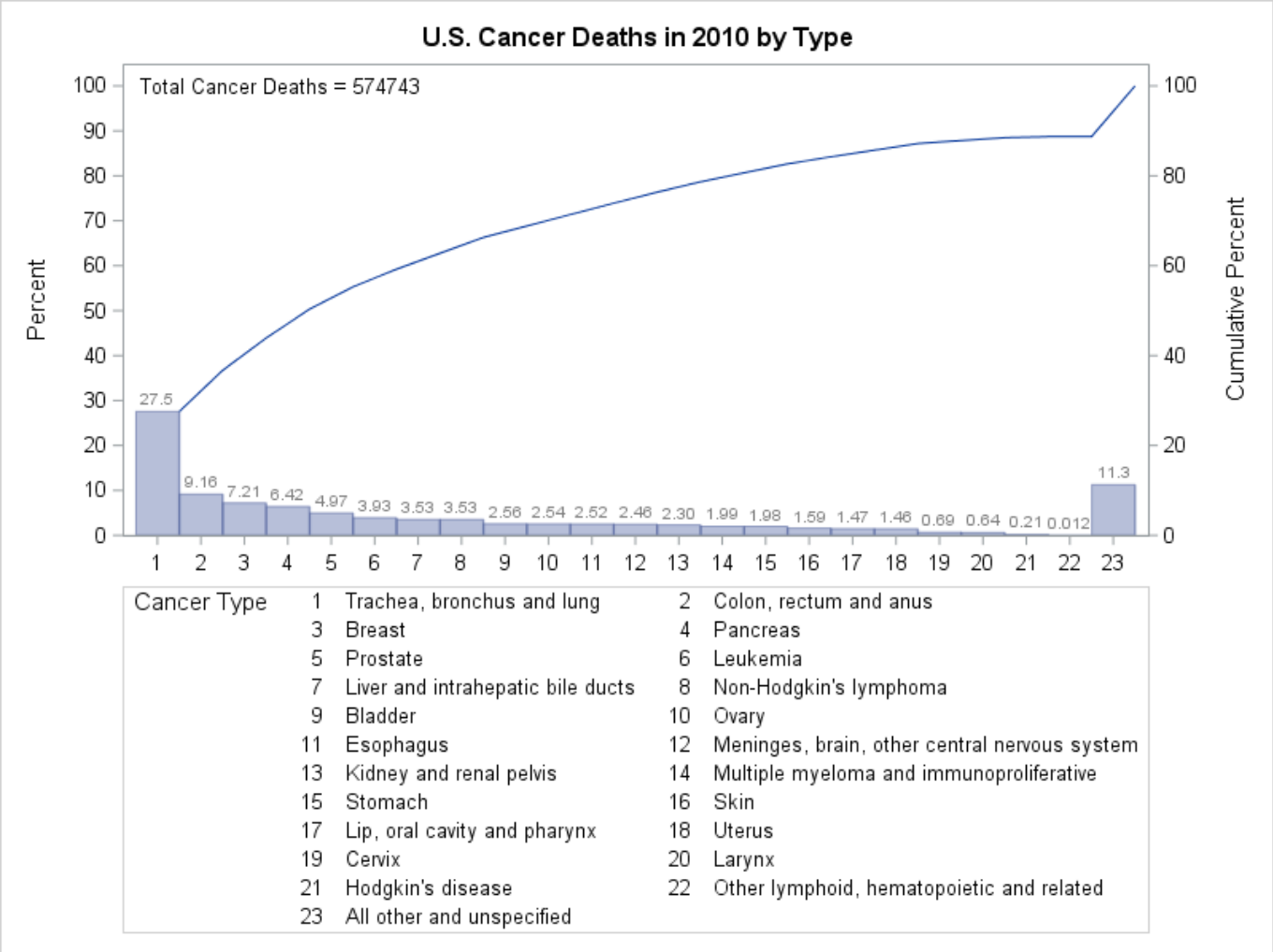
```
ods graphics / width=800px;
title 'U.S. Cancer Deaths in 2010 by Type';
proc pareto data=CancerDeaths2010;
   vbar Type / freq       = Deaths
               barlabel   = value
               last       = 'All other and unspecified'
               nocatlabel
```

```
                catleglabel = 'Cancer Type'
                freqaxis    = 0 to 100 by 10
                nlegend     = 'Total Cancer Deaths'
                odstitle    = title
                out         = CSummary;
                ;
run;
```

The BARLABEL= option labels each bar with its value in frequency axis units, which in this case is the percentage of cancer deaths that were caused by that type of cancer. The NOCATLABEL option saves some space by eliminating the category axis label, and the CATLEGLABEL= option produces a more informative label for the category legend. The NLEGEND= option displays the total sample size with an appropriate label. The ODSTITLE= option replaces the default graph title with the one specified in the TITLE statement. The OUT= option saves a summary of the Pareto chart in the data set CSummary.

The improved Pareto chart is shown in Output 16.12.3, and a listing of CSummary is shown in Output 16.12.4.

Output 16.12.3 Improved Pareto Chart of 2010 Cancer Deaths



Output 16.12.4 CSummary Data Set

U.S. Cancer Deaths in 2010 by Type

Obs	Type	_COUNT_	_PCT_	_CMPCT_
1	Trachea, bronchus and lung	158318	27.5459	27.546
2	Colon, rectum and anus	52622	9.1557	36.702
3	Breast	41435	7.2093	43.911
4	Pancreas	36888	6.4182	50.329
5	Prostate	28561	4.9694	55.298
6	Leukemia	22569	3.9268	59.225
7	Liver and intrahepatic bile ducts	20305	3.5329	62.758
8	Non-Hodgkin's lymphoma	20294	3.5310	66.289
9	Bladder	14731	2.5631	68.852
10	Ovary	14572	2.5354	71.388
11	Esophagus	14490	2.5211	73.909
12	Meninges, brain, other central nervous system	14164	2.4644	76.373
13	Kidney and renal pelvis	13219	2.3000	78.673
14	Multiple myeloma and immunoproliferative	11428	1.9884	80.661
15	Stomach	11390	1.9818	82.643
16	Skin	9154	1.5927	84.236
17	Lip, oral cavity and pharynx	8474	1.4744	85.710
18	Uterus	8402	1.4619	87.172
19	Cervix	3939	0.6853	87.858
20	Larynx	3691	0.6422	88.500
21	Hodgkin's disease	1231	0.2142	88.714
22	Other lymphoid, hematopoietic and related	68	0.0118	88.726
23	All other and unspecified	64798	11.2743	100.000

The Pareto chart in Output 16.12.3 has 23 categories, some of which account for only a small percentage of the total deaths. Often only a relatively few categories that have the highest frequencies are of interest. The PARETO procedure provides options for limiting the number of categories that are displayed on a chart. For an example of restricting the number of categories by using the MAXNCAT= and OTHER= options, see the section "Restricting the Number of Pareto Categories" on page 1074.

The original CancerDeaths2010 data set appears to have been summarized in advance, with the `'All other and unspecified'` category containing the total count for unspecified cancers plus those types that account for fewer deaths than the 22 distinct types that are shown in Output 16.12.3. The `'All other and unspecified'` category has the second highest frequency, accounting for 11.3% of all deaths.

The chart statement options that limit the number of categories to be displayed omit or merge low-frequency categories. In this case, it is more useful to merge the low-frequency categories into the existing `'All other and unspecified'` category. The following DATA step merges each type that accounts for less than 2% of cancer deaths into the `'All other and unspecified'` category:

```
data CSummary;
   set CSummary;
   if _PCT_ < 2.0 then Type='All other and unspecified';
run;
```

The modified CSummary data set is shown in Output 16.12.5.

Output 16.12.5 Modified CSummary Data Set

U.S. Cancer Deaths in 2010 by Type

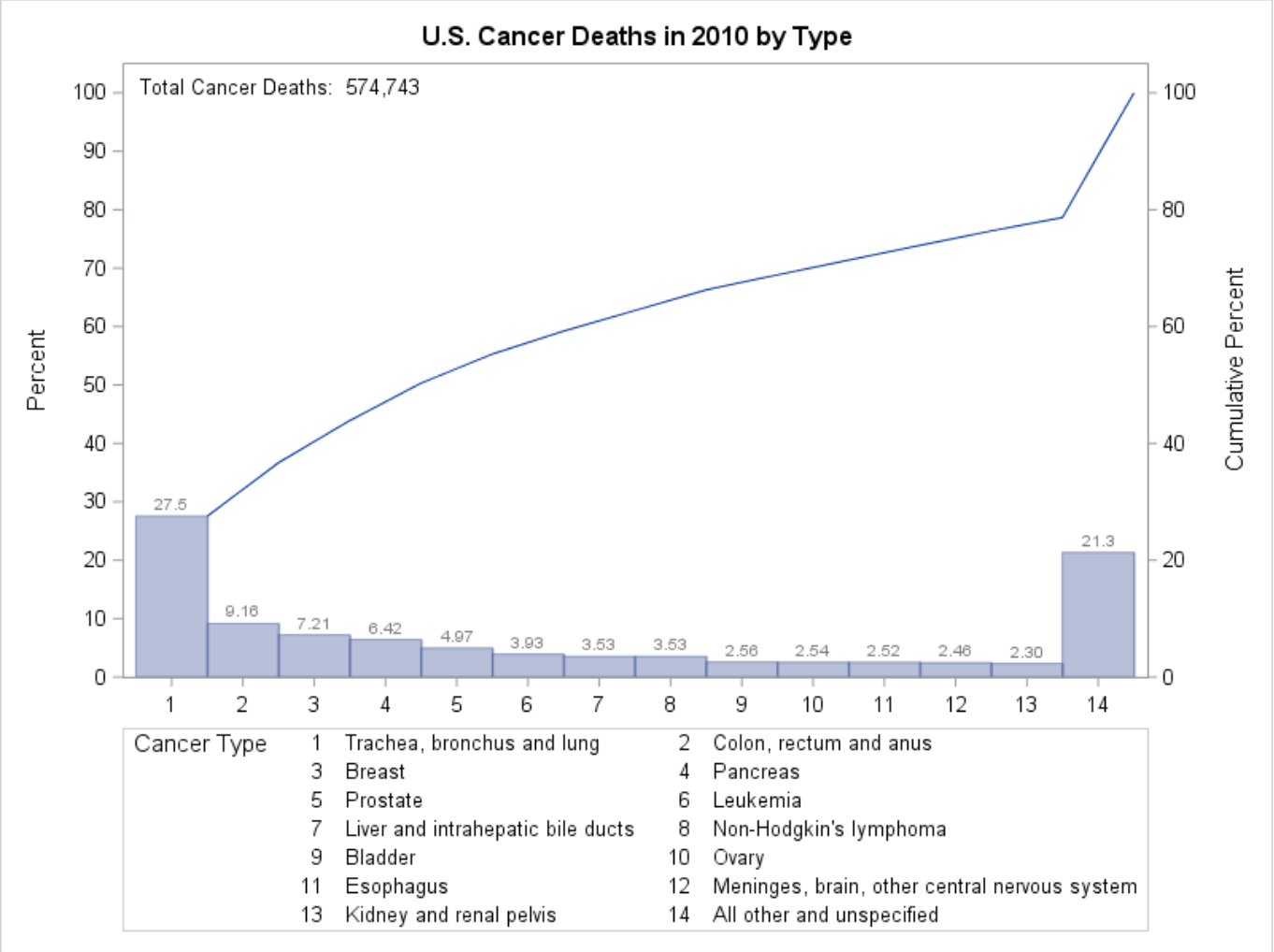
Obs	Type	_COUNT_	_PCT_	_CMPCT_
1	Trachea, bronchus and lung	158318	27.5459	27.546
2	Colon, rectum and anus	52622	9.1557	36.702
3	Breast	41435	7.2093	43.911
4	Pancreas	36888	6.4182	50.329
5	Prostate	28561	4.9694	55.298
6	Leukemia	22569	3.9268	59.225
7	Liver and intrahepatic bile ducts	20305	3.5329	62.758
8	Non-Hodgkin's lymphoma	20294	3.5310	66.289
9	Bladder	14731	2.5631	68.852
10	Ovary	14572	2.5354	71.388
11	Esophagus	14490	2.5211	73.909
12	Meninges, brain, other central nervous system	14164	2.4644	76.373
13	Kidney and renal pelvis	13219	2.3000	78.673
14	All other and unspecified	11428	1.9884	80.661
15	All other and unspecified	11390	1.9818	82.643
16	All other and unspecified	9154	1.5927	84.236
17	All other and unspecified	8474	1.4744	85.710
18	All other and unspecified	8402	1.4619	87.172
19	All other and unspecified	3939	0.6853	87.858
20	All other and unspecified	3691	0.6422	88.500
21	All other and unspecified	1231	0.2142	88.714
22	All other and unspecified	68	0.0118	88.726
23	All other and unspecified	64798	11.2743	100.000

Note that although CSummary contains frequency data, it can contain multiple observations that have the same category value. The following statements create a Pareto chart from the modified CSummary data set:

```
proc pareto data=CSummary;
   vbar Type / freq        = _COUNT_
               last        = 'All other and unspecified'
               barlabel    = value
               nocatlabel
               catleglabel = 'Cancer Type'
               freqaxis    = 0 to 100 by 10
               odstitle    = title;
   inset n='Total Cancer Deaths:'(comma7.) / noframe;
run;
```

Note that the sample size legend in Output 16.12.3 displays the sample size as an unformatted integer. By using an INSET statement instead of the NLEGEND= option, you can specify a format for the sample size. (For a complete description of the INSET statement, see the section "INSET Statement" on page 1085.) The resulting chart is shown in Output 16.12.6.

Output 16.12.6 Cancer Deaths Pareto Chart with Fewer Categories



Output 16.12.6 shows that 21.3% of deaths are assigned to `'All other and unspecified'` and that the bar frequencies sum to 100%.

References

Burr, J. T. (1990). "The Tools of Quality, Part 6: Pareto Charts." *Quality Progress* 23:59–61.

Cleveland, W. S. (1985). *The Elements of Graphing Data*. Monterey, CA: Wadsworth.

Ishikawa, K. (1976). *Guide to Quality Control*. Tokyo: Asian Productivity Organization.

Kume, H. (1985). *Statistical Methods for Quality Improvement*. Tokyo: AOTS Chosakai.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986). *Modern Methods for Quality Control and Improvement*. New York: John Wiley & Sons.

Wilkinson, L. (2006). "Revising the Pareto Chart." *American Statistician* 60:332–334.

Chapter 17
The RAREEVENTS Procedure

Contents

Overview: RAREEVENTS Procedure **1169**
Rare Events Charts and *c* Charts 1170
Getting Started: RAREEVENTS Procedure **1172**
Syntax: RAREEVENTS Procedure **1175**
PROC RAREEVENTS Statement 1176
BY Statement 1176
ID Statement 1177
CHART Statement 1177
COMPARE Statement 1179
Common CHART and COMPARE Statement Options 1180
Details: RAREEVENTS Procedure **1183**
Constructing Rare Events Charts 1183
EDF Goodness-of-Fit Tests 1184
Input Data Sets 1187
Output Data Sets 1189
ODS Table Names 1190
ODS Graphics 1191
Examples: RAREEVENTS Procedure **1191**
Example 17.1: Monitoring Urinary Tract Infections 1191
Example 17.2: Airline Crashes 1194
References **1201**

Overview: RAREEVENTS Procedure

The RAREEVENTS procedure produces control charts for rare events. A control chart is a graphical and analytical tool for detecting unusual variation in a process and deciding whether the process is stable and predictable. A rare event is one that occurs infrequently, with a low probability.

In this chapter, a control chart for rare events is referred to as a *rare events chart*. The data that are plotted in a rare events chart represent the times between successive events. Usually these are adverse events that are unwanted outcomes in a process, such as an incorrectly recorded bank deposit, a patient falling in a hospital, or a chemical spill. Rare events charts have gained acceptance in health care quality improvement applications because of their ease of use and suitability to processes that have low defect rates (Benneyan 1999).

An important assumption for a rare events chart is that the events are independent. The occurrence of one event does not affect the probability that another will occur, and the probability of an occurrence is approximately constant over time. Rare events charts should not be used to monitor clusters of events, such as cases of a contagious disease, which violate this assumption. See Woodall (2006) for a thorough discussion of different control charts that are applicable to health care quality improvement.

The data for a rare events chart are often the times between consecutive events, such as the intervals between accidental needle sticks in a hospital. The intervals can be recorded as integer or continuous values. The opportunities for events to occur must be approximately constant over time. For example, the number of times that needles are handled should be about the same each day if you are monitoring the number of days between accidental sticks. Alternatively, the data can be explicit counts of opportunities for occurrence that come between events, such as the number of surgeries performed between occurrences of postsurgical infection. These kinds of data are preferable but often are not available.

A rare events chart has two decision limits: an upper probability limit (UPL) and a lower probability limit (LPL). By default, these are based on a geometric distribution for integer data and an exponential distribution for continuous data. A data value that is greater than the UPL or less than the LPL signals unusual variation in the process. A value that is greater than the UPL indicates that the time between events might be increasing, in which case the events are occurring less frequently. Because the events of interest are usually adverse, this can signal an improvement in the process. Conversely, a value less than the LPL indicates that events are occurring more frequently, which can signal a decline in the process.

You can use the RAREEVENTS procedure to do the following:

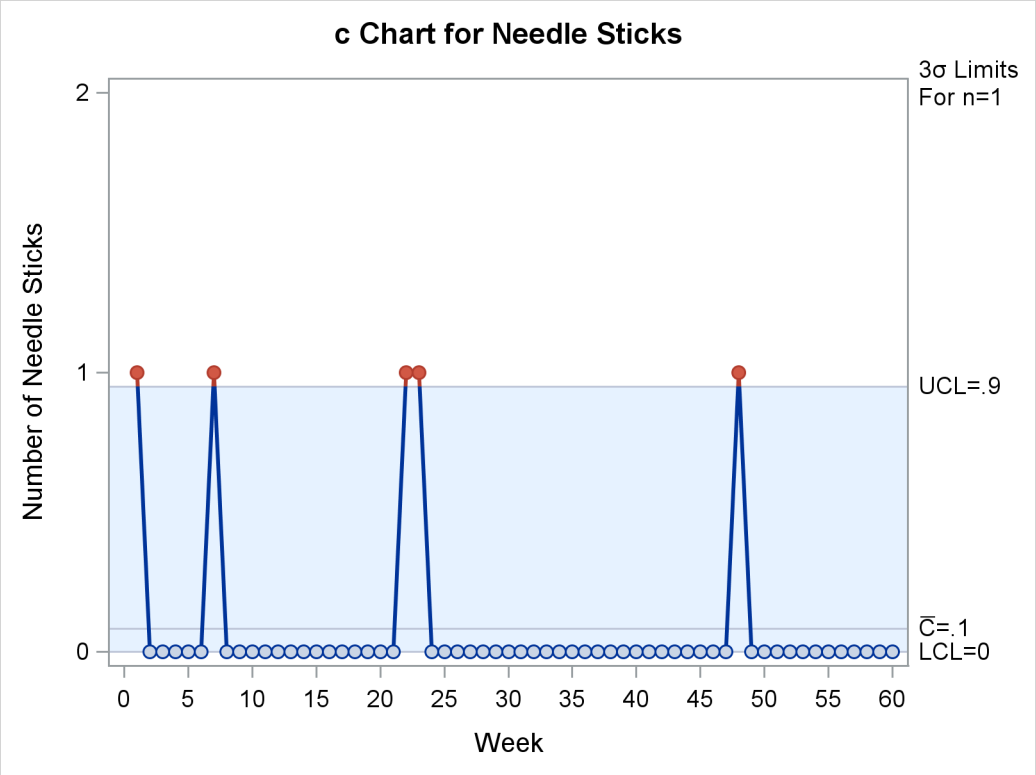
- produce a rare events chart with probability limits that are computed from the data
- create a graph that you can use to compare the distribution of the input data with a reference probability distribution
- specify the probability distribution that is used to compute the probability limits or to compare with the input data
- save probability limits in an output data set
- produce a rare events chart that uses preestablished probability limits that are read from a data set
- save process measurements, probability limits, and probability distribution information in an output data set

Rare Events Charts and *c* Charts

The traditional control chart that is most comparable to a rare events chart is the *c* chart, which is used to monitor counts of unwanted process outcomes. (See the section "CCHART Statement: SHEWHART Procedure" on page 1482 for a detailed description of *c* charts and how to produce them by using the SHEWHART procedure.) However, as explained by Kaminsky et al. (1992), Benneyan (2001a), and others, *c* charts and other traditional control charts do not always perform well when used to monitor rare events.

Figure 17.1 shows a *c* chart of needle sticks per week in a hospital. Weeks are identified on the horizontal axis, and the weekly counts of needle sticks are plotted. The control limits are used to detect unusual variation in the number of needle sticks.

Figure 17.1 c Chart for Needle Sticks per Week

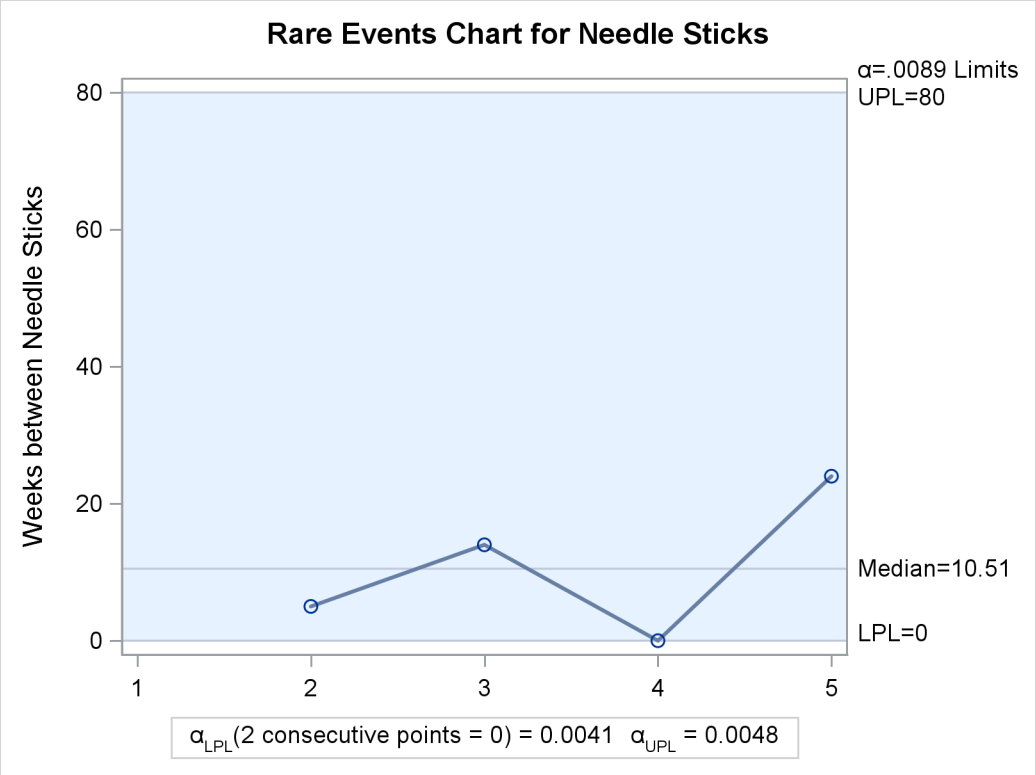


In this case needle sticks are truly rare. Almost all the weekly counts are 0, no week has more than 1, and the mean count is very low. Because the upper control limit (UCL) is less than 1, each individual needle stick signals unusual variation. This c chart might be too sensitive to provide useful information.

To address this problem, you could increase the counts (and therefore the UCL) by increasing the length of the time periods over which you accumulate the counts. In this case, grouping the needle stick counts into 15 four-week periods produces a c chart with $\mathrm{UCL} = 2.07$ and a maximum count of 2, and no unusual variation is signaled. A drawback of this approach is that data are available for analysis only every four weeks, so a change in the process might not be detected quickly. Another possibility would be to modify the way that the control limits are computed by basing them on a discrete distribution other than the Poisson distribution. A better alternative is to use a rare events chart.

Figure 17.2 shows a rare events chart that is used to plot the same needle stick data, which are transformed into weeks between sticks. Individual needle sticks are identified in order of occurrence on the horizontal axis. For each event, the time in weeks since the previous event is plotted.

Figure 17.2 Rare Events Chart for Weeks between Needle Sticks



The rare events chart does not signal any unusual variation. There would be a signal if two consecutive data values were zero, indicating needle sticks in three consecutive weeks. For more information, see the section "Probability Limits Based on a Geometric Distribution" on page 1183.

When you use a rare events chart, you do not need to wait until the end of a reporting period or collect a large sample of data before plotting a point on the chart. Instead, you can add a point to the chart immediately when an event occurs. Therefore you can construct a useful chart in a more timely manner, which improves your chances of detecting process changes. Because the values that are plotted in a rare events chart are times between events, the simple occurrence of a single event will not signal unusual variation. In summary, a rare events chart is better suited than traditional control charts to detecting changes in the frequency of low-probability events.

Getting Started: RAREEVENTS Procedure

This example illustrates the basic features of the RAREEVENTS procedure. The data are adapted from Benneyan (1998b). The following statements create a SAS data set named Infections by reading the dates of occurrences of an infectious disease and computing DaysBetween, the numbers of days between successive infections:

```
data Infections;
   input InfectionDate mmddyy10.;
   InfectionNumber = _n_;
   DaysBetween = InfectionDate - lag(InfectionDate);
   format InfectionDate  mmddyy10.;
datalines;
04/17/1995
04/17/1995
04/17/1995
04/19/1995
04/20/1995
05/03/1995
05/05/1995
05/05/1995
05/06/1995
05/07/1995
05/08/1995
05/09/1995
05/09/1995
05/10/1995
05/11/1995
05/27/1995
05/27/1995
05/28/1995
05/29/1995
05/31/1995
06/10/1995
06/11/1995
06/12/1995
06/14/1995
06/16/1995
06/16/1995
06/18/1995
06/21/1995
06/21/1995
;
```

Figure 17.3 shows a partial listing of the Infections data set.

Figure 17.3 Partial Listing of the Infections Data Set

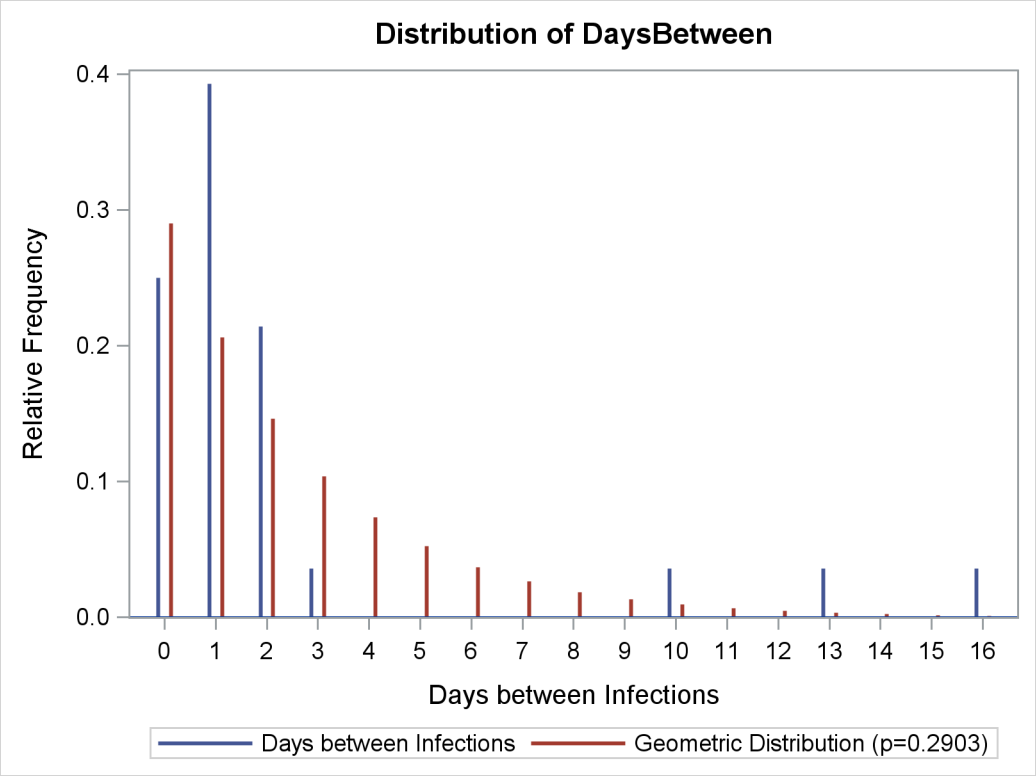
InfectionDate	InfectionNumber	DaysBetween
04/17/1995	1	.
04/17/1995	2	0
04/17/1995	3	0
04/19/1995	4	2
04/20/1995	5	1
05/03/1995	6	13
05/05/1995	7	2

The following statements produce a comparison plot and a rare events chart for the variable DaysBetween. Because its values are integers, a geometric distribution is used by default to make the comparison and to compute the probability limits for the rare events chart. The value of parameter p for the geometric distribution is estimated from the data. InfectionNumber is an optional index variable whose values are used to label the rare event chart's horizontal axis.

```
ods graphics on;
proc rareevents data=Infections;
   compare DaysBetween;
   chart DaysBetween * InfectionNumber;
   label DaysBetween = 'Days between Infections';
run;
```

The ODS GRAPHICS ON statement enables ODS Graphics, which is necessary for the procedure to produce graphical output. The COMPARE statement produces the needle plot that is shown in Figure 17.4.

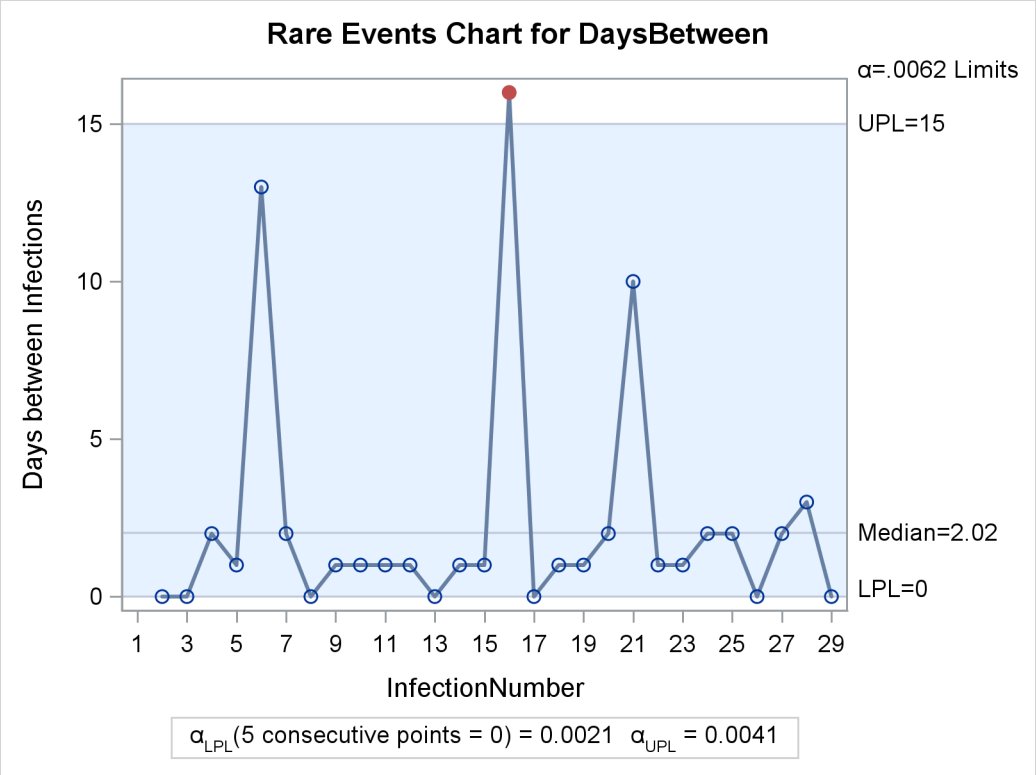
Figure 17.4 Distribution of Days between Infections



Interpreting a comparison plot of a small data sample can be difficult, but the data have the same general shape as the geometric distribution. The graph does not indicate that the geometric distribution is *not* appropriate for these data.

Figure 17.5 shows the rare events chart of the DaysBetween data that the CHART statement produces.

Figure 17.5 Rare Events Chart for Urinary Tract Infections



The number of days between infections 15 and 16 exceeds the UPL, signaling unusual variation. Here the unusual variation is welcome, because less frequent infections are desirable.

The median and probability limits for the chart are computed as described in the section "Constructing Rare Events Charts" on page 1183. The chart legend displays the probability, α_{UPL}, that a value from the geometric distribution is greater than the UPL. Note that the LPL in Figure 17.5 is equal to 0, which means that the probability of a DaysBetween value less than the LPL is 0. It is not unusual for the LPL to be equal to the minimum possible data value in a chart of integer data. When this is the case, the procedure checks for sequences of consecutive values equal to the LPL as an indication of unusual variation. The probability, α_{LPL}, of five consecutive 0 values from the geometric distribution is 0.0021, as indicated in the legend. The label outside the upper right corner of the chart shows the overall $\alpha = \alpha_{\mathrm{LPL}} + \alpha_{\mathrm{UPL}}$.

Syntax: RAREEVENTS Procedure

PROC RAREEVENTS < *options* > ;
  BY *variables* ;
  ID *variables* ;
  CHART < / *options* > ;
  COMPARE < / *options* > ;

The following sections describe the PROC RAREEVENTS statement and then describe the other statements in alphabetical order.

PROC RAREEVENTS Statement

PROC RAREEVENTS < *options* > **;**

The PROC RAREEVENTS statement invokes the RAREEVENTS procedure and specifies the input data sets. You can specify the following *options*:

DATA=*SAS-data-set*
specifies an input SAS data set that contains process data, which are measurements of times between events. You cannot specify the TABLE= option together with the DATA= option. For more information about DATA= data sets, see the section "DATA= Data Set" on page 1187.

LIMITS=*SAS-data-set*
specifies an input SAS data set that contains probability limits for a rare events chart.

TABLE=*SAS-data-set*
specifies an input SAS data set that contains summary information from a rare events chart. You can produce a TABLE= data set by specifying the OUTTABLE= option in a CHART statement. You can use a TABLE= input data set to display a previously computed rare events chart. You cannot specify the DATA= option together with the TABLE= option. For more information, see the section "TABLE= Data Set" on page 1189.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC RAREEVENTS to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the RAREEVENTS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The values of the ID *variables* are displayed in tooltips associated with points on a rare events chart when you create HTML output and specify the IMAGEMAP option in the ODS GRAPHICS statement. For more information, see Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

CHART Statement

CHART *process-variable* < ∗ *index-variable* > < / *options* > **;**

The CHART statement produces a rare events chart. The *process-variable* contains measurements of times between events. You can use the optional *index-variable* to label the tick marks on the chart's horizontal axis. If you specify a numeric *index-variable*, its values must be in ascending order in the input data: the procedure does not sort its input. When you do not specify an index variable, the tick marks on the horizontal axis are numbered sequentially, starting with 1.

Table 17.1 summarizes the *options* available in the CHART statement.

Table 17.1 CHART Statement Options

Option	Description
ALPHALPL=	Specifies the probability that is used to compute the lower probability limit
ALPHAUPL=	Specifies the probability that is used to compute the upper probability limit
DIST=	Specifies the distribution that is used to compute probability limits
EXCHART	Specifies that a chart be displayed only if there are points outside the probability limits
NOCHART	Suppresses creation of the rare events chart
NOHLABEL	Suppresses the horizontal axis label of the chart
NOVLABEL	Suppresses the vertical axis label of the chart
NPANELPOS=	Specifies the number of horizontal axis plotting positions per panel
ODSFOOTNOTE=	Adds a footnote to the chart
ODSFOOTNOTE2=	Adds a secondary footnote to the chart
ODSTITLE=	Specifies a title for the chart
ODSTITLE2=	Specifies a secondary title for the chart
OUTLIMITS=	Creates a SAS data set that contains probability limits for the chart
OUTTABLE=	Creates a SAS data set that contains a summary of the rare events chart
TOTPANELS=	Specifies the number of panels that are used to display the chart

You can specify the following options only in the CHART statement. For detailed descriptions of options common to the CHART and COMPARE statements, see the section "Common CHART and COMPARE Statement Options" on page 1180.

ALPHALPL=α_{LPL}

specifies the probability ($0 < \alpha_{\mathrm{LPL}} < 1$) that is used to compute the lower probability limit (LPL) for the rare events chart, based on the probability distribution that you specify in the DIST= option. The LPL is computed so that the probability of a measurement from the distribution being less than the LPL is α_{LPL}. By default, $\alpha_{\mathrm{LPL}} = 0.005$.

With a discrete probability distribution, it is not possible in general to compute a LPL for which this probability is exactly α_{LPL}. In that case, the chart includes a legend that shows the α_{LPL} that corresponds to the computed LPL.

ALPHAUPL=α_{UPL}

specifies the probability ($0 < \alpha_{\mathrm{UPL}} < 1$) that is used to compute the upper probability limit (UPL) for the rare events chart, based on the probability distribution that you specify in the DIST= option. The UPL is computed so that the probability of a measurement from the distribution being greater than the UPL is α_{UPL}. By default, $\alpha_{\mathrm{UPL}} = 0.005$.

With a discrete probability distribution, it is not possible in general to compute a UPL for which this probability is exactly α_{UPL}. In that case, the chart includes a legend that shows the α_{UPL} that corresponds to the computed UPL.

EXCHART< (LOWER | UPPER) >

specifies that a rare events chart be displayed only when one or more measurements lie outside the probability limits. If you specify EXCHART(LOWER), then a chart is displayed only when a measurement is less than the lower probability limit. If you specify EXCHART(UPPER), then a chart is displayed only when a measurement is greater than the upper probability limit.

NOCHART

suppresses display of the rare events chart. You can use the NOCHART option together with the OUTLIMITS= or OUTTABLE= option to create output data sets without displaying a chart.

NPANELPOS=*n*

NPANEL=*n*

specifies the number of horizontal axis plotting positions per panel in the chart. You usually specify this option to display more points in a panel than the default number, which is 50.

You can specify a positive or negative value for *n*. The absolute value of *n* must be at least 5. If *n* is positive, the number of positions is adjusted so that it is approximately equal to *n* and so that all panels display approximately the same number of positions. If *n* is negative, then no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions.

OUTLIMITS=*SAS-data-set*

creates an output SAS data set that contains the probability limits and related information for the rare events chart. For more information about the OUTLIMITS= data set, see the section "OUTLIMITS= Data Set" on page 1189.

OUTTABLE=*SAS-data-set*

creates an output SAS data set that contains the information plotted in the rare events chart, including the process measurements and the probability limits. For more information about the OUTTABLE= data set, see the section "OUTTABLE= Data Set" on page 1190.

TOTPANELS=*n*

specifies the number of panels that are used to display the chart. By default, the number of panels is determined by the value that you specify in the NPANELPOS= option. If you specify both the TOTPANELS= and NPANELPOS= options, the TOTPANELS= value takes precedence.

COMPARE Statement

COMPARE *process-variable* < */ options* > **;**

The COMPARE statement produces a graph that compares the process data to a reference probability distribution. By default, integer data are displayed in a needle plot. Continuous data are displayed in a histogram. When the reference distribution is an exponential or Weibull distribution, the COMPARE statement also produces a table of goodness-of-fit statistics.

Table 17.2 COMPARE Statement Options

Option	Description
DIST=	Specifies the reference distribution that is compared to the sample
NBINS=	Specifies the number of bins that are used to display the data distribution
NOHLABEL	Suppresses the horizontal axis label of a comparison chart
NOVLABEL	Suppresses the vertical axis label of a comparison chart
ODSFOOTNOTE=	Adds a footnote to a comparison chart
ODSFOOTNOTE2=	Adds a secondary footnote to a comparison chart
ODSTITLE=	Specifies a title for a comparison chart
ODSTITLE2=	Specifies a secondary title for a comparison chart
PROCESS=	Specifies how integer process data are displayed in a comparison chart
REFERENCE=	Specifies how an integer reference probability distribution is displayed in a comparison chart

You can specify the following options only in a COMPARE statement. For detailed descriptions of options common to the CHART and COMPARE statements, see the section "Common CHART and COMPARE Statement Options" on page 1180.

NBINS=*n*

specifies the number of bins that are used to display the process data in a comparison plot. For integer data, the default number of bins in the comparison plot is

$$\min(\max(n_{\max} - a + 1, 15), 50)$$

where $n_{\max}$ is the maximum data value and a is the minimum possible data value. For continuous data, the default number of histogram bins is based on the data range and the number of observations, using the method of Terrell and Scott (1985).

PROCESS=BAR | MARKER | NEEDLE
specifies how integer process data are displayed in a comparison chart. You can specify the following keywords:

BAR displays the process data by using bars.

MARKER plots the process data by using markers.

NEEDLE displays the process data by using needles.

By default, PROCESS=NEEDLE. The PROCESS= option has no effect on a comparison chart when a continuous reference distribution is in effect.

REFERENCE=BAR | MARKER | NEEDLE
specifies how an integer reference data distribution is displayed in a comparison chart. You can specify the following values:

BAR displays the reference data distribution by using bars.

MARKER displays the reference data distribution by using markers.

NEEDLE displays the reference data distribution by using needles.

By default, REFERENCE=NEEDLE. The REFERENCE= option has no effect on a comparison chart when a continuous reference distribution is in effect.

Common CHART and COMPARE Statement Options

You can specify the following *options* after a slash (/) in a CHART or COMPARE statement.

DIST=*distribution*
specifies the probability distribution that is compared to the input data by a COMPARE statement and that is used to compute probability limits for a rare events chart that you create by using a CHART statement. You can specify the following distributions:

EXPONENTIAL< (*exponential-options***) >**
requests an exponential distribution. You can specify the following *exponential-options*:

SIGMA=σ
specifies the scale parameter for the exponential distribution. By default, σ is estimated from the process data.

THETA=θ **| EST**
specifies the threshold parameter for the exponential distribution. By default, $\theta = 0$. The specified value must be greater than or equal to 0. You can specify THETA=EST to compute an estimate of θ from the process data. If any data value is less than θ, the procedure issues a warning and sets θ to the minimum data value.

GEOMETRIC< (*geometric-options*) >
requests a geometric distribution. You can specify the following *geometric-options*:

P=*p* | MLE | MVUE
specifies the probability of success in a single Bernoulli trial on which the geometric distribution is based. This is the probability that an opportunity for a rare event to occur will actually result in an occurrence. You can specify P=MLE to compute a maximum likelihood estimate (MLE) of *p* or P=MVUE to compute a minimum variance unbiased estimate (MVUE) of *p*. By default, an MVUE is computed if the SHIFT= parameter value is 0 or 1, and an MLE is computed otherwise.

SHIFT=*a*
specifies the minimum possible value ($a \geq 0$) for the geometric distribution. By default, $a = 0$. If a measurement from the input data represents the time *until* an event occurs (including the event itself) instead of times *between* events, then you should specify $a = 1$. If any data value is less than a, the procedure issues a warning and sets a to the minimum data value.

WEIBULL< (*weibull-options*) >
requests a Weibull distribution. You can specify the following *weibull-options*:

C=*c*
specifies the shape parameter for the Weibull distribution. By default, *c* is estimated from the process data.

SIGMA=σ
specifies the scale parameter for the exponential distribution. By default, σ is estimated from the process data.

THETA=θ | EST
specifies the threshold parameter for the Weibull distribution. By default, $\theta = 0$. The specified value must be greater than or equal to 0. You can specify THETA=EST to compute an estimate of θ from the process data. If any data value is less than θ, the procedure issues a warning and sets θ to the minimum data value.

The procedure determines whether the process data have continuous or integer values. By default, an exponential distribution is used for continuous data and a geometric distribution is used for integer data.

NOHLABEL
suppresses the horizontal axis label in the graph.

NOVLABEL
suppresses the vertical axis label in the graph.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string*'
adds a footnote to the graph. If you specify the FOOTNOTE (or FOOTNOTE1) keyword, the value of the SAS FOOTNOTE statement is used as the graph footnote. If you specify a quoted string, that string is used as the footnote. The quoted string can contain the following escape characters, which are replaced by the values indicated:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the process variable has no label).

ODSFOOTNOTE2=FOOTNOTE2 | '*string***'**

adds a secondary footnote to the graph. If you specify the FOOTNOTE2 keyword, the value of the SAS FOOTNOTE2 statement is used as the secondary graph footnote. If you specify a quoted string, that string is used as the secondary footnote. The quoted string can contain the following escape characters, which are replaced by the values indicated:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the process variable has no label).

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | '*string***'**

specifies a title for the graph. You can specify the following values:

TITLE (or **TITLE1**) uses the value of the SAS TITLE statement as the graph title.

NONE suppresses all graph titles.

DEFAULT uses the default title.

If you specify a quoted string, that string is used as the graph title. The quoted string can contain the following escape characters, which are replaced by the values indicated:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the analysis variable has no label).

ODSTITLE2=TITLE2 | '*string***'**

specifies a secondary title for the graph. If you specify the TITLE2 keyword, the value of the SAS TITLE2 statement is used as the secondary graph title. If you specify a quoted string, that string is used as the secondary title. The quoted string can contain the following escape characters, which are replaced by the values indicated:

\n is replaced by the process variable name.

\l is replaced by the process variable label (or name if the analysis variable has no label).

Details: RAREEVENTS Procedure

Constructing Rare Events Charts

Each point on the rare events chart indicates the value of an individual measurement from the input data set. You compute the lower probability limit (LPL), median, and upper probability limit (UPL) by solving for their values in the following equations, which use the cumulative distribution function (cdf) of the probability distribution that you specify in the DIST= option:

- $\mathrm{cdf}(\mathrm{LPL}) = \alpha_{\mathrm{LPL}}$
- $\mathrm{cdf}(\mathrm{median}) = 0.5$
- $\mathrm{cdf}(\mathrm{UPL}) = 1 - \alpha_{\mathrm{UPL}}$

Probability Limits Based on an Exponential Distribution

The cumulative distribution function of an exponential distribution with scale parameter σ and threshold parameter θ is

$$\mathrm{cdf}(x) = 1 - \exp\left(-\frac{(x-\theta)}{\sigma}\right)$$

Solving the equations listed previously, the median and probability limits values are as follows:

- $\mathrm{LPL} = \theta - \sigma \ln(1 - \alpha_{\mathrm{LPL}})$
- $\mathrm{median} = \theta + \sigma \ln(2)$
- $\mathrm{UPL} = \theta - \sigma \ln(\alpha_{\mathrm{UPL}})$

Probability Limits Based on a Geometric Distribution

The cumulative distribution function of a geometric distribution with shift parameter a and probability p is

$$\mathrm{cdf}(x) = 1 - (1-p)^{x-a+1}$$

Because the geometric distribution is used with integer data, meaningful probability limits must have integer values. Therefore the solutions to the equations listed previously are:

- $\mathrm{LPL} = \left\lfloor \frac{\ln(1-\alpha_{\mathrm{LPL}})}{\ln(1-p)} + a \right\rfloor$
- $\mathrm{median} = \frac{\ln(0.5)}{\ln(1-p)} + a$
- $\mathrm{UPL} = \left\lceil \frac{\ln(\alpha_{\mathrm{UPL}})}{\ln(1-p)} + a - 1 \right\rceil$

The probability of a value from the distribution being greater than the UPL is as close as possible to α_{UPL} without exceeding it, and the probability of a value from the distribution being less than the LPL is as close as possible to α_{LPL} without exceeding it. The α_{UPL} and α_{LPL} values that correspond to the computed limits are displayed in a legend on the rare events chart.

With integer probability limits, it is not unusual for the computed LPL to be equal to the minimum possible data value, so that no data value can be less than the LPL. In that case, the following value is computed:

$$m = \left\lceil \frac{\ln(\alpha_{\mathrm{LPL}})}{\ln(p)} \right\rceil$$

The probability of a sequence of m consecutive values from the geometric distribution each being equal to the LPL is as close to α_{LPL} as possible without exceeding it. The RAREEVENTS procedure flags any sequence of m consecutive measurements equal to the LPL as a sign of unusual variation.

Probability Limits Based on a Weibull Distribution

The cumulative distribution function of a Weibull distribution with scale parameter σ, shape parameter c, and threshold parameter θ is

$$\mathrm{cdf}(x) = 1 - \exp\left(-\left(\frac{(x-\theta)}{\sigma}\right)^{c}\right)$$

This produces the following probability limits:

- $\mathrm{LPL} = \theta + \sigma\left(-\ln(1-\alpha_{\mathrm{LPL}})\right)^{1/c}$
- $\mathrm{median} = \theta + \sigma\left(\ln(2)\right)^{1/c}$
- $\mathrm{UPL} = \theta + \sigma\left(-\ln(\alpha_{\mathrm{UPL}})\right)^{1/c}$

EDF Goodness-of-Fit Tests

When a continuous reference distribution is in effect, the COMPARE statement provides a series of goodness-of-fit tests based on the empirical distribution function (EDF). For a thorough discussion, see D'Agostino and Stephens (1986).

The empirical distribution function is defined for a set of n independent observations $X_1, \ldots, X_n$ with a common distribution function $F(x)$. Denote the observations ordered from smallest to largest as $X_{(1)}, \ldots, X_{(n)}$. The empirical distribution function, $F_n(x)$, is defined as

$$\begin{aligned} F_n(x) &= 0, \quad x < X_{(1)} \\ F_n(x) &= \tfrac{i}{n}, \quad X_{(i)} \le x < X_{(i+1)} \quad i = 1, \ldots, n-1 \\ F_n(x) &= 1, \quad X_{(n)} \le x \end{aligned}$$

Note that $F_n(x)$ is a step function that takes a step of height $\frac{1}{n}$ at each observation. This function estimates the distribution function $F(x)$. At any value x, $F_n(x)$ is the proportion of observations less than or equal to x,

while $F(x)$ is the probability of an observation less than or equal to x. EDF statistics measure the discrepancy between $F_n(x)$ and $F(x)$.

The computational formulas for the EDF statistics make use of the probability integral transformation $U = F(X)$. If $F(X)$ is the distribution function of X, the random variable U is uniformly distributed between 0 and 1.

Given n observations $X_{(1)}, \ldots, X_{(n)}$, the values $U_{(i)} = F(X_{(i)})$ are computed by applying the transformation, as shown in the following sections.

The COMPARE statement provides three EDF tests:

- Kolmogorov-Smirnov
- Anderson-Darling
- Cramér-von Mises

These tests are based on various measures of the discrepancy between the empirical distribution function $F_n(x)$ and the reference parametric cumulative distribution function $F(x)$.

The following sections provide formal definitions of the EDF statistics.

Kolmogorov-Smirnov Statistic

The Kolmogorov-Smirnov statistic (D) is defined as

$$D = \sup_x |F_n(x) - F(x)|$$

The Kolmogorov-Smirnov statistic belongs to the supremum class of EDF statistics. This class of statistics is based on the largest vertical difference between $F(x)$ and $F_n(x)$.

The Kolmogorov-Smirnov statistic is computed as the maximum of D^+ and D^-, where D^+ is the largest vertical distance between the EDF and the distribution function when the EDF is greater than the distribution function, and D^- is the largest vertical distance when the EDF is less than the distribution function.

$$\begin{aligned} D^+ &= \max_i \left(\frac{i}{n} - U_{(i)} \right) \\ D^- &= \max_i \left(U_{(i)} - \frac{i-1}{n} \right) \\ D &= \max \left(D^+, D^- \right) \end{aligned}$$

Anderson-Darling Statistic

The Anderson-Darling statistic and the Cramér-von Mises statistic belong to the quadratic class of EDF statistics. This class of statistics is based on the squared difference $(F_n(x) - F(x))^2$. Quadratic statistics have the following general form:

$$Q = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, \psi(x) dF(x)$$

The function $\psi(x)$ weights the squared difference $(F_n(x) - F(x))^2$.

The Anderson-Darling statistic (A^2) is defined as

$$A^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \left[F(x)\left(1 - F(x)\right)\right]^{-1} dF(x)$$

Here the weight function is $\psi(x) = \left[F(x)\left(1 - F(x)\right)\right]^{-1}$.

The Anderson-Darling statistic is computed as

$$A^2 = -n - \frac{1}{n}\sum_{i=1}^{n}\left[(2i-1)\log U_{(i)} + (2n+1-2i)\log\left(\{1 - U_{(i)}\}\right)\right]$$

Cramér-von Mises Statistic

The Cramér-von Mises statistic (W^2) is defined as

$$W^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \, dF(x)$$

Here the weight function is $\psi(x) = 1$.

The Cramér-von Mises statistic is computed as

$$W^2 = \sum_{i=1}^{n}\left(U_{(i)} - \frac{2i-1}{2n}\right)^2 + \frac{1}{12n}$$

Probability Values for EDF Tests

For the probability values (p-values) associated with the EDF test statistics, the RAREEVENTS procedure uses internal tables of probability levels similar to those given by D'Agostino and Stephens (1986). If the value is between two probability levels, then linear interpolation is used to estimate the probability value. The probability value depends upon the parameters that are known and the parameters that are estimated for the distribution you are fitting. Table 17.3 summarizes the combinations of estimated parameters for which EDF tests are available.

Table 17.3 Availability of EDF Tests

Distribution	Parameters			Tests Available
	Threshold	Scale	Shape	
Exponential	θ known,	σ known		all
	θ known	σ unknown		all
	θ unknown	σ known		all
	θ unknown	σ unknown		all
Weibull	θ known	σ known	c known	all
	θ known	σ unknown	c known	A^2 and W^2
	θ known	σ known	c unknown	A^2 and W^2
	θ known	σ unknown	c unknown	A^2 and W^2

Table 17.3 (*continued*)

Distribution	Parameters			Tests Available
	Threshold	**Scale**	**Shape**	
	θ unknown	σ known	$c > 2$ known	all
	θ unknown	σ unknown	$c > 2$ known	all
	θ unknown	σ known	$c > 2$ unknown	all
	θ unknown	σ unknown	$c > 2$ unknown	all

Input Data Sets

The RAREEVENTS procedure accepts a single primary input data set of either of two types:

- A DATA= data set contains process measurements to be analyzed.
- A TABLE= data set contains a summary of a rare events chart, which consists of the measurements, probability limits, and other information.

These options are mutually exclusive. If you do not specify an option that identifies a primary input data set, PROC RAREEVENTS uses the most recently created SAS data set as a DATA= data set. Valid process measurements are greater than or equal to zero. Missing and negative values are ignored.

You can also specify a LIMITS= data set that contains probability limits for a rare events chart.

DATA= Data Set

A DATA= data set must include a process variable that contains measurements of the times between rare events. These measurements can be integers (for example, a count of days between events) or continuous values. In addition to the process variable, a DATA= data set can include the following:

- BY variables
- ID variables
- index variable

The values of the optional index variable are used to label the horizontal axis tick marks on a rare events chart that is produced by a CHART statement. The index variable has no application in a COMPARE statement.

LIMITS= Data Set

A LIMITS= data set contains probability limit information for a rare events chart. Usually, you create a LIMITS= data set by specifying the OUTLIMITS= option in a CHART statement. You can use a LIMITS= data set to specify historical probability limits for a process or custom probability limits that are computed by other means.

Table 17.4 lists the variables that a LIMITS= data set can contain.

Table 17.4 LIMITS= Data Set Variables

Variable	Description
ALPHALPL	Probability associated with the lower probability limit
ALPHAUPL	Probability associated with the upper probability limit
_C_	Shape parameter for a Weibull distribution
DIST	Name of the distribution used to compute the probability limits
INDEX	Name of the optional index variable
LPL	Lower probability limit
MEDIAN	Median of the probability distribution
_P_	Probability of success in a single Bernoulli trial on which a geometric distribution is based
PARMEST	Specifies whether distribution parameters are estimated or specified
SHIFT	Minimum possible value for a geometric distribution
SIGMA	Scale parameter for an exponential or Weibull distribution
THETA	Threshold parameter for an exponential or Weibull distribution
UPL	Upper probability limit
VAR	Name of the process variable that contains measurements of times between events

A LIMITS= data set must contain the variables corresponding to the parameters of the distribution indicated by the value of the _DIST_ variable:

EXPONENTIAL _THETA_, _SIGMA_

GEOMETRIC _P_

WEIBULL _THETA_, _SIGMA_, _C_

The variable _PARMEST_ contains a code indicating whether the probability distribution parameters are specified or estimated. The _PARMEST_ code is the sum of codes for each parameter. If a parameter is specified, its code is zero. If a parameter is estimated, its code is as show in Table 17.5.

Table 17.5 LIMITS= Data Set Variables

Distribution	Parameter	Estimated Code
Exponential	θ	1
	σ	2
Geometric	p	1
Weibull	θ	1
	σ	2
	c	4

For example, the _PARMEST_ value for a Weibull distribution with θ estimated, σ specified, and c estimated is $1 + 0 + 4 = 5$.

TABLE= Data Set

A TABLE= data set contains a summary of a rare events chart. Usually, you create a TABLE= data set by specifying the OUTTABLE= option in a CHART statement. You can use a TABLE= data set to display a previously created rare events chart or to specify custom probability limits by computing your own _LPL_ and _UPL_ values.

Table 17.6 lists the variables that a TABLE= data set contains.

Table 17.6 TABLE= Data Set Variables

Variable	Description
ALPHALPL	Probability associated with the lower probability limit
ALPHAUPL	Probability associated with the upper probability limit
DIST	Name of the distribution used to compute the probability limits
EXLIM	Flag that indicates that a probability limit was exceeded
index	Optional index variable
LPL	Lower probability limit
MEDIAN	Median of the probability distribution
process	Process variable containing measurements of times between events
UPL	Upper probability limit

Output Data Sets

OUTLIMITS= Data Set

You can save probability limits and related information in an output data set by specifying the OUTLIMITS= option in a CHART statement. Table 17.7 lists the variables that an OUTLIMITS= data set contains.

Table 17.7 OUTLIMITS= Data Set Variables

Variable	Description
ALPHALPL	Probability associated with the lower probability limit
ALPHAUPL	Probability associated with the upper probability limit
_C_	Shape parameter for a Weibull distribution
DIST	Name of the distribution used to compute the probability limits
INDEX	Name of the optional index variable
LPL	Lower probability limit
MEDIAN	Median of the probability distribution
_P_	Probability of success in a single Bernoulli trial on which a geometric distribution is based
PARMEST	Specifies whether distribution parameters are estimated or specified
SHIFT	Minimum possible value for a geometric distribution
SIGMA	Scale parameter for an exponential or Weibull distribution
THETA	Threshold parameter for an exponential or Weibull distribution
UPL	Upper probability limit

Table 17.7 *(continued)*

Variable	Description
VAR	Name of the process variable that contains measurements of times between events

When the probability limits are based on an exponential distribution, the OUTLIMITS= data set contains the variables _SIGMA_ and _THETA_. When the probability limits are based on a geometric distribution, the OUTLIMITS= data set contains the variables _P_ and _SHIFT_. When the probability limits are based on a Weibull distribution, the OUTLIMITS= data set contains the variables _C_, _SIGMA_ and _THETA_.

OUTTABLE= Data Set

You can save process measurements, probability limits, and related information in an output data set by specifying the OUTTABLE= option in a CHART statement. Table 17.8 lists the variables that an OUTTABLE= data set contains.

Table 17.8 OUTTABLE= Data Set Variables

Variable	Description
ALPHALPL	Probability associated with the lower probability limit
ALPHAUPL	Probability associated with the upper probability limit
DIST	Name of the distribution used to compute the probability limits
EXLIM	Flag that indicates that a probability limit was exceeded
index	Optional index variable
LPL	Lower probability limit
MEDIAN	Median of the probability distribution
process	Process variable containing measurements of times between events
UPL	Upper probability limit

ODS Table Names

PROC RAREEVENTS assigns a name to each table that it creates. You can use these names to refer to the tables when you use the Output Delivery System (ODS) to select tables and create output data sets. The ODS table names are listed in Table 17.9.

Table 17.9 ODS Tables Produced by PROC RAREEVENTS

ODS Table Name	Description	Statement	Option
GoodnessOfFit	Goodness-of-fit tests for fitted distribution	COMPARE	DIST=EXPONENTIAL (default for continuous data) DIST=WEIBULL

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The RAREEVENTS procedure assigns a name to each graph that it creates using ODS Graphics. You can use these names to refer to the graphs when you use ODS. The graph names are listed in Table 17.10.

Table 17.10 ODS Graphics Produced by PROC RAREEVENTS

ODS Graph Name	Plot Description	Statement or Option
RareEventsChart	Rare events chart of process data	CHART statement
ComparisonPlot	Comparison plot	COMPARE statement

Examples: RAREEVENTS Procedure

Example 17.1: Monitoring Urinary Tract Infections

The data for this example are from Santiago and Smith (2013).

A hospital system tracked the frequency of urinary tract infections (UTIs) acquired by patients while in one of its hospitals. The following statements create a SAS data set with the variable DaysBetween, which contains the number of days between discharges from the hospital of male patients who acquired UTIs while there:

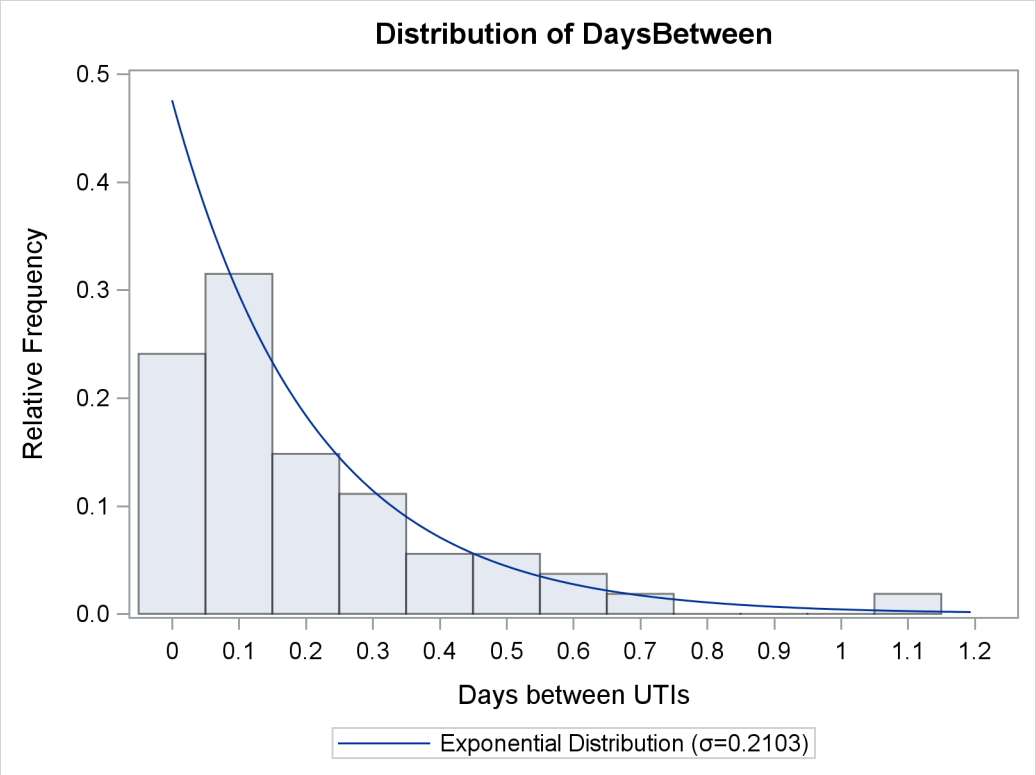
```
data UrinaryTractInfections;
   input DaysBetween @@;
   label DaysBetween = 'Days between UTIs';
datalines;
0.57014 0.07431 0.15278 0.14583 0.13889
0.14931 0.03333 0.08681 0.33681 0.03819
0.24653 0.29514 0.11944 0.05208 0.12500
0.25000 0.40069 0.02500 0.12014 0.11458
0.00347 0.12014 0.04861 0.02778 0.32639
0.64931 0.14931 0.01389 0.03819 0.46806
0.22222 0.29514 0.53472 0.15139 0.52569
0.07986 0.27083 0.04514 0.13542 0.08681
0.40347 0.12639 0.18403 0.70833 0.15625
0.24653 0.04514 0.01736 1.08889 0.05208
0.02778 0.03472 0.23611 0.35972
;
```

The following statements produce a graph that compares the data to a reference distribution whose parameters are estimated from the data. The RAREEVENTS procedure uses an exponential distribution by default because the data are continuous.

```
proc rareevents data=UrinaryTractInfections;
   compare DaysBetween / nbins=12;
run;
```

The NBINS= option specifies that 12 histogram bins be used to display the data. Output 17.1.1 shows the resulting histogram of the data overlaid with the exponential curve.

Output 17.1.1 Distribution of Intervals between UTIs



Because a continuous distribution is in effect, the COMPARE statement also produces a table of goodness-of-fit statistics, which is shown in Output 17.1.2.

Output 17.1.2 Goodness-of-Fit Statistics for UTIs

The RAREEVENTS Procedure

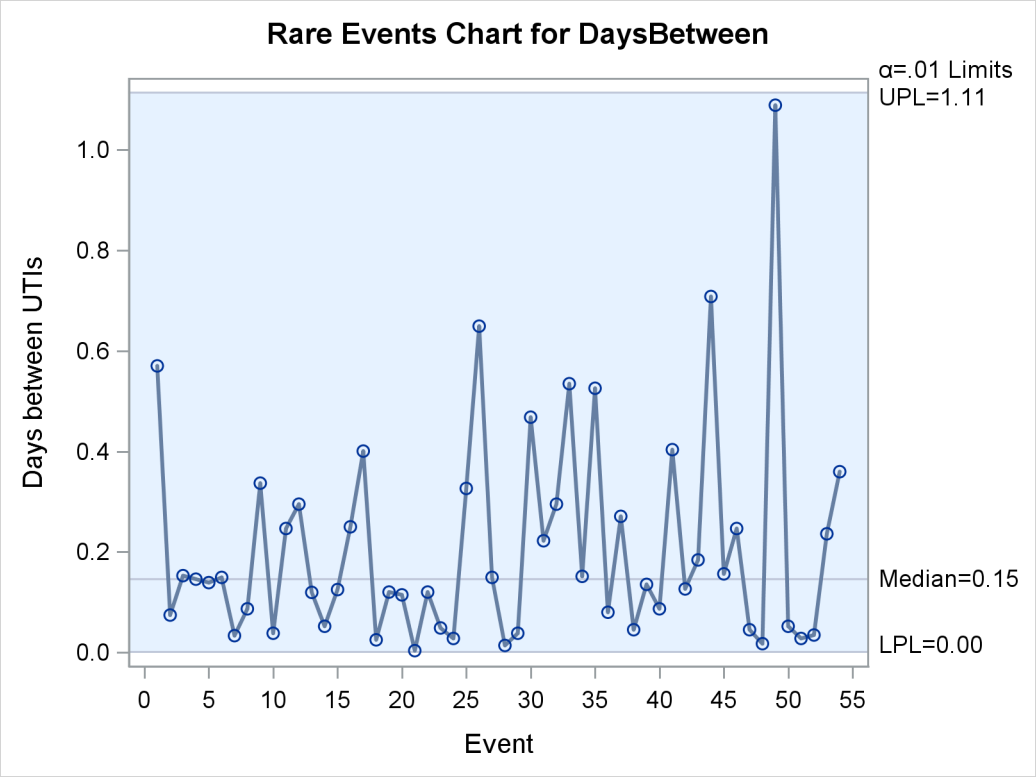
Goodness-of-Fit Tests for Exponential Distribution				
Test		**Statistic**	**p Value**	
Kolmogorov-Smirnov	**D**	0.08673920	**Pr > D**	>0.500
Cramer-von Mises	**W-Sq**	0.04104603	**Pr > W-Sq**	>0.500
Anderson-Darling	**A-Sq**	0.26919944	**Pr > A-Sq**	>0.500

The histogram and the goodness-of-fit tests indicate that an exponential distribution is appropriate for the data. The following statements produce a rare events chart for the days between UTIs:

```
proc rareevents data=UrinaryTractInfections;
   chart DaysBetween / totpanels=1;
run;
```

The TOTPANELS= option specifies that all the observations be displayed in a single panel, or page. No index variable is specified, so the DaysBetween values are numbered consecutively, starting with 1. Output 17.1.3 shows the resulting chart.

Output 17.1.3 Rare Events Chart for Urinary Tract Infections



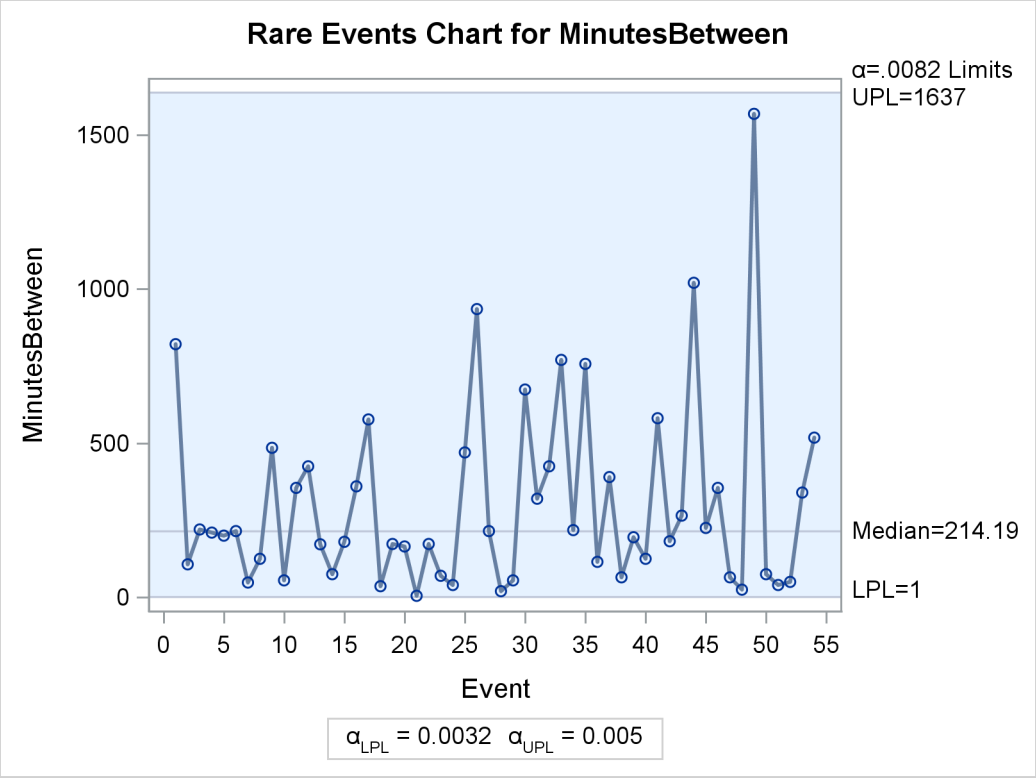
The rare events chart shows no indication of unusual variation in the incidence of UTIs among male patients.

Although Santiago and Smith (2013) provide the data as the (continuous) numbers of days between patient discharges, they could just as well have been recorded as the (integer) number of minutes between discharges. The following statements compute the variable MinutesBetween, which contains counts of the minutes between infections, and produce a rare events chart of the counts. Because the data are integer values, the probability limits are based on a geometric distribution.

```
data UrinaryTractInfections;
   set UrinaryTractInfections;
   MinutesBetween = round( DaysBetween * 1440, 1 );
run;

proc rareevents data=UrinaryTractInfections;
   chart MinutesBetween / totpanels=1;
run;
```

Output 17.1.4 shows the rare events chart for MinutesBetween. The median and probability limits for this chart are very close, but not exactly equal, to the corresponding values measured in days in Output 17.1.3.

Output 17.1.4 Rare Events Chart for Urinary Tract Infections



Example 17.2: Airline Crashes

The following statements create a SAS data set that contains data from the National Transportation Safety Board (NTSB) Aviation Accident Database. You can query the database at http://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx. These data involve commercial airline crashes that resulted in fatalities and took place in the United States from 1982 through 2013. The DATA step creates a new variable, DaysBetweenCrashes, that records the number of days between successive crashes.

```
data AirCrashes;
   input EventID : $14. EventDate mmddyy10. Location & $32.;
   DaysBetweenCrashes = EventDate - lag(EventDate);
   label DaysBetweenCrashes = 'Days';
datalines;
20020917X01907 01/13/1982 WASHINGTON, DC
20020917X01909 01/23/1982 BOSTON, MA
20020917X03104 07/09/1982 NEW ORLEANS, LA
20020917X04908 11/11/1982 MIAMI, FL
20001214X41967 01/09/1983 BRAINERD, MN
20001214X41968 01/11/1983 DETROIT, MI
20001214X44795 10/11/1983 PINCKNEYVILLE, IL
20001214X45258 12/20/1983 SIOUX FALLS, SD
```

```
20001214X39535 05/30/1984 CHALKHILL, PA
20001214X35492 01/09/1985 KANSAS CITY, KS
20001214X35493 01/21/1985 RENO, NV
20001214X36375 05/31/1985 NASHVILLE, TN
20001214X37434 08/02/1985 DALLAS/FT WORTH, TX
20001214X37757 09/06/1985 MILWAUKEE, WI
20001213X34942 10/04/1986 KELLY AFB, TX
20001213X35148 11/06/1986 TAMPA, FL
20001213X30626 04/13/1987 KANSAS CITY, MO
20001213X31759 08/16/1987 ROMULUS, MI
20001213X32505 11/15/1987 DENVER, CO
20001213X32679 12/07/1987 SAN LUIS OBISPO, CA
20001213X25439 04/28/1988 MAUI, HI
20001213X26528 08/31/1988 DALLAS/FT WORTH, TX
20001213X27734 02/09/1989 SALT LAKE CITY, UT
20001213X27705 02/24/1989 HONOLULU, HI
20001213X27867 03/15/1989 WEST LAFAYETTE, IN
20001213X27869 03/18/1989 SAGINAW, TX
20001213X28786 07/19/1989 SIOUX CITY, IA
20001213X29335 09/20/1989 FLUSHING, NY
20001213X29644 10/07/1989 ORLANDO, FL
20001213X29997 12/27/1989 MIAMI, FL
20001212X22400 01/18/1990 ATLANTA, GA
20001212X22386 01/31/1990 INDIANAPOLIS, IN
20001212X22742 03/13/1990 PHOENIX, AZ
20001212X24506 10/03/1990 CAPE CANAVERAL, FL
20001212X24751 12/03/1990 ROMULUS, MI
20001212X24751 12/03/1990 ROMULUS, MI
20001212X16433 02/01/1991 LOS ANGELES, CA
20001212X16434 02/17/1991 CLEVELAND, OH
20001212X16583 03/03/1991 COLORADO SPGS, CO
20001212X18366 10/12/1991 BRIDGEPORT, CT
20001211X14094 02/15/1992 SWANTON, OH
20001211X14270 03/22/1992 FLUSHING, NY
20001211X14503 04/08/1992 DAYTON, OH
20001211X16222 12/08/1992 FLUSHING, NY
20001211X12079 04/04/1993 CHICAGO, IL
20001206X01727 07/02/1994 CHARLOTTE, NC
20001206X02233 09/08/1994 ALIQUIPPA, PA
20001206X02420 10/31/1994 ROSELAWN, IN
20001206X02586 11/22/1994 BRIDGETON, MO
20001208X05743 05/11/1996 MIAMI, FL
20001208X06203 07/06/1996 PENSACOLA, FL
20001208X06204 07/17/1996 EAST MORICHES, NY
20001208X06132 07/20/1996 RUSSIAN MISSION, AK
20001208X07619 03/27/1997 JAMAICA, NY
20001208X08607 08/07/1997 MIAMI, FL
20001208X09291 12/28/1997 PACIFIC OCEAN
20001212X18961 06/01/1999 LITTLE ROCK, AR
20001212X19260 07/28/1999 LITTLE ROCK, AR
20001212X20339 01/31/2000 Port Hueneme, CA
20001212X20472 02/16/2000 RANCHO CORDOVA, CA
20001212X22314 11/20/2000 MIAMI, FL
20010904X01867 08/05/2001 Washington, DC
```

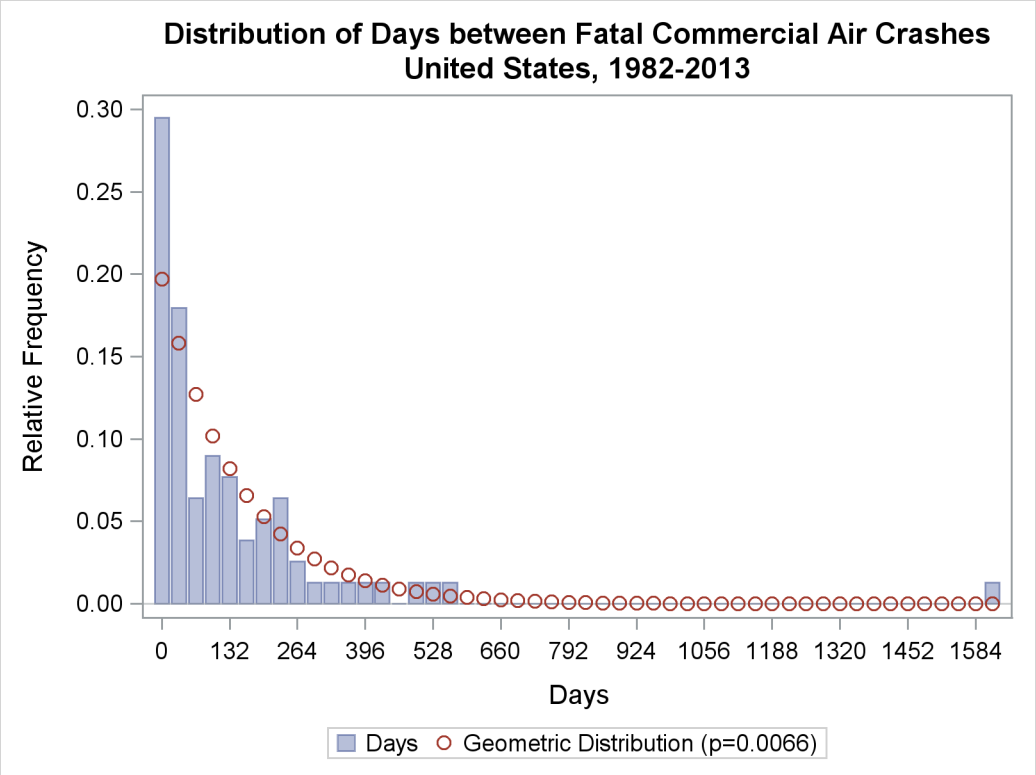
```
20020123X00106 09/11/2001 Shanksville, PA
20020123X00105 09/11/2001 Arlington, VA
20020123X00104 09/11/2001 New York City, NY
20020123X00103 09/11/2001 New York City, NY
20011130X02321 11/12/2001 Belle Harbor, NY
20030110X00049 01/08/2003 Charlotte, NC
20030917X01555 09/12/2003 Norfolk, VA
20040825X01286 08/13/2004 Florence, KY
20041020X01659 10/19/2004 Kirksville, MO
20050609X00744 06/07/2005 Washington, DC
20051213X01964 12/08/2005 Chicago, IL
20060106X00018 12/19/2005 Miami, FL
20060131X00140 01/16/2006 El Paso, TX
20060828X01244 08/27/2006 Lexington, KY
20070718X00958 07/10/2007 Tunica, MS
20090213X13613 02/12/2009 Clarence Center, NY
20130814X15751 08/14/2013 Birmingham, AL
;
```

The following statements produce a comparison plot and a rare events chart for DaysBetweenCrashes:

```
proc rareevents data=AirCrashes;
   id EventId EventDate Location;
   compare DaysBetweenCrashes /
      process=bar
      reference=marker
      odstitle='Distribution of Days between Fatal Commercial Air Crashes'
      odstitle2='United States, 1982-2013'
      ;
   chart DaysBetweenCrashes /
      odstitle='Days between Fatal Commercial Air Crashes'
      odstitle2='United States, 1982-2013'
      nohlabel
      ;
run;
```

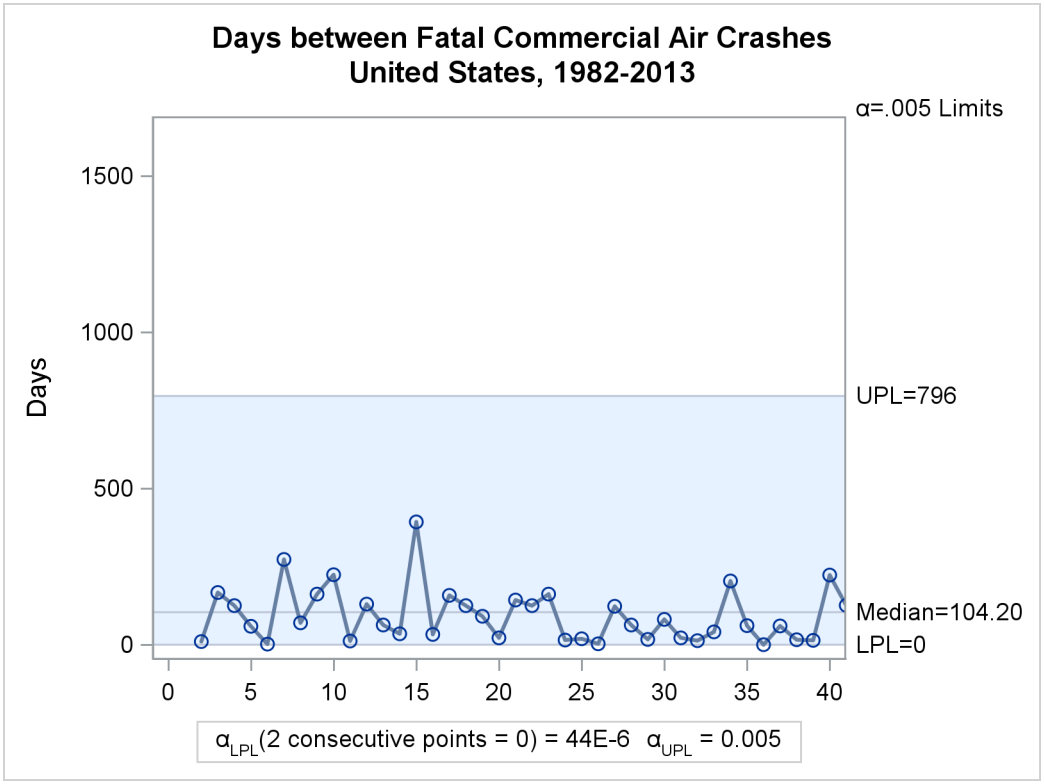
The PROCESS= and REFERENCE= options determine how the process data and reference distribution are displayed in the comparison chart. The ODSTITLE= and ODSTITLE2= options specify titles for the graphs. The NOHLABEL option suppresses the horizontal axis label in the rare events chart. Output 17.2.1 compares the data to a geometric distribution and indicates that the distribution reasonably describes the data.

Output 17.2.1 Comparison Plot for Days between Crashes

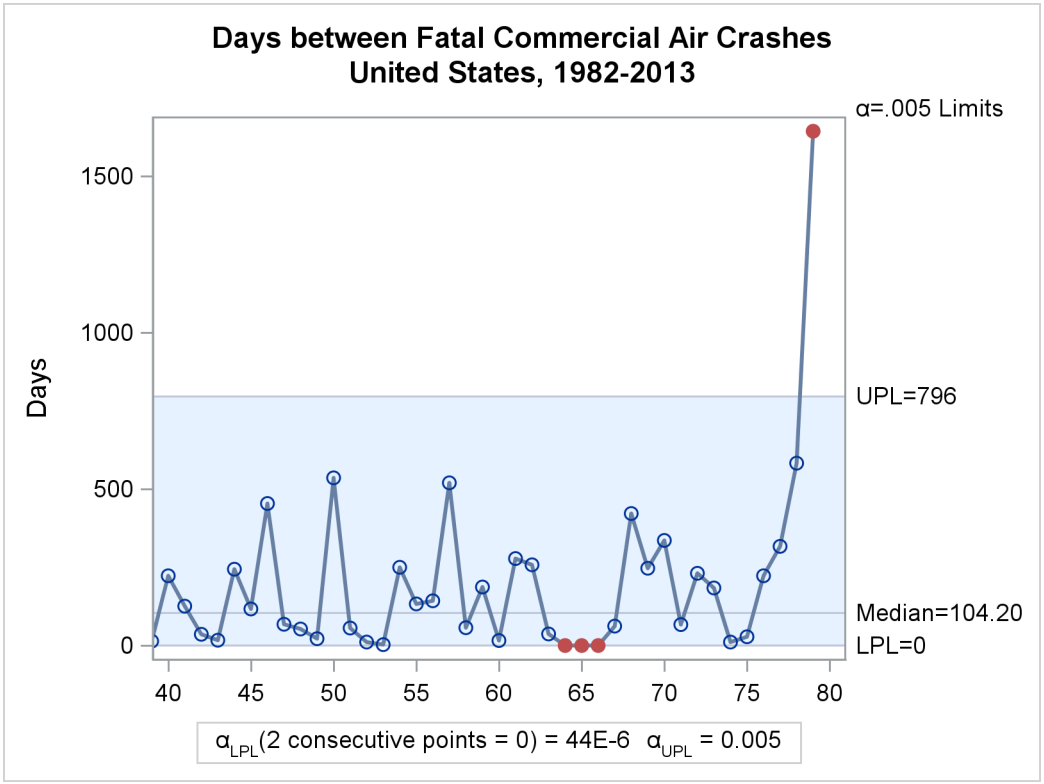


Output 17.2.2 and Output 17.2.3 show the two panels of the chart.

Output 17.2.2 Rare Events Chart for Air Crashes (Panel 1)



Output 17.2.3 Rare Events Chart for Air Crashes (Panel 2)



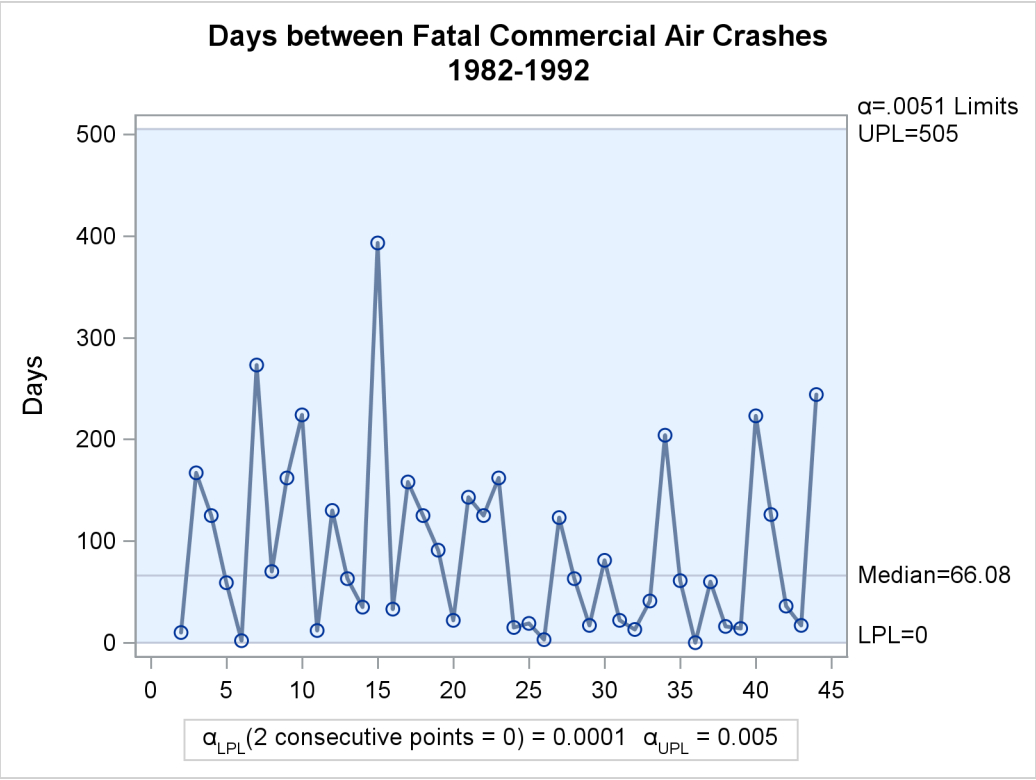
In Output 17.2.3, the three consecutive points equal to the LPL that signal unusual variation correspond to the terrorist attacks on September 11, 2001. Note that the counts of days between crashes are generally smaller in the first panel of the chart (Output 17.2.2) than in the second panel. Those measurements correspond approximately to the years from 1982 to 1992.

There appears to have been a significant change in the process around that time. The following statements create a variable called Years that divides the data into periods before and after the end of 1992. A rare events chart is produced for the first period.

```
data AirCrashes2;
   set AirCrashes;
   if EventDate <= '31dec1992'd then
      Years = '1982-1992';
   else
      Years = '1993-2013';
run;

proc rareevents data=AirCrashes2;
   id EventId EventDate Location;
   where Years = '1982-1992';
   chart DaysBetweenCrashes /
      odstitle='Days between Fatal Commercial Air Crashes'
      odstitle2='1982-1992'
      nohlabel
      outlimits=AirLimits ;
run;
```

The OUTLIMITS= option in the CHART statement saves the probability limits in the data set AirLimits. Output 17.2.4 shows the rare events chart for the years 1982–1992.

Output 17.2.4 Rare Events Chart for Air Crashes (1982–1992)



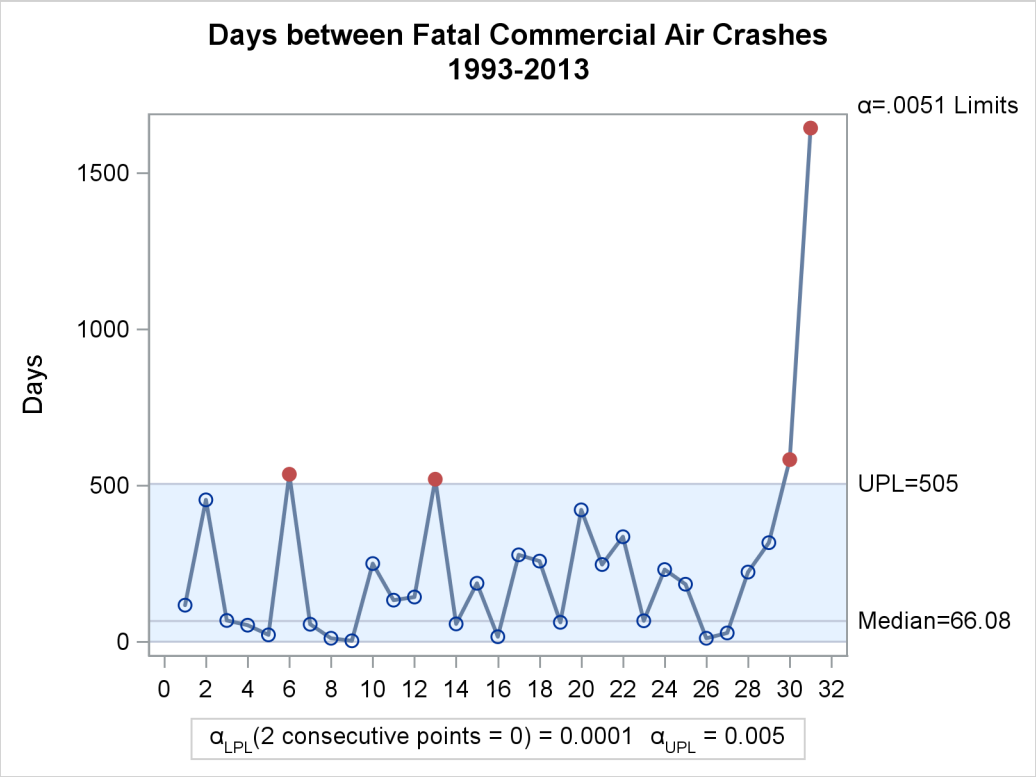
Note the difference in probability limits between Output 17.2.4 and Output 17.2.2, whose limits were computed using all the data. The new chart indicates a stable process from 1982 to 1992.

The following statements create a rare events chart for the years 1993 to 2013, with the data that correspond to the September 11 crashes removed from the analysis:

```
proc rareevents data=AirCrashes2 limits=AirLimits;
   id EventId EventDate Location;
   where Years = '1993-2013' and EventDate ne '11sep2001'd;
   chart DaysBetweenCrashes /
      odstitle='Days between Fatal Commercial Air Crashes'
      odstitle2='1993-2013'
      nohlabel;
run;
```

The LIMITS= option applies the historical probability limits computed for the earlier period to these data. Output 17.2.5 shows the rare events chart for the years 1993 to 2013.

Output 17.2.5 Rare Events Chart for Air Crashes (1993–2013)



You can see that 4 of 32 measurements on the chart exceed the UPL that was computed from data for the years 1982 to 1992. This is strong evidence of a change in the process, with fatal airline crashes becoming less frequent.

References

Benneyan, J. (2001a). "Number-Between *g*-Type Statistical Control Charts." *Health Care Management Science* 4:305–318.

Benneyan, J. (2001b). "Performance of Number-Between *g*-Type Statistical Control Charts for Monitoring Adverse Events." *Health Care Management Science* 4:319–336.

Benneyan, J. C. (1998a). "Statistical Quality Control Methods in Infection Control and Hospital Epidemiology, Part I: Introduction and Basic Theory." *Infection Control and Hospital Epidemiology* 19:194–214.

Benneyan, J. C. (1998b). "Statistical Quality Control Methods in Infection Control and Hospital Epidemiology, Part II: Chart Use, Statistical Properties, and Research Issues." *Infection Control and Hospital Epidemiology* 19:265–283.

Benneyan, J. C. (1999). "Geometric-Based *g*-Type Statistical Control Charts for Infrequent Adverse Events: New Quality Control Charts for Hospital Infections." In *Institute of Industrial Engineers Society for Health Systems 1999 Conference Proceedings*, 175–185. Norcross, GA: Institute of Industrial Engineers, Society for Health Systems.

Benneyan, J. C. (2006). "Discussion: Statistical Process Control Methods in Health Care." *Journal of Quality Technology* 38:113–123.

D'Agostino, R. B., and Stephens, M., eds. (1986). *Goodness-of-Fit Techniques*. New York: Marcel Dekker.

Kaminsky, F. C., Benneyan, J. C., Davis, R. D., and Burke, R. J. (1992). "Statistical Control Charts Based on a Geometric Distribution." *Journal of Quality Technology* 24:63–69.

Santiago, E., and Smith, J. (2013). "Control Charts Based on the Exponential Distribution: Adapting Runs Rules for the *t* Chart." *Quality Engineering* 25:85–96.

Terrell, G. R., and Scott, D. W. (1985). "Oversmoothed Nonparametric Density Estimates." *Journal of the American Statistical Association* 80:209–214.

Woodall, W. H. (2006). "The Use of Control Charts in Health-Care and Public-Health Surveillance." *Journal of Quality Technology* 38:89–104.

Chapter 18
The RELIABILITY Procedure

Contents

Overview: RELIABILITY Procedure **1204**
Getting Started: RELIABILITY Procedure **1206**
Analysis of Right-Censored Data from a Single Population 1206
Weibull Analysis Comparing Groups of Data 1210
Analysis of Accelerated Life Test Data 1214
Weibull Analysis of Interval Data with Common Inspection Schedule 1219
Lognormal Analysis with Arbitrary Censoring 1224
Regression Modeling 1229
Regression Model with Nonconstant Scale 1235
Regression Model with Two Independent Variables 1238
Weibull Probability Plot for Two Combined Failure Modes 1241
Analysis of Recurrence Data on Repairs 1245
Comparison of Two Samples of Repair Data 1250
Analysis of Interval Age Recurrence Data 1257
Analysis of Binomial Data 1260
Three-Parameter Weibull 1263
Parametric Model for Recurrent Events Data 1266
Parametric Model for Interval Recurrent Events Data 1268
Syntax: RELIABILITY Procedure **1271**
Primary Statements 1271
Secondary Statements 1271
Graphical Enhancement Statements 1273
PROC RELIABILITY Statement 1273
ANALYZE Statement 1273
BY Statement 1279
CLASS Statement 1279
DISTRIBUTION Statement 1280
EFFECTPLOT Statement 1281
ESTIMATE Statement 1282
FMODE Statement 1283
FREQ Statement 1284
INSET Statement 1284
LOGSCALE Statement 1288
LSMEANS Statement 1288
LSMESTIMATE Statement 1290
MAKE Statement 1291

MCFPLOT Statement 1291
MODEL Statement 1302
NENTER Statement 1310
NLOPTIONS Statement 1310
PROBPLOT Statement 1311
RELATIONPLOT Statement 1325
SLICE Statement 1338
STORE Statement 1339
TEST Statement 1339
UNITID Statement 1339
Details: RELIABILITY Procedure **1340**
Abbreviations and Notation 1340
Types of Lifetime Data 1340
Probability Distributions 1340
Probability Plotting 1343
Nonparametric Confidence Intervals for Cumulative Failure Probabilities 1352
Parameter Estimation and Confidence Intervals 1354
Regression Model Statistics Computed for Each Observation for Lifetime Data 1370
Regression Model Statistics Computed for Each Observation for Recurrent Events Data 1375
Recurrence Data from Repairable Systems 1376
ODS Table Names 1388
ODS Graphics 1390
References **1391**

Overview: RELIABILITY Procedure

The RELIABILITY procedure provides tools for reliability and survival data analysis and for recurrent events data analysis. You can use this procedure to

- construct probability plots and fitted life distributions with left-censored, right-censored, and interval-censored lifetime data
- fit regression models, including accelerated life test models, to combinations of left-censored, right-censored, and interval-censored data
- analyze recurrence data from repairable systems

These tools benefit reliability engineers and industrial statisticians working with product life data and system repair data. They also aid workers in other fields, such as medical research, pharmaceuticals, social sciences, and business, where survival and recurrence data are analyzed.

Most practical problems in reliability data analysis involve right-censored, left-censored, or interval-censored data. The RELIABILITY procedure provides probability plots of uncensored, right-censored, interval-censored, and arbitrarily censored data.

Features of the RELIABILITY procedure include

- probability plotting and parameter estimation for the common life distributions: Weibull, three-parameter Weibull, exponential, extreme value, normal, lognormal, logistic, and log-logistic. The data can be complete, right censored, or interval censored.
- maximum likelihood estimates of distribution parameters, percentiles, and reliability functions
- both asymptotic normal and likelihood ratio confidence intervals for distribution parameters and percentiles. Asymptotic normal confidence intervals for the reliability function are also available.
- estimation of distribution parameters by least squares fitting to the probability plot
- Weibayes analysis, where there are no failures and where the data analyst specifies a value for the Weibull shape parameter
- estimates of the resulting distribution when specified failure modes are eliminated
- plots of the data and the fitted relation for life versus stress in the analysis of accelerated life test data
- fitting of regression models to life data, where the life distribution location parameter is a linear function of covariates. The fitting yields maximum likelihood estimates of parameters of a regression model with a Weibull, exponential, extreme value, normal, lognormal, logistic and log-logistic, or generalized gamma distribution. The data can be complete, right censored, left censored, or interval censored. For example, accelerated life test data can be modeled with such a regression model.
- nonparametric estimates and plots of the mean cumulative function for cost or number of recurrences and associated confidence intervals from data with exact or interval recurrence ages
- maximum likelihood estimation of the parameters of parametric models for recurrent events data
- horizontal plots of failure times for recurrent events data

Some of the features provided in the RELIABILITY procedure are available in other SAS procedures.

- You can construct probability plots of life data with the CAPABILITY procedure; however, the CAPABILITY procedure is intended for process capability analysis rather than reliability analysis, and the data must be complete (that is, uncensored).
- The LIFEREG procedure fits regression models with life distributions such as the Weibull, lognormal, and log-logistic to left-, right-, and interval-censored data. The RELIABILITY procedure fits the same distributions and regression models as the LIFEREG procedure and, in addition, provides a graphical display of life data in probability plots.

Lawless (2003), Meeker and Escobar (1998), Nelson (1982, 1990), Abernethy (2006), and Tobias and Trindade (1995) provide many examples taken from diverse fields and describe the analyses provided by the RELIABILITY procedure.

The features of the procedure that deal with the nonparametric analysis of recurrent events data from repairable systems are based on the work of Doganaksoy and Nelson (1998), Nelson (1988, 1995, 2002), and Nelson and Doganaksoy (1989), who provide examples of repair data analysis. Meeker and Escobar (1998), Rigdon and Basu (2000), Cook and Lawless (2007), Abernethy (2006), Tobias and Trindade (1995), Crowder et al. (1991), and US Army (2000) provide details of parametric models for recurrent events data.

Getting Started: RELIABILITY Procedure

This section introduces the RELIABILITY procedure with examples that illustrate some of the analyses that it performs.

Analysis of Right-Censored Data from a Single Population

The Weibull distribution is used in a wide variety of reliability analysis applications. This example illustrates the use of the Weibull distribution to model product life data from a single population. The following statements create a SAS data set containing observed and right-censored lifetimes of 70 diesel engine fans (Nelson 1982, p. 318):

```
data fan;
   input Lifetime censor @@;
   Lifetime = Lifetime/1000;
   label lifetime='Fan Life (1000s of Hours)';
   datalines;
 450 0    460 1   1150 0   1150 0   1560 1
1600 0   1660 1   1850 1   1850 1   1850 1
1850 1   1850 1   2030 1   2030 1   2030 1
2070 0   2070 0   2080 0   2200 1   3000 1
3000 1   3000 1   3000 1   3100 0   3200 1
3450 0   3750 1   3750 1   4150 1   4150 1
4150 1   4150 1   4300 1   4300 1   4300 1
4300 1   4600 0   4850 1   4850 1   4850 1
4850 1   5000 1   5000 1   5000 1   6100 1
6100 0   6100 1   6100 1   6300 1   6450 1
6450 1   6700 1   7450 1   7800 1   7800 1
8100 1   8100 1   8200 1   8500 1   8500 1
8500 1   8750 1   8750 0   8750 1   9400 1
9900 1  10100 1  10100 1  10100 1  11500 1
;
```

Some of the fans had not failed at the time the data were collected, and the unfailed units have right-censored lifetimes. The variable Lifetime represents either a failure time or a censoring time in thousands of hours. The variable Censor is equal to 0 if the value of Lifetime is a failure time, and it is equal to 1 if the value is a censoring time.

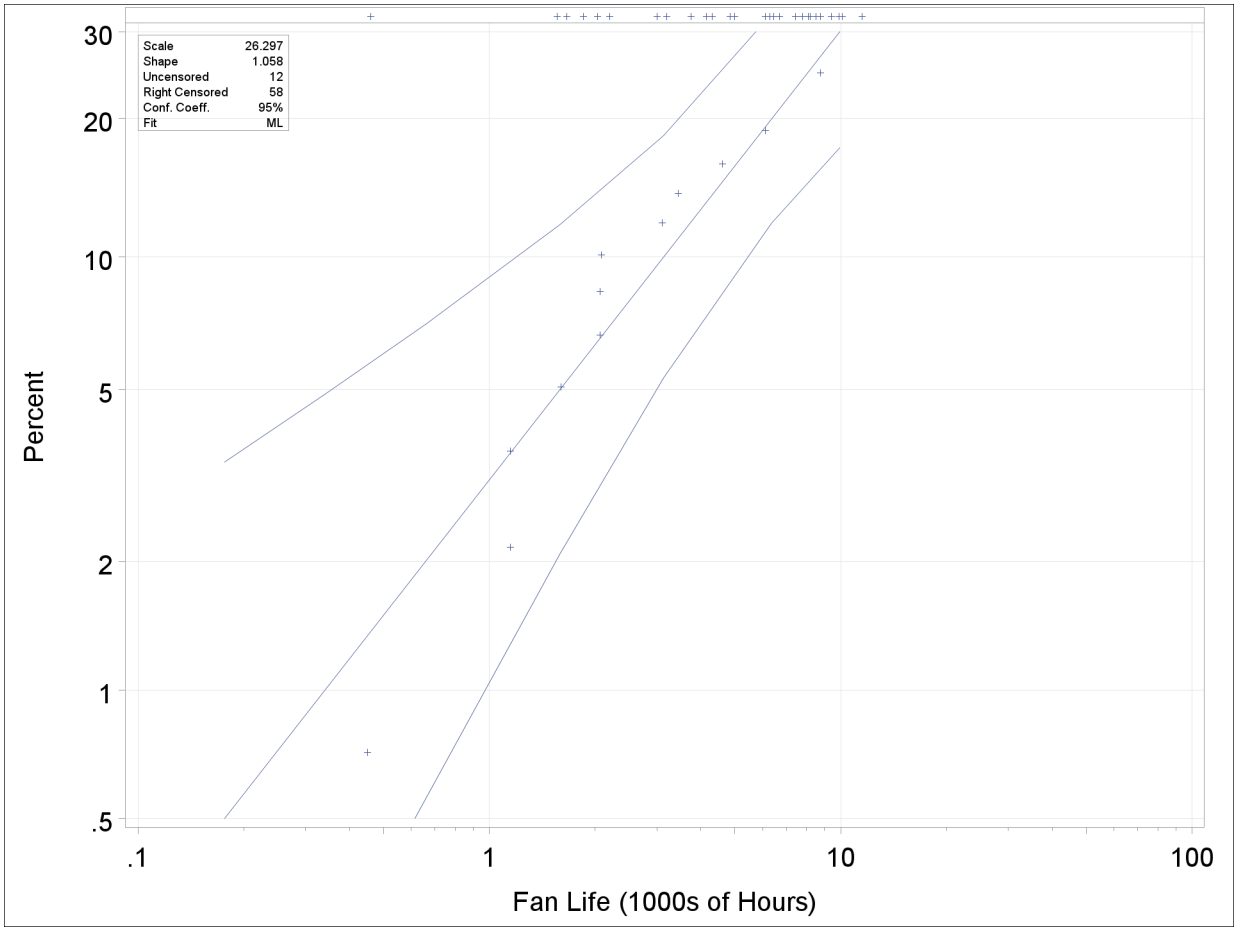
If ODS Graphics is disabled, graphical output is created using traditional graphics; otherwise, ODS Graphics is used. The following statements use the RELIABILITY procedure to produce the traditional graphical output shown in Figure 18.1:

```
ODS Graphics OFF;
proc reliability data=fan;
   distribution Weibull;
   pplot lifetime*censor( 1 ) /  covb ;
run;
ODS Graphics ON;
```

The DISTRIBUTION statement specifies the Weibull distribution for probability plotting and maximum likelihood (ML) parameter estimation. The PROBPLOT statement produces a probability plot for the variable Lifetime and specifies that the value of 1 for the variable Censor denotes censored observations. You can specify any value, or group of values, for the *censor-variable* (in this case, Censor) to indicate censoring times. The option COVB requests the ML parameter estimate covariance matrix.

The graphical output, displayed in Figure 18.1, consists of a probability plot of the data, an ML fitted distribution line, and confidence intervals for the percentile (lifetime) values. An *inset* box containing summary statistics, Weibull scale and shape estimates, and other information is displayed on the plot by default. The locations of the right-censored data values are plotted as plus signs in an area at the top of the plot.

Figure 18.1 Weibull Probability Plot for Engine Fan Data (Traditional Graphics)

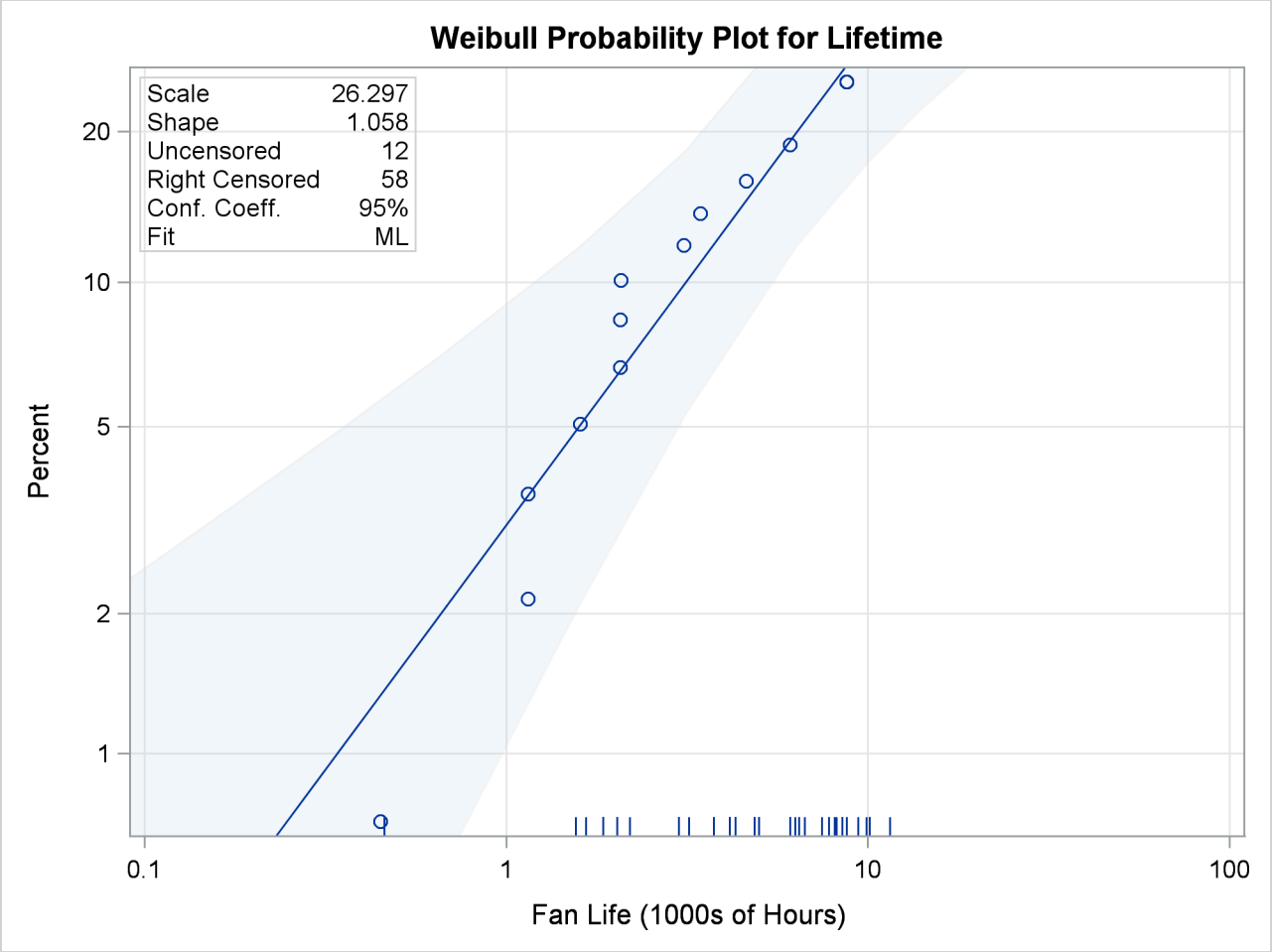


If ODS Graphics is enabled, you can create the probability plot by using ODS Graphics. The following SAS statements use ODS Graphics to create the probability plot shown in Figure 18.1:

```
proc reliability data=fan;
   distribution Weibull;
   pplot lifetime*censor( 1 ) /  covb;
run;
```

The plot is shown in Figure 18.2.

Figure 18.2 Weibull Probability Plot for Engine Fan Data (ODS Graphics)



The tabular output produced by the preceding SAS statements is shown in Figure 18.3 and Figure 18.4. This consists of summary data, fit information, parameter estimates, distribution percentile estimates, standard errors, and confidence intervals for all estimated quantities.

Figure 18.3 Tabular Output for the Fan Data Analysis

The RELIABILITY Procedure

Model Information	
Input Data Set	WORK.FAN
Analysis Variable	Lifetime Fan Life (1000s of Hours)
Censor Variable	censor
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	70

Algorithm converged.

Figure 18.3 *continued*

Summary of Fit	
Observations Used	70
Uncensored Values	12
Right Censored Values	58
Maximum Loglikelihood	-42.248

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
EV Location	3.2694	0.4659	2.3563	4.1826
EV Scale	0.9448	0.2394	0.5749	1.5526
Weibull Scale	26.2968	12.2514	10.5521	65.5344
Weibull Shape	1.0584	0.2683	0.6441	1.7394

Other Weibull Distribution Parameters

Parameter	Value
Mean	25.7156
Mode	1.7039
Median	18.6002
Standard Deviation	24.3066

Estimated Covariance Matrix Weibull Parameters

	EV Location	EV Scale
EV Location	0.21705	0.09044
EV Scale	0.09044	0.05733

Estimated Covariance Matrix Weibull Parameters

	Weibull Scale	Weibull Shape
Weibull Scale	150.09724	-2.66446
Weibull Shape	-2.66446	0.07196

Figure 18.4 Percentile Estimates for the Fan Data

Weibull Percentile Estimates

Percent	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
0.1	0.03852697	0.05027782	0.002985	0.49726229
0.2	0.07419554	0.08481353	0.00789519	0.69725757
0.5	0.17658807	0.16443381	0.02846732	1.09540855
1	0.34072273	0.2635302	0.07482449	1.55152389
2	0.65900116	0.40845639	0.19556981	2.22060107
5	1.58925244	0.68465855	0.68311002	3.69738878
10	3.13724079	0.99379006	1.68620756	5.83693255
20	6.37467675	1.74261908	3.73051433	10.8930029
30	9.92885165	3.00353842	5.48788931	17.9635721
40	13.9407124	4.85766683	7.04177638	27.5986417
50	18.6002319	7.40416922	8.52475116	40.5840149
60	24.2121441	10.8733301	10.0408557	58.3842593
70	31.3378076	15.750336	11.7018888	83.9230489
80	41.2254517	23.1787018	13.6956839	124.092954
90	57.8253251	36.9266698	16.5405275	202.156081
95	74.1471722	51.6127806	18.9489625	290.137423
99	111.307797	88.1380261	23.5781482	525.462197
99.9	163.265082	144.264145	28.8905203	922.637827

Weibull Analysis Comparing Groups of Data

This example illustrates probability plotting and distribution fitting for data grouped by the levels of a special *group-variable*. The data are from an accelerated life test of an insulating fluid and are the times to electrical breakdown of the fluid under different high voltage levels. Each voltage level defines a subset of data for which a separate analysis and Weibull plot are produced. These data are the 26kV, 30kV, 34kV, and 38kV groups of the data provided by Nelson (1990, p. 129). The following statements create a SAS data set containing the lifetimes and voltages:

```
data fluid;
   input Time voltage $ @@;
   datalines;
5.79    26kv    1579.52 26kv
2323.7  26kv    7.74    30kv
17.05   30kv    20.46   30kv
21.02   30kv    22.66   30kv
43.4    30kv    47.3    30kv
139.07  30kv    144.12  30kv
175.88  30kv    194.90  30kv
0.19     34kv    .78     34kv
0.96     34kv    1.31    34kv
2.78    34kv    3.16    34kv
4.15    34kv    4.67    34kv
4.85    34kv    6.50    34kv
7.35    34kv    8.01    34kv
8.27    34kv    12.06   34kv
31.75   34kv    32.52   34kv
33.91   34kv    36.71   34kv
72.89   34kv    .09     38kv
0.39     38kv    .47     38kv
0.73     38kv    .74     38kv
1.13    38kv    1.40    38kv
2.38    38kv
;
```

The variable Time provides the time to breakdown in minutes, and the variable Voltage provides the voltage level at which the test was conducted. These data are not censored.

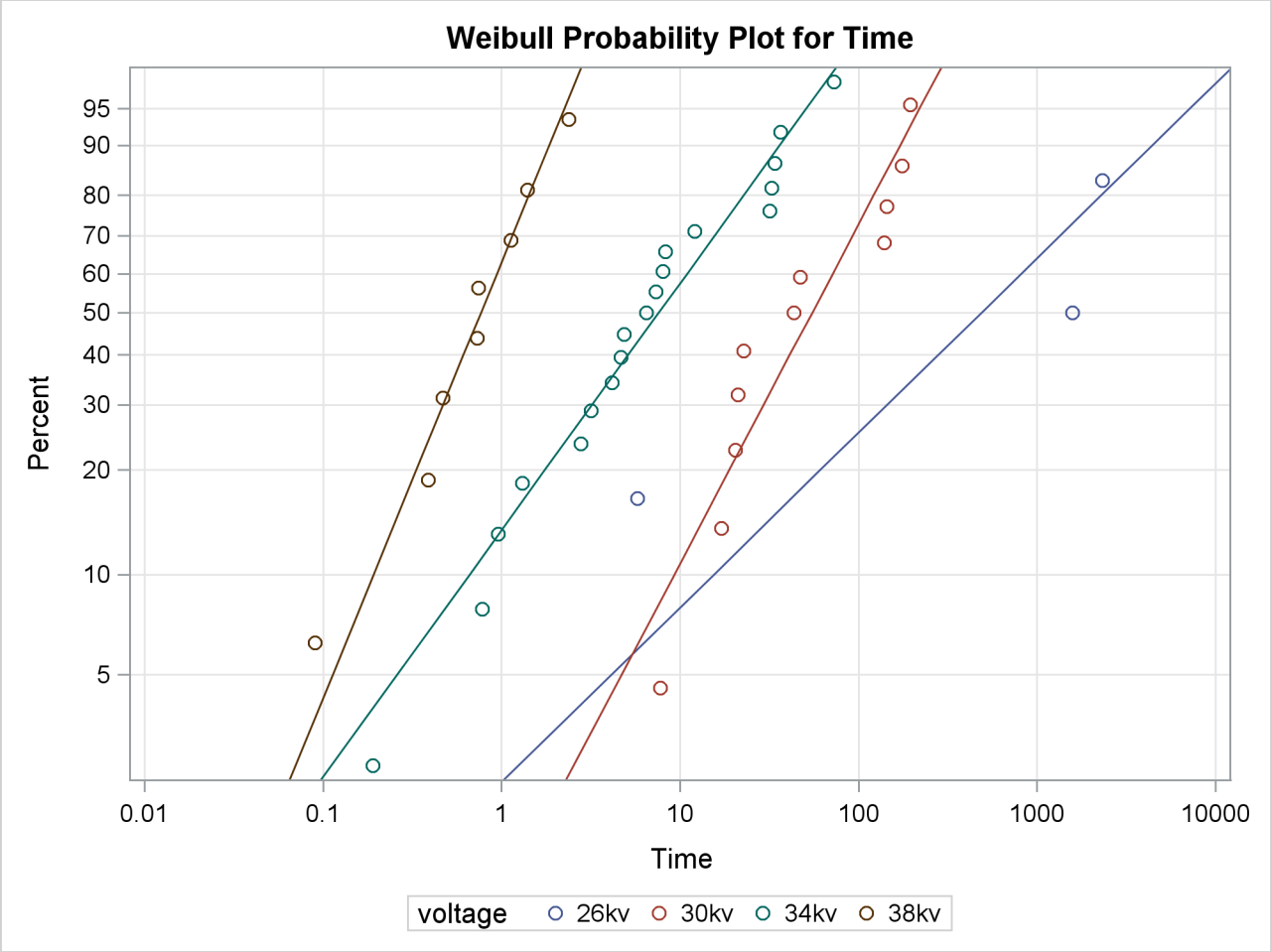
The RELIABILITY procedure plots the data for the different voltage levels on the same Weibull probability plot, fits a separate distribution to the data at each voltage level, and superimposes distribution lines on the plot.

The following statements produce the probability plot shown in Figure 18.5 for the variable Time at each level of the *group-variable* Voltage:

```
proc reliability data=fluid;
   distribution Weibull;
   pplot time=voltage  / overlay
                         noconf;
run;
```

The input data set FLUID is specified by the DATA= option in the PROC RELIABILITY statement. The PROBPLOT statement option OVERLAY specifies that plots for the groups are to be overlaid rather than displayed separately. The option NOCONF specifies that no confidence bands are to be plotted, since these can interfere with one another on overlaid plots; confidence bands are displayed by default.

Figure 18.5 Weibull Probability Plot for the Insulating Fluid Data



A summary table that contains information for all groups is displayed. In addition, information identical to that shown in Figure 18.3 is tabulated for each level of voltage. The summary table for all groups and the tables for the 26kV group are shown in Figure 18.6 and Figure 18.7.

Figure 18.6 Partial Listing of the Tabular Output for the Insulating Fluid Data

The RELIABILITY Procedure

Model Information - All Groups	
Input Data Set	WORK.FLUID
Analysis Variable	Time
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	41

The RELIABILITY Procedure

Algorithm converged for group 26kv.

Figure 18.6 *continued*

Summary of Fit		
		Group
Observations Used	3	26kv
Uncensored Values	3	26kv
Maximum Loglikelihood	-6.845551	26kv

Figure 18.7 Partial Listing of the Tabular Output for the Insulating Fluid Data

The RELIABILITY Procedure

Model Information - All Groups	
Input Data Set	WORK.FLUID
Analysis Variable	Time
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	41

Weibull Parameter Estimates					
			Asymptotic Normal 95% Confidence Limits		
Parameter	**Estimate**	**Standard Error**	**Lower**	**Upper**	**Group**
EV Location	6.8625	1.1040	4.6986	9.0264	26kv
EV Scale	1.8342	0.9611	0.6568	5.1226	26kv
Weibull Scale	955.7467	1055.1862	109.7941	8319.6794	26kv
Weibull Shape	0.5452	0.2857	0.1952	1.5226	26kv

Other Weibull Distribution Parameters		
Parameter	**Value**	**Group**
Mean	1649.4882	26kv
Mode	0.0000	26kv
Median	487.9547	26kv
Standard Deviation	3279.0212	26kv

Figure 18.7 *continued*

Weibull Percentile Estimates

Percent	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper	Group
0.1	0.00300636	0.02113841	3.11203E-9	2904.27046	26kv
0.2	0.01072998	0.06838144	4.03597E-8	2852.65767	26kv
0.5	0.0577713	0.31803193	1.19079E-6	2802.78862	26kv
1	0.20695478	1.00385021	0.00001538	2784.16263	26kv
2	0.74484901	3.12705686	0.00019885	2790.0941	26kv
5	4.1142692	13.7388263	0.00591379	2862.3304	26kv
10	15.406565	41.4763373	0.07873508	3014.69497	26kv
20	61.0231127	125.020566	1.10053199	3383.65475	26kv
30	144.246801	242.203982	5.36856883	3875.73303	26kv
40	278.770459	398.048692	16.9761581	4577.77125	26kv
50	487.954708	610.02855	42.0948552	5656.26835	26kv
60	814.147288	920.537706	88.770543	7466.84412	26kv
70	1343.42243	1433.97868	165.818889	10884.0666	26kv
80	2287.87124	2445.52431	281.5628	18590.3635	26kv
90	4412.96962	5148.34986	448.419608	43428.7452	26kv
95	7150.89745	9248.2654	566.892142	90202.9338	26kv
99	15735.8513	24666.0388	728.831025	339745.437	26kv
99.9	33104.172	62018.1074	841.826189	1301796.28	26kv

Analysis of Accelerated Life Test Data

The following example illustrates the analysis of an accelerated life test for Class B electrical motor insulation. The data are provided by Nelson (1990, p. 243). Forty insulation specimens were tested at four temperatures: 150°, 170°, 190°, and 220°C. The purpose of the test is to estimate the median life of the insulation at the design operating temperature of 130°C.

The following SAS program creates the data listed in Figure 18.8. Ten specimens of the insulation were tested at each test temperature. The variable Time provides a specimen time to failure or a censoring time, in hours. The variable Censor is equal to 1 if the value of the variable Time is a right-censoring time and is equal to 0 if the value is a failure time. Some censor times and failure times are identical at some of the temperatures. Rather than repeating identical observations in the input data set, the variable Count provides the number of specimens with identical times and temperatures. The variable Temp provides the test temperature in degrees centigrade. The variable Cntrsl is a control variable specifying that percentiles are to be computed only for the first value of Temp (130°C). The value of Temp in the first observation (130°C) does not correspond to a test temperature. The missing values in the first observation cause the observation to be excluded from the model fit, and the value of 1 for the variable Cntrl causes percentiles corresponding to a temperature of 130°C to be computed.

```
data classb;
   input hours temp count censor;
   if _n_ = 1 then cntrl=1;
   else cntrl=0;
   label hours='Hours';
   datalines;
   . 130 .  .
8064 150 10 1
1764 170 1  0
2772 170 1  0
3444 170 1  0
3542 170 1  0
3780 170 1  0
4860 170 1  0
5196 170 1  0
5448 170 3  1
 408 190 2  0
1344 190 2  0
1440 190 1  0
1680 190 5  1
 408 220 2  0
 504 220 3  0
 528 220 5  1
;
```

Figure 18.8 Listing of the Class B Insulation Data

Obs	hours	temp	count	censor	cntrl
1	.	130	.	.	1
2	8064	150	10	1	0
3	1764	170	1	0	0
4	2772	170	1	0	0
5	3444	170	1	0	0
6	3542	170	1	0	0
7	3780	170	1	0	0
8	4860	170	1	0	0
9	5196	170	1	0	0
10	5448	170	3	1	0
11	408	190	2	0	0
12	1344	190	2	0	0
13	1440	190	1	0	0
14	1680	190	5	1	0
15	408	220	2	0	0
16	504	220	3	0	0
17	528	220	5	1	0

An Arrhenius-lognormal model is fitted to the data in this example. In other words, the failure times follow a lognormal (base 10) distribution, and the lognormal location parameter μ depends on the centigrade temperature Temp through the Arrhenius relationship

$$\mu(x) = \beta_0 + \beta_1 x$$

where

$$x = \frac{1000}{\text{Temp} + 273.15}$$

is 1000 times the reciprocal absolute temperature. The lognormal (base e) distribution is also available.

The following SAS statements fit the Arrhenius-lognormal model, and they display the fitted model distributions side-by-side on the probability and the relation plots shown in Figure 18.9:

```
proc reliability;
   distribution lognormal10;
   freq count;
   model hours*censor(1) = temp /
      relation = arr
      obstats(quantile = .1 .5 .9 control = cntrl);
   rplot hours*censor(1) = temp /
      pplot
      fit = model
      noconf
      relation = arr
      plotdata
      plotfit 10 50 90
      lupper = 1.e5
      slower = 120;
run;
```

The PROC RELIABILITY statement invokes the procedure and specifies CLASSB as the input data set. The DISTRIBUTION statement specifies that the lognormal (base 10) distribution is to be used for maximum likelihood parameter estimation and probability plotting. The FREQ statement specifies that the variable Count is to be used as a frequency variable; that is, if Count=n, then there are n specimens with the time and temperature specified in the observation.

The MODEL statement fits a linear regression equation for the distribution location parameter as a function of independent variables. In this case, the MODEL statement also transforms the independent variable through the Arrhenius relationship. The dependent variable is specified as Time. A value of 1 for the variable Censor indicates that the corresponding value of Time is a right-censored observation; otherwise, the value is a failure time. The temperature variable Temp is specified as the independent variable in the model. The MODEL statement option RELATION=ARR specifies the Arrhenius relationship.

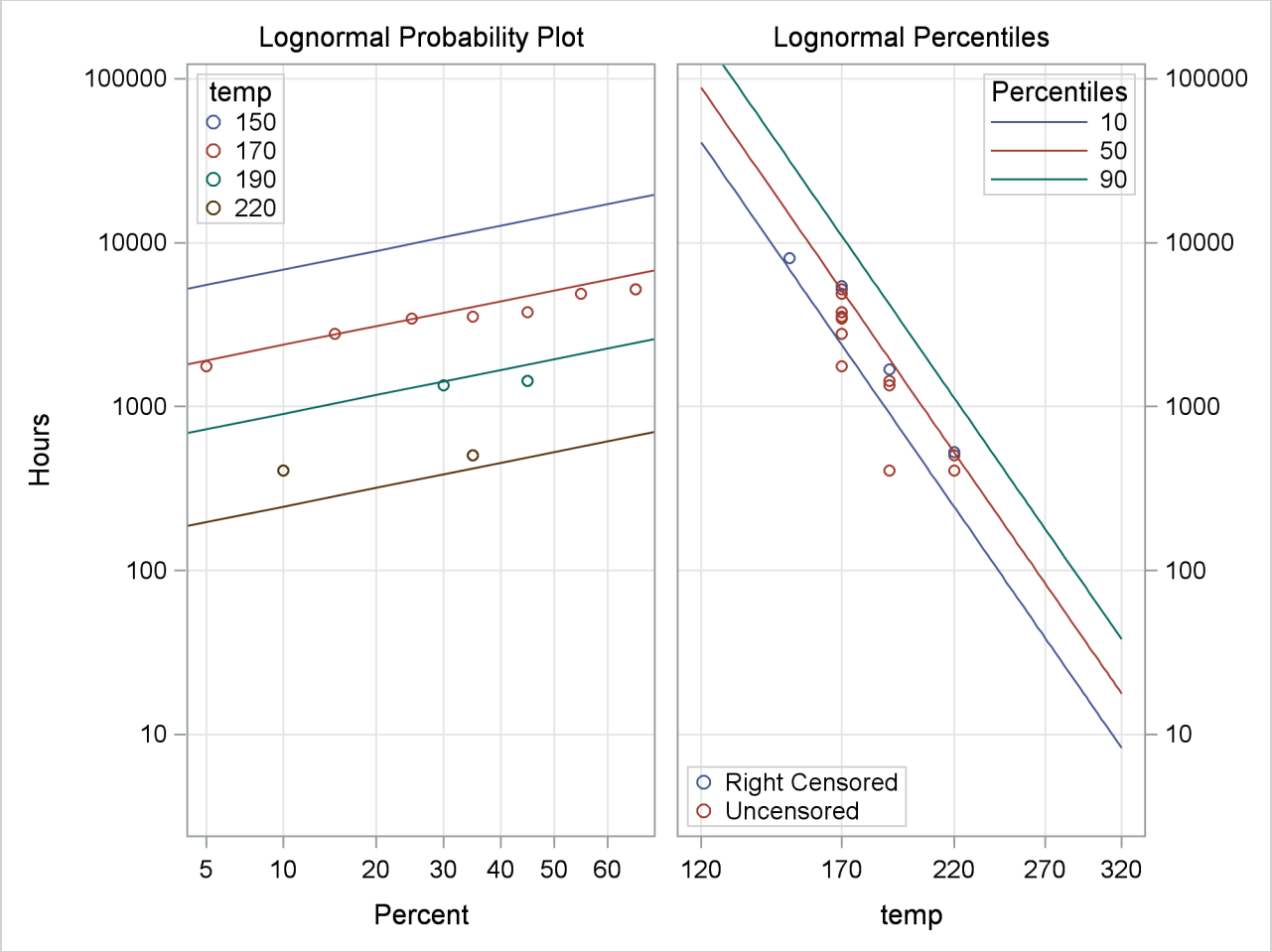
The option OBSTATS requests statistics computed for each observation in the input data set. The options in parentheses following OBSTATS indicate which statistics are to be computed. In this case, QUANTILE=.1 .5 .9 specifies that quantiles of the fitted distribution are to be computed for the value of the variable Temp at each observation. The CONTROL= option requests quantiles only for those observations in which the variable Cntrl has a value of 1. This eliminates unnecessary quantiles in the OBSTATS table since, in this case, only the quantiles at the design temperature of 130°C are of interest.

The RPLOT, or RELATIONPLOT, statement displays a plot of the lifetime data and the fitted model. The dependent variable Time, the independent variable Temp, and the censoring indicator Censor are the same as in the MODEL statement. The option FIT=MODEL specifies that the model fitted with the preceding MODEL statement is to be used for probability plotting and in the relation plot. The option RELATION=ARR specifies an Arrhenius scale for the horizontal axis of the relation plot. The PPLOT option specifies that a probability plot is to be displayed alongside the relation plot. The type of probability plot is determined by the distribution named in the DISTRIBUTION statement, in this case, a lognormal (base 10) distribution. Weibull, extreme value, lognormal (base *e*), normal, log-logistic, and logistic distributions are also available. The NOCONF option suppresses the default percentile confidence bands on the probability plot. The PLOTDATA option specifies that the failure times are to be plotted on the relation plot. The PLOTFIT option specifies that the 10th, 50th, and 90th percentiles of the fitted relationship are to be plotted on the relation plot. The options LUPPER and SLOWER specify an upper limit on the life axis scale and a lower limit on the stress (temperature) axis scale in the plots.

The plots produced by the preceding statements are shown in Figure 18.9. The plot on the left is an overlaid lognormal probability plot of the data and the fitted model. The plot on the right is a relation plot showing the data and the fitted relation. The fitted straight lines are percentiles of the fitted distribution at each temperature. An Arrhenius relation fitted to the data, plotted on an Arrhenius plot, yields straight percentile lines.

Since all the data at 150°C are right censored, there are no failures corresponding to 150°C on the probability plot. However, the fitted distribution at 150°C is plotted on the probability plot.

Figure 18.9 Probability and Relation Plots for the Class B Insulation Data



The tabular output requested with the MODEL statement is shown in Figure 18.10. The "Model Information" table provides general information about the data and model. The "Summary of Fit" table shows the number of observations used, the number of failures and of censored values (accounting for the frequency count), and the maximum log likelihood for the fitted model.

The "Lognormal Parameter Estimates" table contains the Arrhenius-lognormal model parameter estimates, their standard errors, and confidence interval estimates. In this table, INTERCEPT is the maximum likelihood estimate of β_0, TEMP is the estimate of β_1, and Scale is the estimate of the lognormal scale parameter, σ.

Figure 18.10 MODEL Statement Output for the Class B Data

The RELIABILITY Procedure

Model Information	
Input Data Set	WORK.CLASSB
Analysis Variable	hours Hours
Relation	Arrhenius(temp)
Censor Variable	censor
Frequency Variable	count
Distribution	Lognormal (Base 10)

Figure 18.10 *continued*

Algorithm converged.

Summary of Fit	
Observations Used	16
Uncensored Values	17
Right Censored Values	23
Missing Observations	1
Maximum Loglikelihood	-12.96533

Lognormal Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	-6.0182	0.9467	-7.8737	-4.1628
temp	4.3103	0.4366	3.4546	5.1660
Scale	0.2592	0.0473	0.1812	0.3708

Observation Statistics

Hours	censor	temp	count	Prob	Pcntl	Stderr	Lower	Upper
.	.	130	.	0.1000	21937.658	6959.151	11780.636	40851.857
.	.	130	.	0.5000	47135.132	16125.548	24106.685	92162.016
.	.	130	.	0.9000	101274.29	42061.1	44872.401	228569.92

The "Observation Statistics" table provides the estimates of the fitted distribution quantiles, their standard errors, and the confidence limits. These are given only for the value of 130°C, as specified with the CONTROL= option in the MODEL statement. The predicted median life at 130°C corresponds to a quantile of 0.5, and it is approximately 47,135 hours.

In addition to the MODEL statement output in Figure 18.10, the RELIABILITY procedure produces tabular output for each temperature that is identical to the output produced with the PROBPLOT statement. This output is not shown.

Weibull Analysis of Interval Data with Common Inspection Schedule

Table 18.1 shows data for 167 identical turbine parts provided by Nelson (1982, p. 415). The parts were inspected at certain times to determine which parts had cracked since the last inspection. The times at which parts develop cracks are to be fitted with a Weibull distribution.

Table 18.1 Turbine Part Cracking Data

Inspection (Months)		Number	
Start	**End**	**Cracked**	**Cumulative**
0	6.12	5	5
6.12	19.92	16	21
19.92	29.64	12	33
29.64	35.40	18	51
35.40	39.72	18	69
39.72	45.24	2	71
45.24	52.32	6	77
52.32	63.48	17	94
63.48	Survived	73	167

Table 18.1 shows the time in months of each inspection period and the number of cracked parts found in each period. These data are said to be interval censored since only the time interval in which failures occurred is known, not the exact failure times. Seventy-three parts had not cracked at the last inspection, which took place at 63.48 months. These 73 lifetimes are right censored, since the lifetimes are known only to be greater than 63.48 months.

The interval data in this example are read from a SAS data set with a special structure. All units must have a common inspection schedule. This type of interval data is called *readout data*. The following SAS program creates the SAS data set named CRACKS, shown in Figure 18.11, and provides the data in Table 18.1 with this structure:

```
data cracks;
   input Time units fail;
   datalines;
6.12  167 5
19.92 162 16
29.64 146 12
35.4  134 18
39.72 116 18
45.24 98  2
52.32 96  6
63.48 90 17
;
```

The variable Time is the inspection time—that is, the upper endpoint of each interval. The variable Units is the number of unfailed units at the beginning of each interval, and the variable Fail is the number of units with cracks at the inspection time.

Figure 18.11 Listing of the Turbine Part Cracking Data

Obs	Time	units	fail
1	6.12	167	5
2	19.92	162	16
3	29.64	146	12
4	35.40	134	18
5	39.72	116	18
6	45.24	98	2
7	52.32	96	6
8	63.48	90	17

The following statements use the RELIABILITY procedure to produce the probability plot in Figure 18.12 for the data in the data set CRACKS:

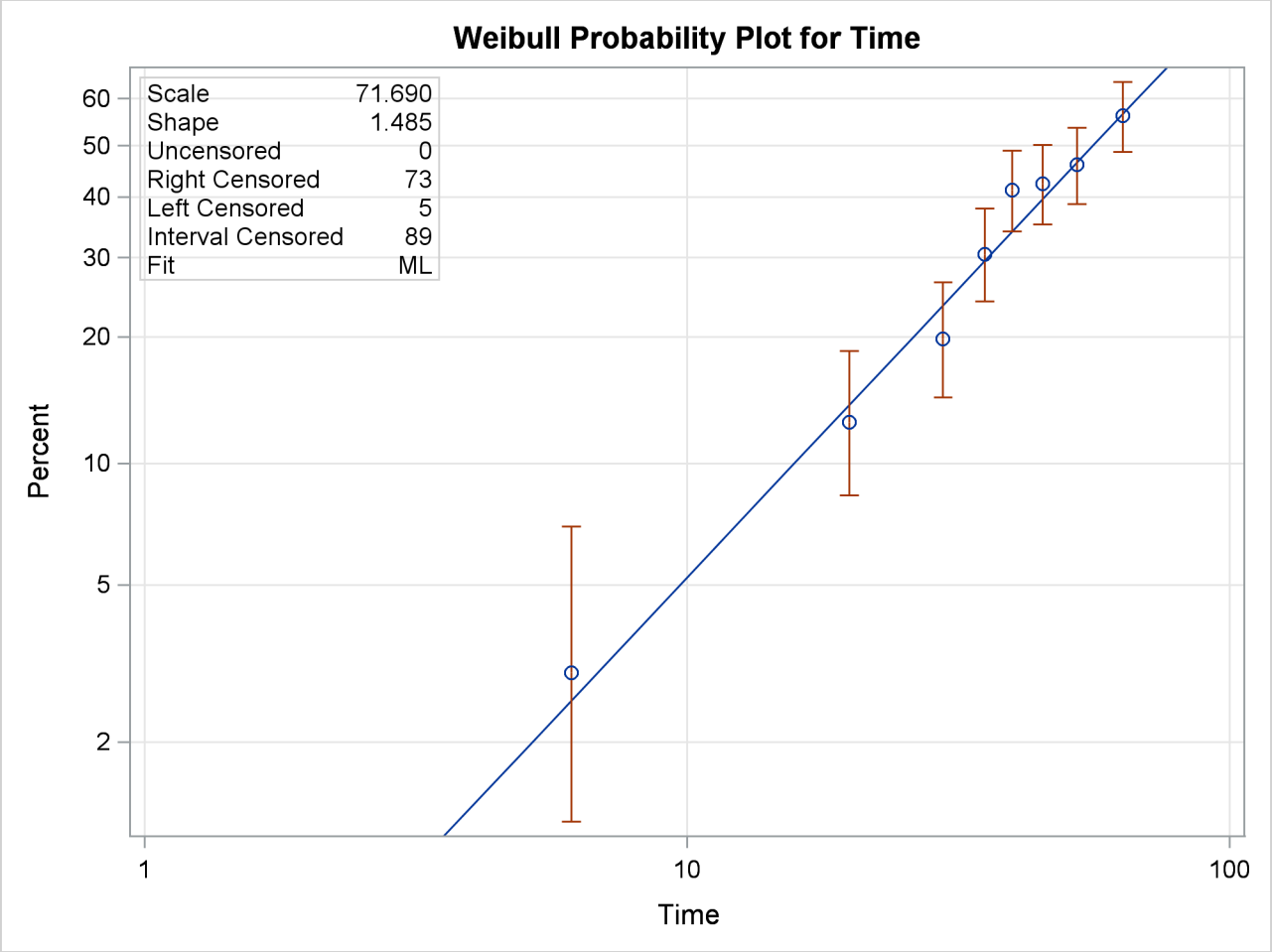
```
proc reliability data=cracks;
   freq fail;
   nenter units;
   distribution Weibull;
   probplot time / readout
                   ppout
                   pconfplt
                   noconf;
run;
```

The FREQ statement specifies that the variable Fail provides the number of failures in each interval. The NENTER statement specifies that the variable Units provides the number of unfailed units at the beginning of each interval. The DISTRIBUTION statement specifies that the Weibull distribution be used for parameter estimation and probability plotting. The PROBPLOT statement requests a probability plot of the data.

The PROBPLOT statement option READOUT indicates that the data in the CRACKS data set are readout (or interval) data. The option PCONFPLT specifies that confidence intervals for the cumulative probability of failure be plotted. The confidence intervals for the cumulative probability are based on the binomial distribution for time intervals until right censoring occurs. For time intervals after right censoring occurs, the binomial distribution is not valid, and a normal approximation is used to compute confidence intervals.

The option NOCONF suppresses the display of confidence intervals for distribution percentiles in the probability plot.

Figure 18.12 Weibull Probability Plot for the Part Cracking Data



A listing of the tabular output produced by the preceding SAS statements is shown in Figure 18.13 and Figure 18.14. By default, the specified Weibull distribution is fitted by maximum likelihood. The line plotted on the probability plot and the tabular output summarize this fit. For interval data, the estimated cumulative probabilities and associated confidence intervals are tabulated. In addition, general fit information, parameter estimates, percentile estimates, standard errors, and confidence intervals are tabulated.

Figure 18.13 Partial Listing of the Tabular Output for the Part Cracking Data

The RELIABILITY Procedure

Model Information	
Input Data Set	WORK.CRACKS
Analysis Variable	Time
Frequency Variable	fail
NENTER Variable	units
Distribution	Weibull
Estimation Method	Maximum Likelihood
Confidence Coefficient	95%
Observations Used	8

Figure 18.13 *continued*

Cumulative Probability Estimates

Lower Lifetime	Upper Lifetime	Cumulative Probability	Pointwise 95% Confidence Limits		Standard Error
			Lower	Upper	
.	6.12	0.0299	0.0125	0.0699	0.0132
6.12	19.92	0.1257	0.0834	0.1852	0.0257
19.92	29.64	0.1976	0.1440	0.2649	0.0308
29.64	35.4	0.3054	0.2403	0.3793	0.0356
35.4	39.72	0.4132	0.3410	0.4893	0.0381
39.72	45.24	0.4251	0.3524	0.5013	0.0383
45.24	52.32	0.4611	0.3869	0.5370	0.0386
52.32	63.48	0.5629	0.4868	0.6361	0.0384

Algorithm converged.

Summary of Fit	
Observations Used	8
Right Censored Values	73
Left Censored Values	5
Interval Censored Values	89
Maximum Loglikelihood	-309.6684

Figure 18.14 Partial Listing of the Tabular Output for the Part Cracking Data

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits	
			Lower	Upper
EV Location	4.2724	0.0744	4.1265	4.4182
EV Scale	0.6732	0.0664	0.5549	0.8168
Weibull Scale	71.6904	5.3335	61.9634	82.9444
Weibull Shape	1.4854	0.1465	1.2242	1.8022

Other Weibull Distribution Parameters

Parameter	Value
Mean	64.7966
Mode	33.7622
Median	56.0144
Standard Deviation	44.3943

Figure 18.14 *continued*

Weibull Percentile Estimates				
			Asymptotic Normal 95% Confidence Limits	
Percent	Estimate	Standard Error	Lower	Upper
0.1	0.68534385	0.29999861	0.29060848	1.61625083
0.2	1.09324674	0.42889777	0.50673224	2.3586193
0.5	2.02798319	0.67429625	1.05692279	3.8912169
1	3.23938972	0.93123832	1.84401909	5.69063837
2	5.18330703	1.2581604	3.22101028	8.34106988
5	9.70579945	1.78869256	6.76335893	13.9283666
10	15.7577991	2.22445157	11.9491109	20.7804776
20	26.1159906	2.6327383	21.4337103	31.821134
30	35.8126238	2.90557264	30.547517	41.9852137
40	45.6100472	3.27409792	39.6239146	52.5005271
50	56.0143651	3.89410377	48.8792027	64.1910859
60	67.5928125	4.90210777	58.6364803	77.917165
70	81.2334227	6.46932648	69.4938134	94.9562075
80	98.7644937	8.95137184	82.6900902	117.963654
90	125.694556	13.5078386	101.821995	155.164133
95	150.057755	18.2060035	118.300075	190.340791
99	200.437864	29.1957544	150.658574	266.66479
99.9	263.348102	44.7205513	188.791789	367.347666

In this example, the number of unfailed units at the beginning of an interval minus the number failing in the interval is equal to the number of unfailed units entering the next interval. This is not always the case since some unfailed units might be removed from the test at the end of an interval, for reasons unrelated to failure; that is, they might be right censored. The special structure of the input SAS data set required for interval data enables the RELIABILITY procedure to analyze this more general case.

Lognormal Analysis with Arbitrary Censoring

This example illustrates analyzing data that have more general censoring than in the previous example. The data can be a combination of exact failure times, left censored, right censored, and interval censored data. The intervals can be overlapping, unlike in the previous example, where the interval endpoints had to be the same for all units.

Table 18.2 shows data from Nelson (1982, p. 409), analyzed by Meeker and Escobar (1998, p. 135). Each of 435 turbine wheels was inspected once to determine whether a crack had developed in the wheel or not. The inspection time (in 100s of hours), the number inspected at the time that had cracked, and the number not cracked are shown in the table. The quantity of interest is the time for a crack to develop.

Table 18.2 Turbine Wheel Cracking Data

Inspection Time (100 hours)	Number Cracked	Number Not Cracked
4	0	39
10	4	49
14	2	31
18	7	66
22	5	25
26	9	30
30	9	33
34	6	7
38	22	12
42	21	19
46	21	15

These data consist only of left and right censored lifetimes. If a unit exhibits a crack at an inspection time, the unit is left censored at the time; if a unit has not developed a crack, it is right censored at the time. For example, there are 4 left-censored lifetimes and 49 right-censored lifetimes at 1000 hours.

The following statements create a SAS data set named TURBINE that contains the data in the format necessary for analysis by the RELIABILITY procedure:

```
data turbine;
   label t1 = 'Time of Cracking (Hours x 100 )';
   input t1 t2 f;
   datalines;
.   4  0
4   .  39
.  10  4
10  .  49
.  14  2
14  .  31
.  18  7
18  .  66
.  22  5
22  .  25
.  26  9
26  .  30
.  30  9
30  .  33
.  34  6
34  .  7
.  38  22
38  .  12
.  42  21
42  .  19
.  46  21
46  .  15
;
```

The variables T1 and T2 represent the inspection times and determine whether the observation is right or left censored. If T1 is missing (.), then T2 represents a left-censoring time; if T2 is missing, T1 represents a right-censoring time. The variable F is the number of units that were found to be cracked for left-censored observations, or not cracked for right-censored observations at an inspection time.

The following statements use the RELIABILITY procedure to produce the probability plot in Figure 18.15 for the data in the data set TURBINE:

```
proc reliability data = turbine;
   distribution lognormal;
   freq f;
   pplot ( t1 t2 ) / maxitem = 5000
                     ppout;
run;
```

The DISTRIBUTION statement specifies that a lognormal probability plot be created. The FREQ statement identifies the frequency variable F. The option MAXITEM=5000 specifies that the iterative algorithm that computes the points on the probability plot takes a maximum of 5000 iterations. The algorithm does not converge for these data in the default 1000 iterations, so the maximum number of iterations needs to be increased for convergence. The option PPOUT specifies that a table of the cumulative probabilities plotted on the probability plot be printed, along with standard errors and confidence limits.

The tabular output for the maximum likelihood lognormal fit for these data is shown in Figure 18.16. Figure 18.15 shows the resulting lognormal probability plot with the computed cumulative probability estimates and the lognormal fit line.

Figure 18.15 Lognormal Probability Plot for the Turbine Wheel Data

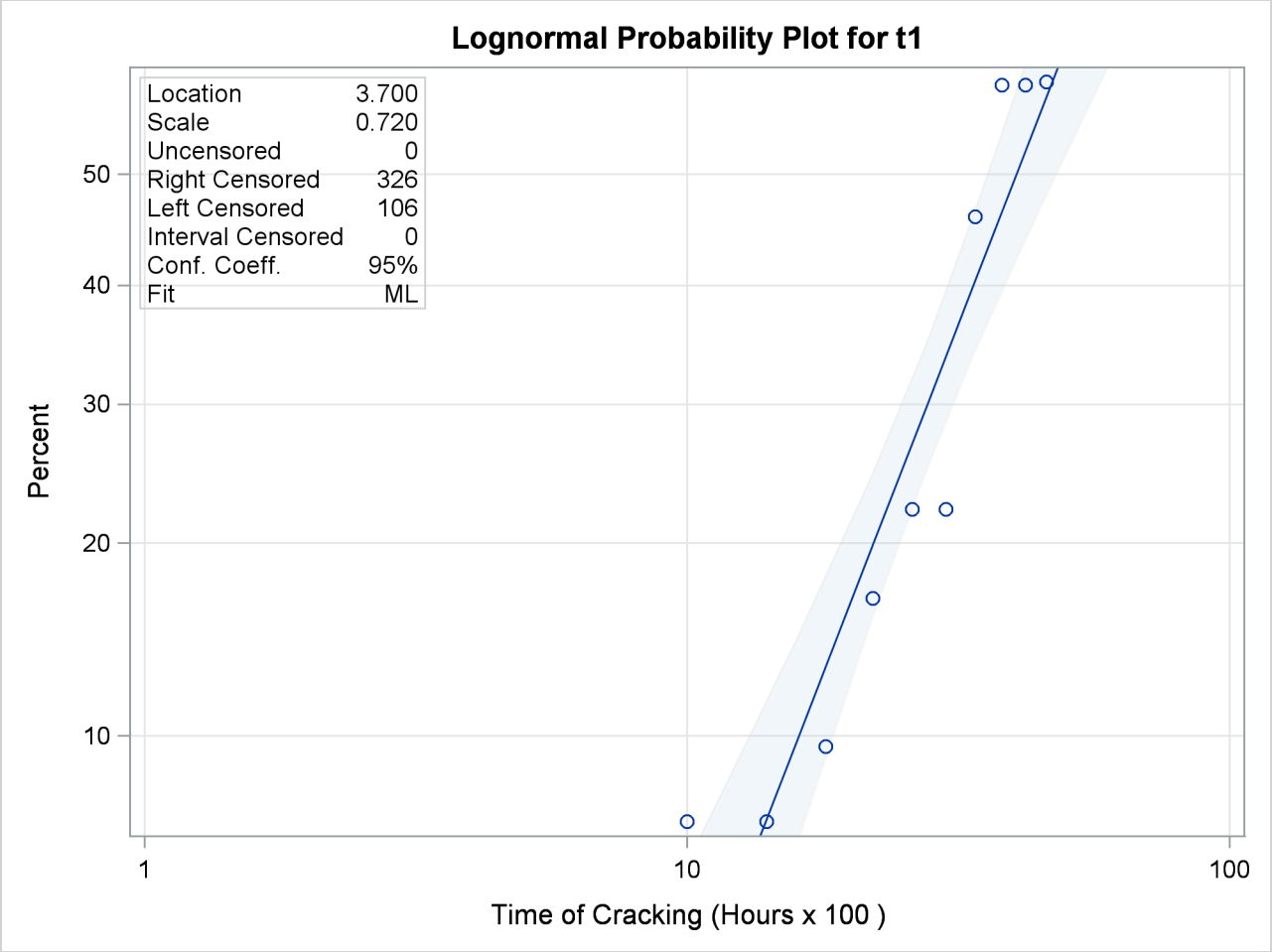


Figure 18.16 Partial Listing of the Tabular Output for the Turbine Wheel Data

The RELIABILITY Procedure

Model Information		
Input Data Set	WORK.TURBINE	
Analysis Variable	t1	Time of Cracking (Hours x 100)
Analysis Variable	t2	
Frequency Variable	f	
Distribution	Lognormal (Base e)	
Estimation Method	Maximum Likelihood	
Confidence Coefficient	95%	
Observations Used	21	

Figure 18.16 *continued*

Cumulative Probability Estimates

Lower Lifetime	Upper Lifetime	Cumulative Probability	Pointwise 95% Confidence Limits		Standard Error
			Lower	Upper	
.	4	0.0000	0.0000	0.0000	0.0000
10	10	0.0698	0.0264	0.1720	0.0337
14	14	0.0698	0.0177	0.2384	0.0473
18	18	0.0959	0.0464	0.1878	0.0345
22	22	0.1667	0.0711	0.3432	0.0680
26	26	0.2222	0.1195	0.3757	0.0657
30	30	0.2222	0.1203	0.3738	0.0650
34	34	0.4615	0.2236	0.7184	0.1383
38	38	0.5809	0.4085	0.7356	0.0865
42	42	0.5809	0.4280	0.7198	0.0766
46	46	0.5836	0.4195	0.7311	0.0822

Algorithm converged.

Summary of Fit

Observations Used	21
Uncensored Values	0
Right Censored Values	326
Left Censored Values	106
Maximum Loglikelihood	-190.7315

Lognormal Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits	
			Lower	Upper
Location	3.6999	0.0708	3.5611	3.8387
Scale	0.7199	0.0887	0.5655	0.9165

Other Lognormal Distribution Parameters

Parameter	Value
Mean	52.4062
Mode	24.0870
Median	40.4436
Standard Deviation	43.1855

Regression Modeling

This example is an illustration of a Weibull regression model that uses a load accelerated life test of rolling bearings, with data provided by Nelson (1990, p. 305). Bearings are tested at four different loads, and lifetimes in 10^6 of revolutions are measured. The data are shown in Table 18.3. An outlier identified by Nelson (1990) is omitted.

Table 18.3 Bearing Lifetime Data

Load	**Life (10^6 Revolutions)**									
0.87	1.67	2.2	2.51	3.00	3.90	4.70	7.53	14.7	27.76	37.4
0.99	0.80	1.0	1.37	2.25	2.95	3.70	6.07	6.65	7.05	7.37
1.09	0.18	0.2	0.24	0.26	0.32	0.32	0.42	0.44	0.88	
1.18	0.073	0.098	0.117	0.135	0.175	0.262	0.270	0.350	0.386	0.456

These data are modeled with a Weibull regression model in which the independent variable is the logarithm of the load. The model is

$$\mu_i = \beta_0 + \beta_1 x_i$$

where μ_i is the location parameter of the extreme value distribution and

$$x_i = \log(\text{load})$$

for the ith bearing. The following statements create a SAS data set containing the loads, log loads, and bearing lifetimes:

```
data bearing;
   input load Life @@;
   lload = log(load);
   datalines;
 .87 1.67    .87 2.2    .87 2.51    .87 3.0    .87 3.9
 .87 4.7     .87 7.53   .87 14.7    .87 27.76  .87 37.4
 .99  .8     .99 1.0    .99 1.37    .99 2.25   .99 2.95
 .99 3.7     .99 6.07   .99 6.65    .99 7.05   .99 7.37
1.09  .18   1.09  .2   1.09  .24   1.09  .26  1.09  .32
1.09  .32   1.09  .42  1.09  .44   1.09  .88  1.18  .073
1.18  .098  1.18  .117 1.18  .135  1.18  .175 1.18  .262
1.18  .270  1.18  .350 1.18  .386  1.18  .456
;
```

Figure 18.17 shows a listing of the bearing data.

Figure 18.17 Listing of the Bearing Data

Obs	load	Life	lload
1	0.87	1.670	-0.13926
2	0.87	2.200	-0.13926
3	0.87	2.510	-0.13926
4	0.87	3.000	-0.13926
5	0.87	3.900	-0.13926
6	0.87	4.700	-0.13926
7	0.87	7.530	-0.13926
8	0.87	14.700	-0.13926
9	0.87	27.760	-0.13926
10	0.87	37.400	-0.13926
11	0.99	0.800	-0.01005
12	0.99	1.000	-0.01005
13	0.99	1.370	-0.01005
14	0.99	2.250	-0.01005
15	0.99	2.950	-0.01005
16	0.99	3.700	-0.01005
17	0.99	6.070	-0.01005
18	0.99	6.650	-0.01005
19	0.99	7.050	-0.01005
20	0.99	7.370	-0.01005
21	1.09	0.180	0.08618
22	1.09	0.200	0.08618
23	1.09	0.240	0.08618
24	1.09	0.260	0.08618
25	1.09	0.320	0.08618
26	1.09	0.320	0.08618
27	1.09	0.420	0.08618
28	1.09	0.440	0.08618
29	1.09	0.880	0.08618
30	1.18	0.073	0.16551
31	1.18	0.098	0.16551
32	1.18	0.117	0.16551
33	1.18	0.135	0.16551
34	1.18	0.175	0.16551
35	1.18	0.262	0.16551
36	1.18	0.270	0.16551
37	1.18	0.350	0.16551
38	1.18	0.386	0.16551
39	1.18	0.456	0.16551

The following statements fit the regression model by maximum likelihood that uses the Weibull distribution:

```
ods output modobstats = Residual;
proc reliability data=bearing;
   distribution Weibull;
   model life = lload /  covb
                         corrb
                         obstats
                         ;
run;
```

The PROC RELIABILITY statement invokes the procedure and identifies BEARING as the input data set. The DISTRIBUTION statement specifies the Weibull distribution for model fitting. The MODEL statement specifies the regression model, identifying Life as the variable that provides the response values (the lifetimes) and Lload as the independent variable (the log loads). The MODEL statement option COVB requests the regression parameter covariance matrix, and the CORRB option requests the correlation matrix. The option OBSTATS requests a table that contains residuals, predicted values, and other statistics. The ODS OUTPUT statement creates a SAS data set named RESIDUAL that contains the table created by the OBSTATS option.

Figure 18.18 shows the tabular output produced by the RELIABILITY procedure. The "Weibull Parameter Estimates" table contains parameter estimates, their standard errors, and 95% confidence intervals. In this table, INTERCEPT corresponds to β_0, LLOAD corresponds to β_1, and SHAPE corresponds to the Weibull shape parameter. Figure 18.19 shows a listing of the output data set RESIDUAL.

Figure 18.18 Analysis Results for the Bearing Data

The RELIABILITY Procedure

Model Information	
Input Data Set	WORK.BEARING
Analysis Variable	Life
Distribution	Weibull

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	lload
Prm3	EV Scale

Algorithm converged.

Summary of Fit	
Observations Used	39
Uncensored Values	39
Maximum Loglikelihood	-51.77737

Figure 18.18 *continued*

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	0.8323	0.1410	0.5560	1.1086
lload	-13.8529	1.2333	-16.2703	-11.4356
EV Scale	0.8043	0.0999	0.6304	1.0260
Weibull Shape	1.2434	0.1545	0.9746	1.5862

Estimated Covariance Matrix
Weibull Parameters

	Prm1	Prm2	Prm3
Prm1	0.01987	-0.04374	-0.00492
Prm2	-0.04374	1.52113	0.01578
Prm3	-0.00492	0.01578	0.00999

Estimated Correlation Matrix
Weibull Parameters

	Prm1	Prm2	Prm3
Prm1	1.0000	-0.2516	-0.3491
Prm2	-0.2516	1.0000	0.1281
Prm3	-0.3491	0.1281	1.0000

Figure 18.19 Listing of Data Set Residual

Obs	Life	lload	Xbeta	Surv	Resid	SRESID	Aresid
1	1.67	-0.139262	2.7614742	0.9407681	-2.248651	-2.795921	-2.795921
2	2.2	-0.139262	2.7614742	0.9175782	-1.973017	-2.453205	-2.453205
3	2.51	-0.139262	2.7614742	0.9036277	-1.841191	-2.289296	-2.289296
4	3	-0.139262	2.7614742	0.8811799	-1.662862	-2.067565	-2.067565
5	3.9	-0.139262	2.7614742	0.8392186	-1.400498	-1.741347	-1.741347
6	4.7	-0.139262	2.7614742	0.8016738	-1.213912	-1.50935	-1.50935
7	7.53	-0.139262	2.7614742	0.6721971	-0.742579	-0.923306	-0.923306
8	14.7	-0.139262	2.7614742	0.4015113	-0.073627	-0.091546	-0.091546
9	27.76	-0.139262	2.7614742	0.1337746	0.562122	0.6989298	0.6989298
10	37.4	-0.139262	2.7614742	0.0542547	0.8601965	1.069549	1.069549
11	0.8	-0.01005	0.971511	0.7973909	-1.194655	-1.485407	-1.485407
12	1	-0.01005	0.971511	0.741702	-0.971511	-1.207955	-1.207955
13	1.37	-0.01005	0.971511	0.6427726	-0.6567	-0.816526	-0.816526
14	2.25	-0.01005	0.971511	0.4408692	-0.160581	-0.199663	-0.199663
15	2.95	-0.01005	0.971511	0.3175927	0.1102941	0.1371372	0.1371372
16	3.7	-0.01005	0.971511	0.2186832	0.3368218	0.4187966	0.4187966
17	6.07	-0.01005	0.971511	0.0600164	0.8318476	1.0343005	1.0343005
18	6.65	-0.01005	0.971511	0.0428027	0.9231058	1.147769	1.147769
19	7.05	-0.01005	0.971511	0.0337583	0.9815166	1.2203956	1.2203956
20	7.37	-0.01005	0.971511	0.0278531	1.0259067	1.2755892	1.2755892
21	0.18	0.0861777	-0.361531	0.8303684	-1.353268	-1.682623	-1.682623
22	0.2	0.0861777	-0.361531	0.809042	-1.247907	-1.55162	-1.55162
23	0.24	0.0861777	-0.361531	0.7665749	-1.065586	-1.324925	-1.324925
24	0.26	0.0861777	-0.361531	0.7455451	-0.985543	-1.225402	-1.225402
25	0.32	0.0861777	-0.361531	0.6837688	-0.777904	-0.967228	-0.967228
26	0.32	0.0861777	-0.361531	0.6837688	-0.777904	-0.967228	-0.967228
27	0.42	0.0861777	-0.361531	0.5868036	-0.50597	-0.629112	-0.629112
28	0.44	0.0861777	-0.361531	0.5684693	-0.45945	-0.57127	-0.57127
29	0.88	0.0861777	-0.361531	0.2625812	0.2336973	0.290574	0.290574
30	0.073	0.1655144	-1.460578	0.7887184	-1.156718	-1.438237	-1.438237
31	0.098	0.1655144	-1.460578	0.7101313	-0.86221	-1.072052	-1.072052
32	0.117	0.1655144	-1.460578	0.6526714	-0.685003	-0.851717	-0.851717
33	0.135	0.1655144	-1.460578	0.6006317	-0.541902	-0.673789	-0.673789
34	0.175	0.1655144	-1.460578	0.4946523	-0.282391	-0.351119	-0.351119
35	0.262	0.1655144	-1.460578	0.3126729	0.1211675	0.1506569	0.1506569
36	0.27	0.1655144	-1.460578	0.2991233	0.1512449	0.1880546	0.1880546
37	0.35	0.1655144	-1.460578	0.1889073	0.4107561	0.5107249	0.5107249
38	0.386	0.1655144	-1.460578	0.1522503	0.5086604	0.6324568	0.6324568
39	0.456	0.1655144	-1.460578	0.0987061	0.6753158	0.8396724	0.8396724

The value of the lifetime Life and the log load Lload are included in this data set, as well as statistics computed from the fitted model. The variable Xbeta is the value of the linear predictor

$$\mathbf{x}'\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \text{Lload}\hat{\beta}_1$$

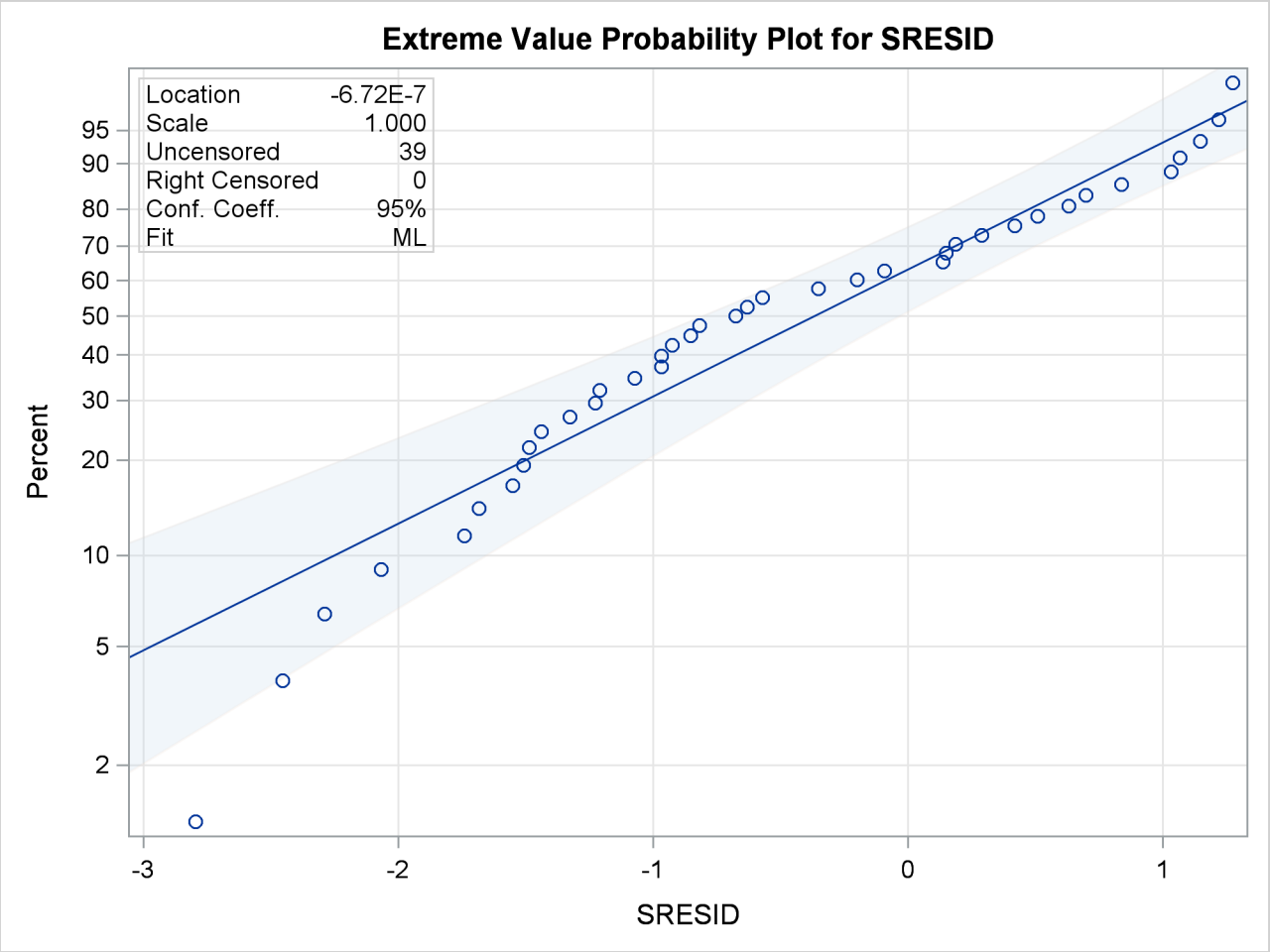
for each observation. The variable Surv contains the value of the reliability function, the variable Sresid contains the standardized residual, and the variable Aresid contains a residual adjusted for right-censored observations. Since there are no censored values in these data, Sresid is equal to Aresid for all the bearings. See Table 18.32 and Table 18.33 for other statistics that are available in the OBSTATS table and data set. See the section "Regression Model Statistics Computed for Each Observation for Lifetime Data" on page 1370 for a description of the residuals and other statistics.

If the fitted regression model is adequate, the standardized residuals have a standard extreme value distribution. You can check the residuals by using the RELIABILITY procedure and the RESIDUAL data set to create an extreme value probability plot of the residuals.

The following statements create the plot in Figure 18.20:

```
proc reliability data=residual;
   distribution ev;
   probplot sresid;
run;
```

Figure 18.20 Extreme Value Probability Plot for the Standardized Residuals



Although the estimated location is near zero and the estimated scale is near one, the plot reveals systematic curvature, indicating that the Weibull regression model might be inadequate.

Regression Model with Nonconstant Scale

Nelson (1990, p. 272) and Meeker and Escobar (1998, p. 439) analyzed data from a strain-controlled fatigue test on 26 specimens of a type of superalloy. The following SAS statements create a SAS data set containing for each specimen the level of pseudo-stress (Pstress), the number of cycles (in thousands) (Kcycles) until failure or removal from the test, and a variable to indicate whether a specimen failed (F) or was right censored (C) (Status):

```
data alloy;
   input pstress kCycles status$ @@;
   cen = ( status = 'C' );
   datalines;
80.3  211.629  F    99.8   43.331  F
80.6  200.027  F   100.1   12.076  F
80.8   57.923  C   100.5   13.181  F
84.3  155.000  F   113.0   18.067  F
85.2   13.949  F   114.8   21.300  F
85.6  112.968  C   116.4   15.616  F
85.8  152.680  F   118.0   13.030  F
86.4  156.725  F   118.4    8.489  F
86.7  138.114  C   118.6   12.434  F
87.2   56.723  F   120.4    9.750  F
87.3  121.075  F   142.5   11.865  F
89.7  122.372  C   144.5    6.705  F
91.3  112.002  F   145.9    5.733  F
;
```

The following statements fit a Weibull regression model with the number of cycles to failure as the response variable:

```
ods output ModObstats = Resids;
proc reliability data = alloy;
   distribution Weibull;
   model kcycles*cen(1) = pstress pstress*pstress / Relation = Pow Obstats;
   logscale pstress;
   rplot kcycles*cen(1) = pstress /  fit=regression
                                     relation = pow
                                     plotfit  10 50 90
                                     slower=60 supper=160
                                     lupper=500;

   label pstress = "Pseudo-Stress";
   label kcycles = "Thousands of Cycles";
run;
```

The data set RESIDS contains standardized residuals created with the ODS OUTPUT statement. The MODEL statement specifies a model quadratic in the log of pseudo-stress for the extreme value location parameter. The quadratic model in pseudo-stress PSTRESS is specified in the MODEL statement, and the RELATION=POW option specifies that the log transformation be applied to Pstress in the MODEL statement and the LOGSCALE statement. The LOGSCALE statement specifies the log of the scale parameter as a linear function of the log of Pstress. The RPLOT statement specifies a plot of the data and the fitted regression model versus the variable Pstress. The FIT=REGRESSION option specifies plotting the regression model

fitted with the preceding MODEL statement. The RELATION=POW option specifies a log stress axis. The PLOTFIT option specifies plotting the 10th, 50th, and 90th percentiles of the regression model at each stress level. The SLOWER, SUPPER, and LUPPER options control limits on the stress and lifetime axes.

Figure 18.21 displays the parameter estimates from the fitted regression model. Parameter estimates for both the model for the location parameter and the scale parameter models are shown. Standard errors and confidence limits for all parameter estimates are included.

Figure 18.21 Parameter Estimates for Fitted Regression Model

The RELIABILITY Procedure

Weibull Parameter Estimates

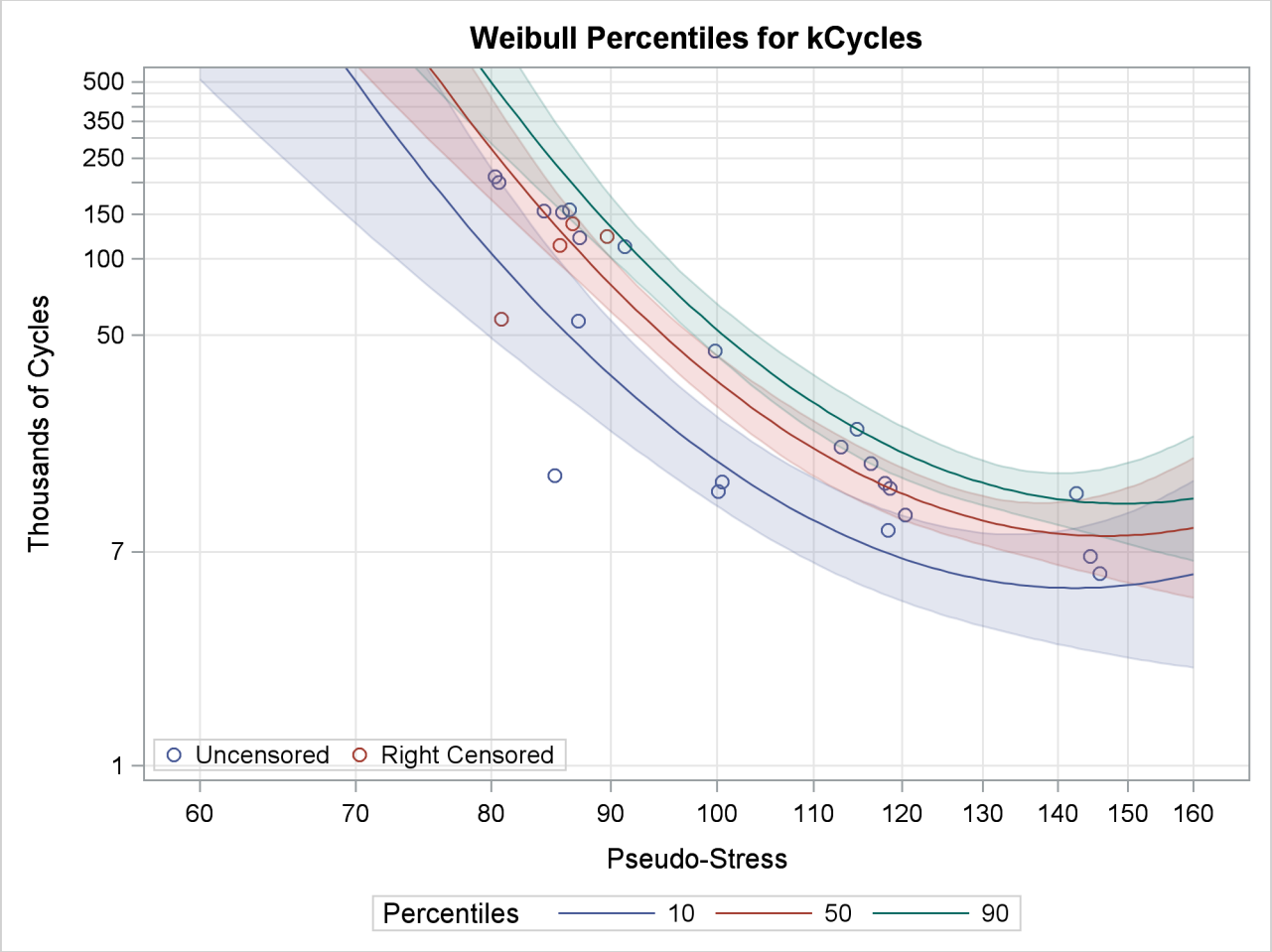
Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	243.1680	58.1777	129.1418	357.1943
pstress	-96.5240	24.7558	-145.0445	-48.0035
pstress*pstress	9.6653	2.6299	4.5107	14.8198

Log-Scale Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	4.4666	4.1745	-3.7152	12.6484
pstress	-1.1757	0.8950	-2.9299	0.5784

Figure 18.22 displays the plot of the data and fitted regression model.

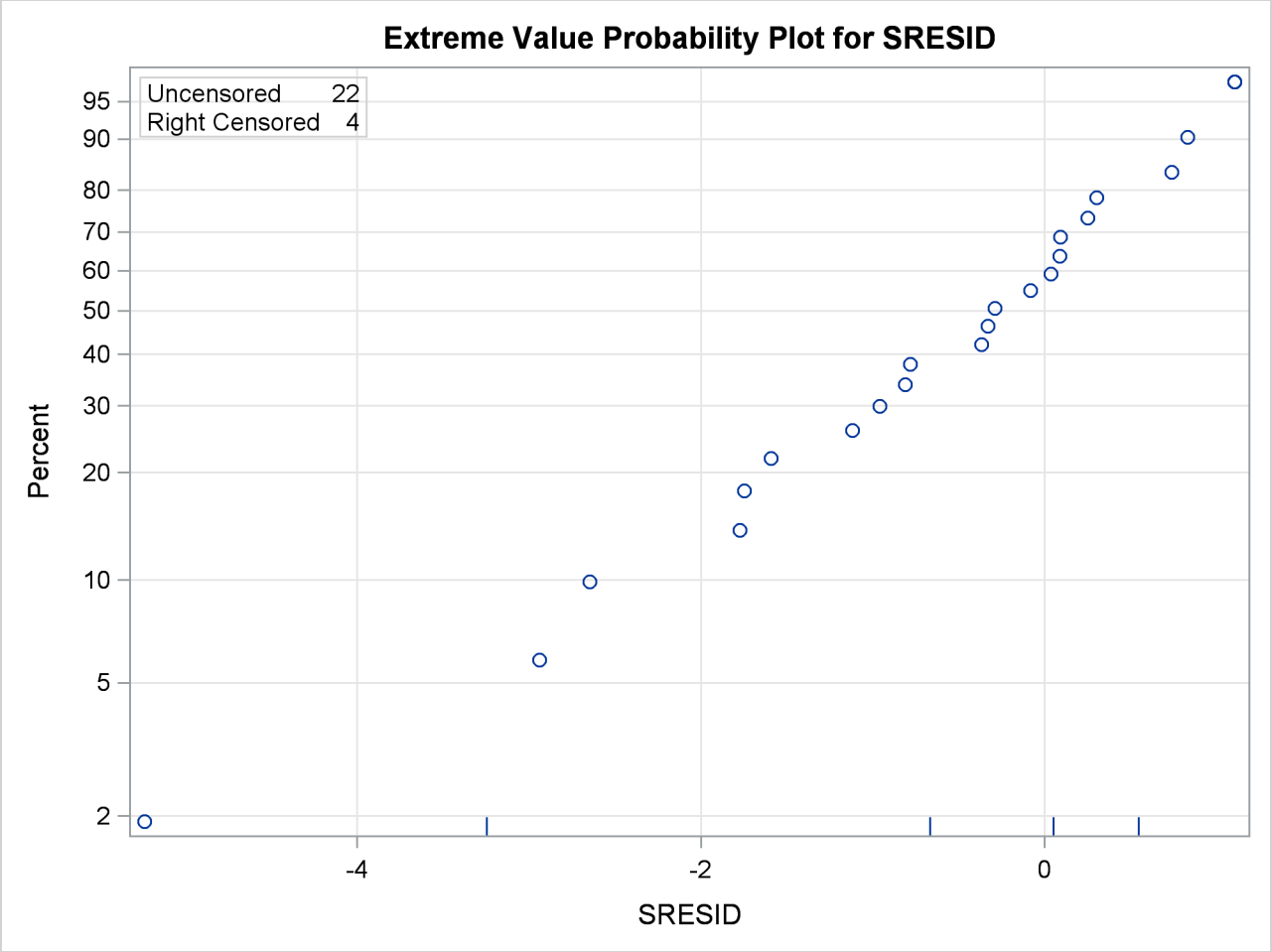
Figure 18.22 Superalloy Fatigue Data with Fitted Regression Model



The following SAS statements create an extreme values probability plot of standardized residuals from the regression model shown in Figure 18.23:

```
proc reliability data = Resids;
   distribution ev;
   pplot sresid*cen(1) / nofit;
run;
```

Figure 18.23 Residuals for Superalloy Fatigue Data Regression Model



Regression Model with Two Independent Variables

Meeker and Escobar (1998, p. 447) analyzed data from an accelerated test on the lifetimes of glass capacitors as a function of operating voltage and temperature. The following SAS statements create a SAS data set containing the data. There are four lifetimes for each of eight combinations and four censored observations after the fourth failure for each combination:

```
data glass;
   input Temp Voltage @;
   do i = 1 to 4;
      cen = 0;
      input Hours @; output;
   end;
   do i = 1 to 4;
      cen = 1;
      output;
   end;
   datalines;
```

```
170 200  439  904  1092  1105
170 250  572  690   904  1090
170 300  315  315   439   628
170 350  258  258   347   588
180 200  959 1065  1065  1087
180 250  216  315   455   473
180 300  241  315   332   380
180 350  241  241   435   455
;
```

The following statements analyze the capacitor data. The MODEL statement fits a regression model with Temp and Voltage as independent variables. Parameter estimates from the fitted regression model are shown in Figure 18.24. An interaction term between Temp and Voltage is included. The PPLOT statement creates a Weibull probability plot shown in Figure 18.25 with all temperature-voltage combinations overlaid on the same plot. The regression model fit is also plotted. The RPLOT statement creates the plot shown in Figure 18.26 of the data and Weibull distribution percentiles from the regression model as a function of voltage for values of temperature of 150, 170, and 180:

```
proc reliability data = glass;
   distribution Weibull;
   model Hours*cen(1) = temp voltage temp * voltage;
   pplot Hours*cen(1) = ( temp voltage ) / fit = model
                                           overlay
                                           noconf
                                           lupper = 2000;
run;

proc reliability data = glass;
   distribution Weibull;
   model Hours*cen(1) = temp voltage temp * voltage;
   rplot Hours*cen(1) = voltage / fit = regression(temp = 150, 170, 180)
                                  plotfit;
run;
```

Figure 18.24 Parameter Estimates for Fitted Regression Model

The RELIABILITY Procedure

Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	9.4135	10.5402	-11.2449	30.0719
Temp	-0.0062	0.0598	-0.1235	0.1110
Voltage	0.0086	0.0374	-0.0648	0.0820
Temp*Voltage	-0.0001	0.0002	-0.0005	0.0003
EV Scale	0.3624	0.0553	0.2687	0.4887
Weibull Shape	2.7593	0.4210	2.0461	3.7209

Figure 18.25 Probability Plot for Glass Capacitor Regression Model

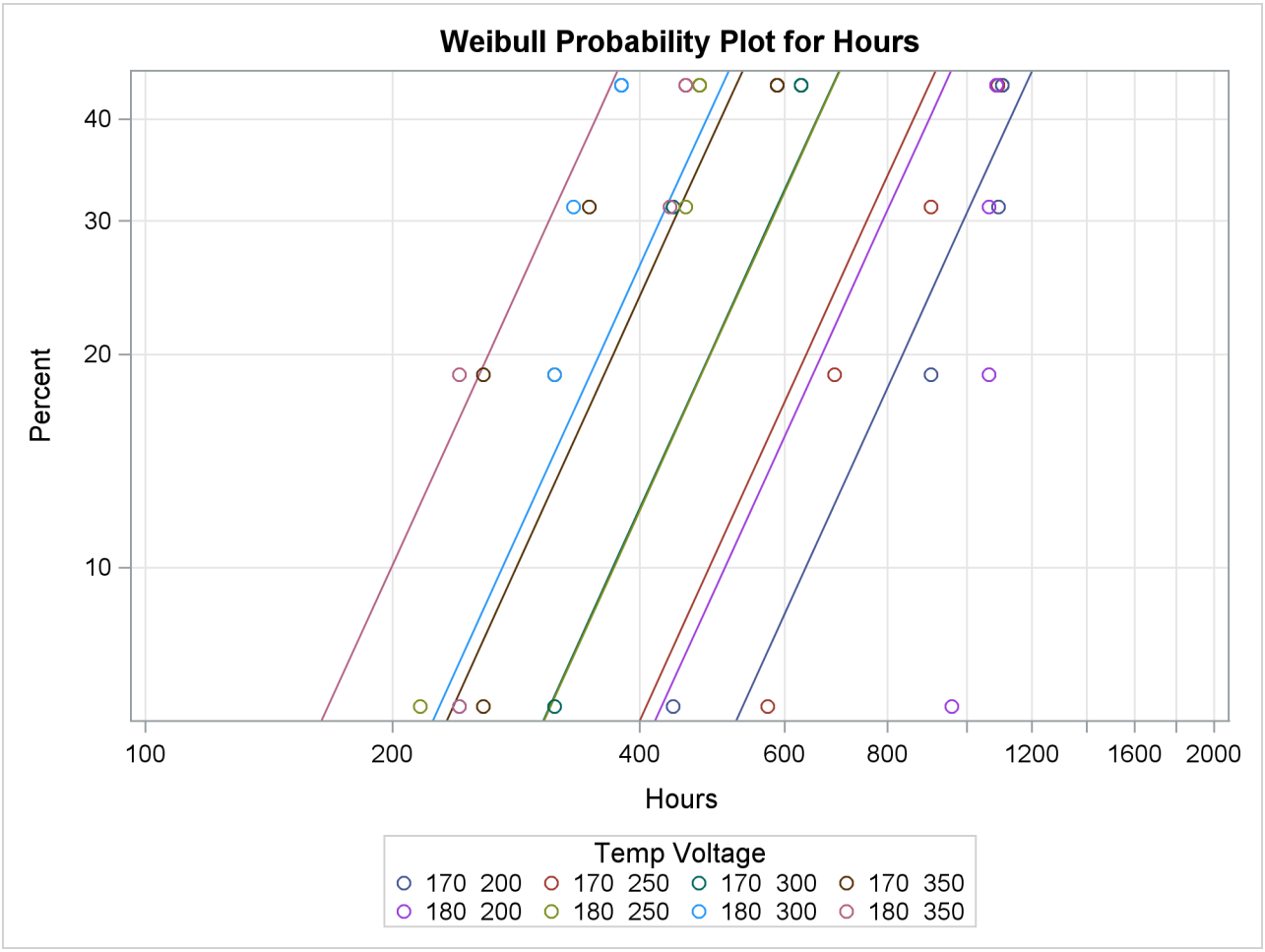
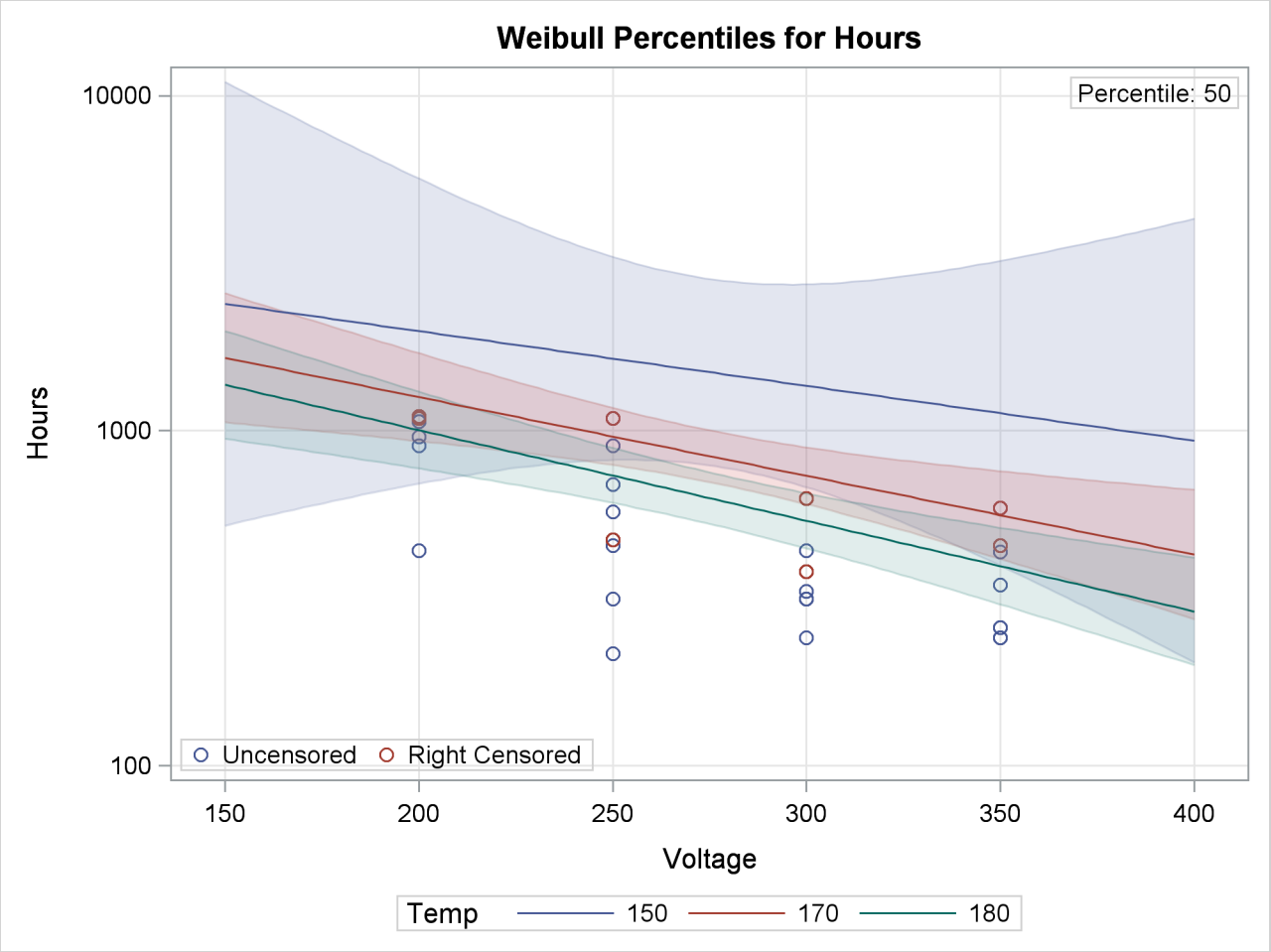


Figure 18.26 Plot of Data and Fitted Weibull Percentiles for Glass Capacitor Regression Model



Weibull Probability Plot for Two Combined Failure Modes

Doganaksoy, Hahn, and Meeker (2002) analyzed failure data for the dielectric insulation of generator armature bars. A sample of 58 segments of bars were subjected to a high voltage stress test. Based on examination of the sample after the test, failures were attributed to one of two modes:

- Mode D (degradation failure): degradation of the organic material. Such failures usually occur later in life.
- Mode E (early failure): insulation defects due to a processing problem. These failures tend to occur early in life.

The following SAS statements create a SAS data set that contains the failure data:

```
data Voltage;
   input Hours Mode$ @@;
   if Mode = 'Cen' then Status = 1;
   else Status = 0;
   datalines;
2   E   3   E   5   E   8   E   13  Cen 21  E
28  E   31  E   31  Cen 52  Cen 53  Cen 64  E
67  Cen 69  E   76  E   78  Cen 104 E   113 Cen
119 E   135 Cen 144 E   157 Cen 160 E   168 D
179 Cen 191 D   203 D   211 D   221 E   226 D
236 E   241 Cen 257 Cen 261 D   264 D   278 D
282 E   284 D   286 D   298 D   303 E   314 D
317 D   318 D   320 D   327 D   328 D   328 D
348 D   348 Cen 350 D   360 D   369 D   377 D
387 D   392 D   412 D   446 D
;
```

The variable Hours represents the number of hours until a failure, or the number of hours on test if the sample unit did not fail. The variable Mode represents the failure mode: D for degradation failure, E for early failures, or Cen if the unit did not fail (i.e., is right-censored). The computed variable Status is a numeric indicator for censored observations.

The following statements fit a Weibull distribution to the individual failure modes (D and E), and compute the failure distribution with both modes acting:

```
proc reliability data=Voltage;
   distribution Weibull;
   pplot Hours*Status(1) / pref(intersect) = 10
                           preflabel = ('10th Percentile')
                           survtime = 100 200 300 400 500 1000
                           lupper = 500;
   fmode combine = Mode( 'D' 'E' );
run;
```

Figure 18.27 contains estimates of the combined failure mode survival function at the times specified with the SURVTIME= option in the PPLOT statement.

Figure 18.27 Survival Function Estimates for Combined Failure Modes

The RELIABILITY Procedure

Combined Failure Modes
Weibull Distribution Function Estimates
With 95% Asymptotic Normal Confidence Limits

X	Pr(<X)	Lower	Upper	Pr(>X)	Lower	Upper
100.00	0.1898	0.1172	0.2926	0.8102	0.7074	0.8828
200.00	0.3115	0.2139	0.4292	0.6885	0.5708	0.7861
300.00	0.5866	0.4621	0.7010	0.4134	0.2990	0.5379
400.00	0.9405	0.8476	0.9782	0.0595	0.0218	0.1524
500.00	0.9998	0.9711	1.0000	0.0002	0.0000	0.0289
1000.00	1.0000	0.0000	1.0000	0.0000	0.0000	1.0000

Figure 18.28 shows Weibull parameter estimates for the two individual failure modes.

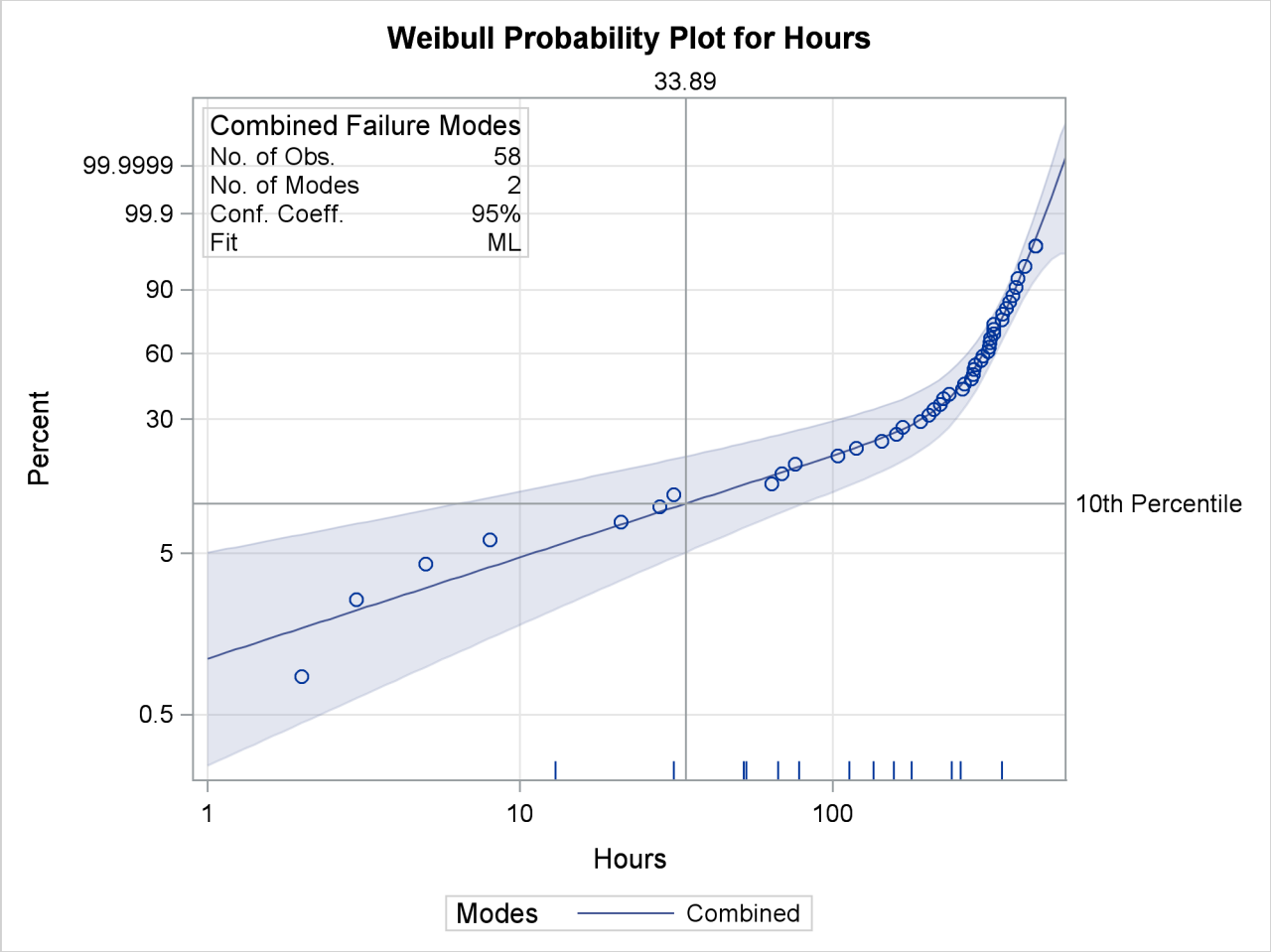
Figure 18.28 Parameter Estimates for Individual Failure Modes

Individual Failure Mode
Weibull Parameter Estimates

			Asymptotic Normal 95% Confidence Limits		
Parameter	Estimate	Standard Error	Lower	Upper	Mode
EV Location	5.8415	0.0350	5.7730	5.9100	D
EV Scale	0.1785	0.0254	0.1350	0.2360	D
Weibull Scale	344.2966	12.0394	321.4903	368.7208	D
Weibull Shape	5.6020	0.7985	4.2365	7.4076	D
EV Location	7.0649	0.5109	6.0637	8.0662	E
EV Scale	1.5739	0.3415	1.0287	2.4080	E
Weibull Scale	1170.1832	597.7903	429.9480	3184.8703	E
Weibull Shape	0.6354	0.1379	0.4153	0.9721	E

Figure 18.29 is a Weibull probability plot of the failure probability distribution with the two failure modes combined, along with approximate pointwise 95% confidence limits. A reference line at the 10% point on the probability axis intersecting the distribution curve shows the tenth percentile of lifetimes when both modes act to be about 34 hours.

Figure 18.29 Weibull Plot for Failure Modes D and E

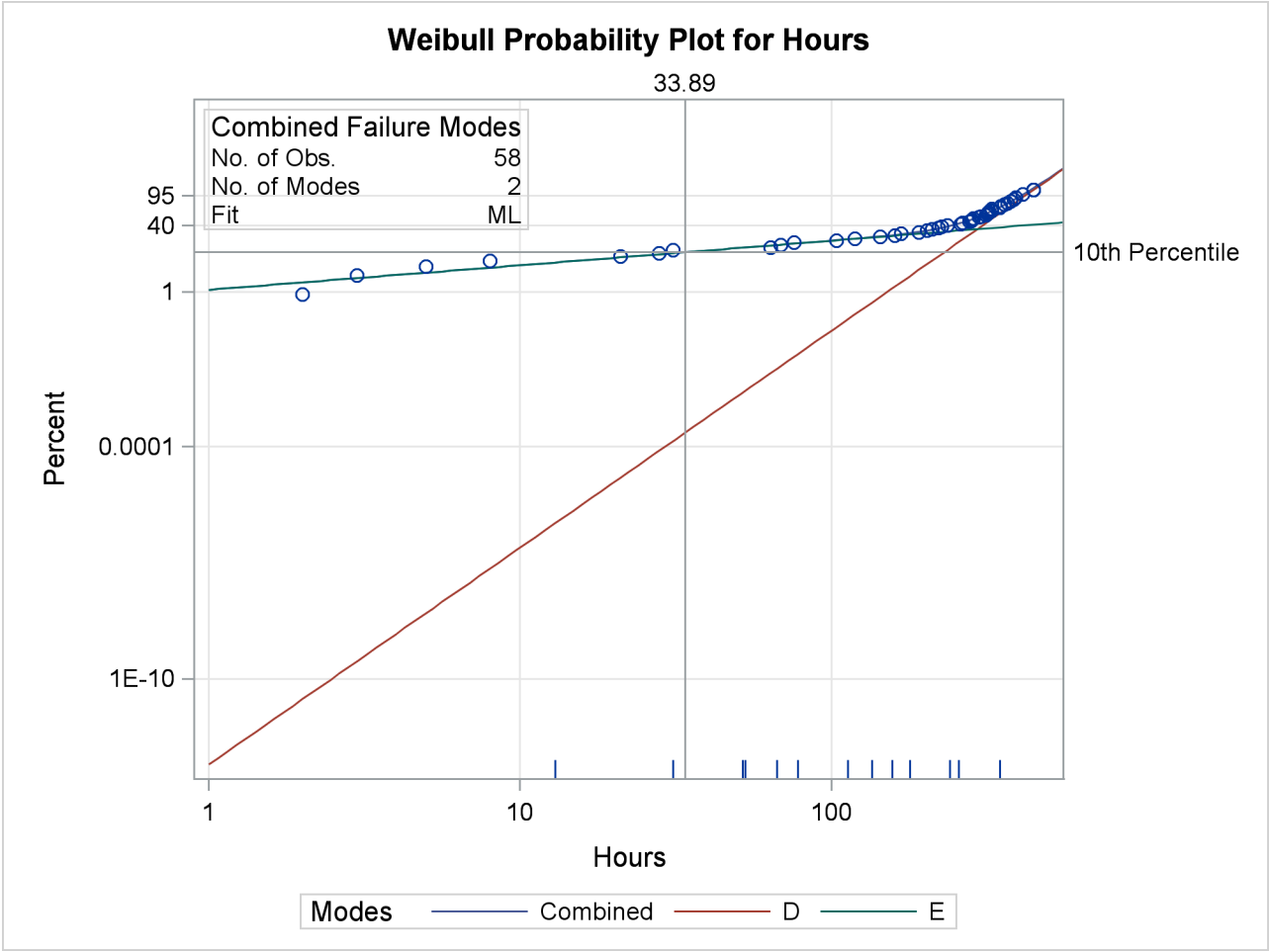


The following SAS statements create the Weibull probability plot in Figure 18.30:

```
proc reliability data=Voltage;
   distribution Weibull;
   pplot Hours*Status(1) / pref(intersect) = 10
                           preflabel = ('10th Percentile')
                           survtime = 100 200 300 400 500 1000
                           noconf
                           lupper = 500;
   fmode combine = Mode( 'D' 'E' ) / plotmodes;
run;
```

The PLOTMODES option in the FMODE statement cause the Weibull fits for the individual failure modes to be included on the probability plot. The combined failure mode curve is almost the same as the fit curve for mode E for lifetimes less than 100 hours, and slightly greater than the fit curve for mode D for lifetimes greater than 100 hours.

Figure 18.30 Weibull Plot for Failure Modes D and E with Individual Modes



Analysis of Recurrence Data on Repairs

This example illustrates analysis of recurrence data from repairable systems. Repair data analysis differs from life data analysis, where units fail only once. As a repairable system ages, it accumulates repairs and costs of repairs. The RELIABILITY procedure provides a nonparametric estimate and plot of the *mean cumulative function* (MCF) for the number or cost of repairs for a population of repairable systems.

The nonparametric estimate of the MCF, the variance of the MCF estimate, and confidence limits for the MCF estimate are based on the work of Nelson (1995). The MCF, also written as $M(t)$, is defined by Nelson (1995) to be the *population mean* of the distribution of the cumulative number or cost of repairs at age t. The method does not assume any underlying structure for the repair process.

The SAS statements that follow create the listing of the SAS data set VALVE shown in Figure 18.31, which contains repair histories of 41 diesel engines in a fleet (Nelson 1995). The valve seats in these engines wear out and must be replaced. The variable Id is a unique identifier for individual engines. The variable Days provides the engine age in days. The value of the variable Value is 1 if the age is a valve seat replacement age or -1 if the age is the end of history, or censoring age, for the engine.

```
data valve;
   input id Days value @@;
   label Days = 'Time of Replacement (Days)';
   datalines;
251 761 -1      252 759 -1      327  98  1      327 667 -1
328 326  1      328 653  1      328 653  1      328 667 -1
329 665 -1      330  84  1      330 667 -1      331  87  1
331 663 -1      389 646  1      389 653 -1      390  92  1
390 653 -1      391 651 -1      392 258  1      392 328  1
392 377  1      392 621  1      392 650 -1      393  61  1
393 539  1      393 648 -1      394 254  1      394 276  1
394 298  1      394 640  1      394 644 -1      395  76  1
395 538  1      395 642 -1      396 635  1      396 641 -1
397 349  1      397 404  1      397 561  1      397 649 -1
398 631 -1      399 596 -1      400 120  1      400 479  1
400 614 -1      401 323  1      401 449  1      401 582 -1
402 139  1      402 139  1      402 589 -1      403 593 -1
404 573  1      404 589 -1      405 165  1      405 408  1
405 604  1      405 606 -1      406 249  1      406 594 -1
407 344  1      407 497  1      407 613 -1      408 265  1
408 586  1      408 595 -1      409 166  1      409 206  1
409 348  1      409 389 -1      410 601 -1      411 410  1
411 581  1      411 601 -1      412 611 -1      413 608 -1
414 587 -1      415 367  1      415 603 -1      416 202  1
416 563  1      416 570  1      416 585 -1      417 587 -1
418 578 -1      419 578 -1      420 586 -1      421 585 -1
422 582 -1
;
```

Figure 18.31 Partial Listing of the Valve Seat Data

Obs	id	Days	value
1	251	761	-1
2	252	759	-1
3	327	98	1
4	327	667	-1
5	328	326	1
6	328	653	1
7	328	653	1
8	328	667	-1
9	329	665	-1
10	330	84	1
11	330	667	-1
12	331	87	1
13	331	663	-1
14	389	646	1
15	389	653	-1
16	390	92	1
17	390	653	-1
18	391	651	-1
19	392	258	1
20	392	328	1

The following statements produce the graphical displays in Figure 18.32 and Figure 18.33.

```
proc reliability;
   unitid id;
   mcfplot Days*value(-1) / nocenprint eventplot;
run;
```

The UNITID statement specifies that the variable Id uniquely identifies each system. The MCFPLOT statement requests a plot of the MCF estimates as a function of the age variable Days, and it specifies -1 as the value of the variable Value, which identifies the end of history for each engine (system). The option NOCENPRINT specifies that only failure times, and not censoring times, be printed in the tabular output. The option EVENTPLOT requests a horizontal plot of failure times for each system.

In Figure 18.32, the MCF estimates and confidence limits are plotted versus system age in days. The end-of-history ages are plotted in an area at the top of the plot. Except for the last few points, the plot is essentially a straight line, suggesting a constant replacement rate. Consequently, the prediction of future replacements of valve seats can be based on a fitted line in this case.

In Figure 18.33, a horizontal line for each system is drawn. Failures are marked by solid circles, and each line terminates at the censoring time for that system.

Figure 18.32 Mean Cumulative Function for the Number of Repairs

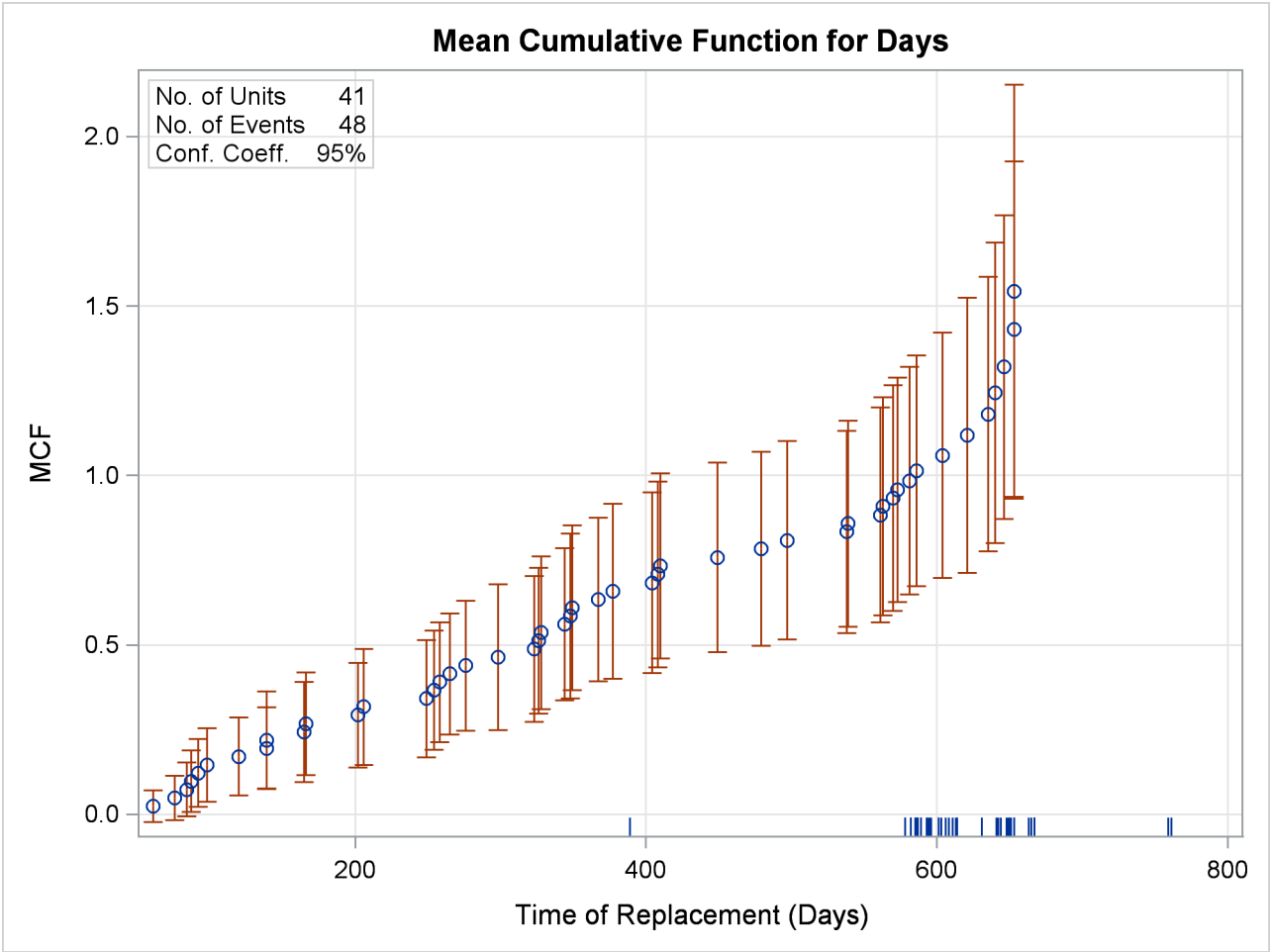
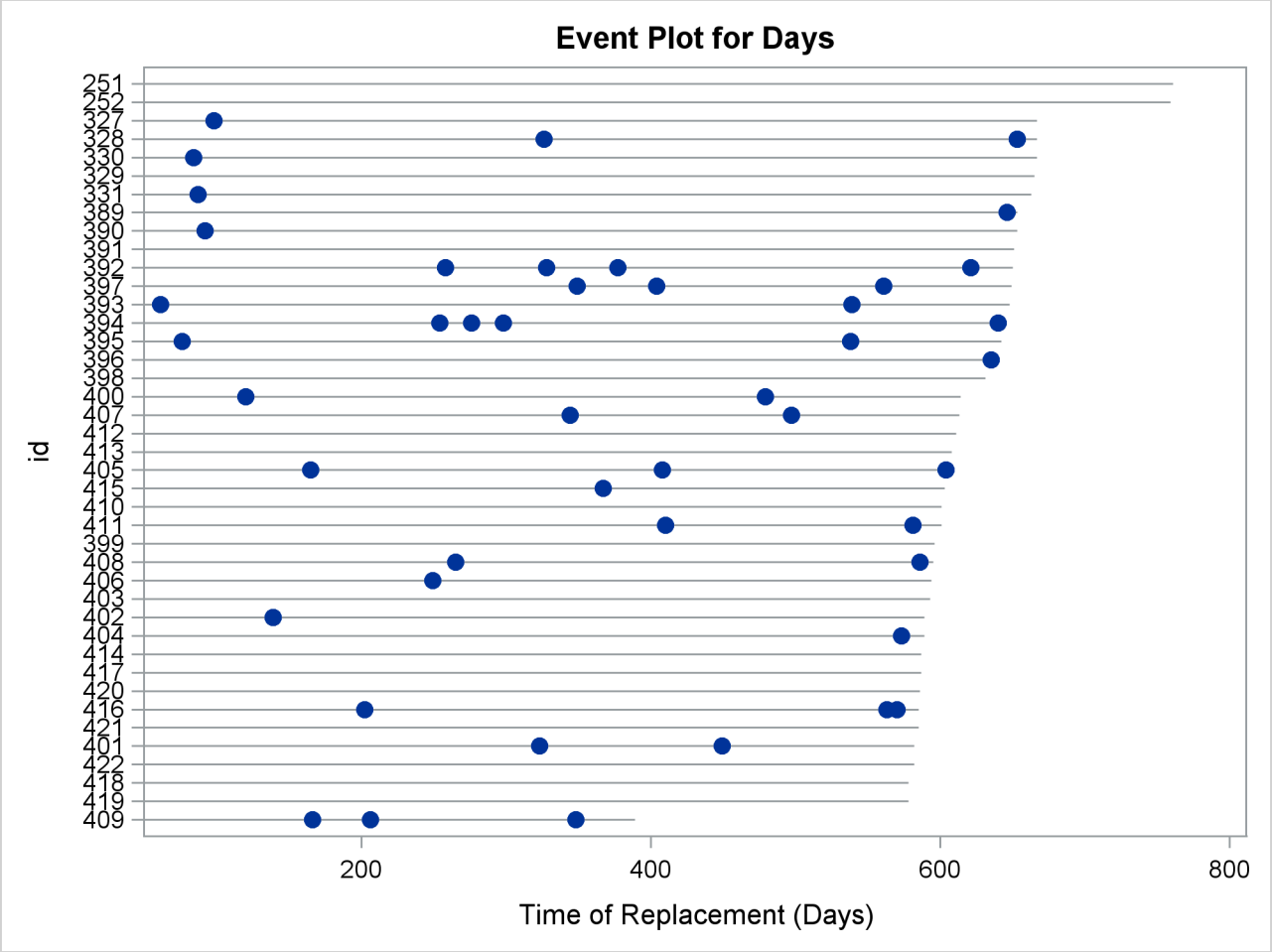


Figure 18.33 Recurrent Events Plot for the Valve Seat Data



A partial listing of the tabular output is shown in Figure 18.34 and Figure 18.35. It contains a summary of the repair data, estimates of the MCF, the Nelson (1995) standard errors, and confidence intervals for the MCF.

Figure 18.34 Partial Listing of the Output for the Valve Seat Data

Recurrence Data Summary	
Input Data Set	WORK.VALVE
Observations Used	89
Number of Units	41
Number of Events	48

Figure 18.35 Partial Listing of the Output for the Valve Seat Data

Recurrence Data Analysis

Age	Sample MCF	Standard Error	95% Confidence Limits Lower	Upper	Unit ID
61.00	0.024	0.024	-0.023	0.072	393
76.00	0.049	0.034	-0.017	0.115	395
84.00	0.073	0.041	-0.007	0.153	330
87.00	0.098	0.046	0.007	0.188	331
92.00	0.122	0.051	0.022	0.222	390
98.00	0.146	0.055	0.038	0.255	327
120.00	0.171	0.059	0.056	0.286	400
139.00	0.195	0.062	0.074	0.316	402
139.00	0.220	0.073	0.076	0.363	402
165.00	0.244	0.075	0.096	0.392	405
166.00	0.268	0.077	0.117	0.420	409
202.00	0.293	0.079	0.138	0.447	416
206.00	0.317	0.088	0.146	0.489	409
249.00	0.341	0.089	0.168	0.515	406
254.00	0.366	0.090	0.190	0.542	394
258.00	0.390	0.090	0.213	0.568	392
265.00	0.415	0.091	0.236	0.593	408
276.00	0.439	0.098	0.247	0.631	394
298.00	0.463	0.110	0.249	0.678	394
323.00	0.488	0.110	0.273	0.703	401
326.00	0.512	0.110	0.297	0.727	328
328.00	0.537	0.115	0.311	0.762	392
344.00	0.561	0.115	0.336	0.786	407
348.00	0.585	0.124	0.342	0.829	409
349.00	0.610	0.124	0.367	0.852	397
367.00	0.634	0.123	0.393	0.876	415
377.00	0.659	0.132	0.400	0.917	392
404.00	0.684	0.136	0.417	0.950	397
408.00	0.709	0.140	0.435	0.983	405
410.00	0.734	0.139	0.461	1.006	411
449.00	0.759	0.143	0.479	1.038	401
479.00	0.784	0.146	0.497	1.070	400
497.00	0.809	0.149	0.516	1.101	407
538.00	0.834	0.152	0.535	1.132	395
539.00	0.859	0.155	0.554	1.163	393
561.00	0.884	0.162	0.567	1.201	397
563.00	0.909	0.164	0.587	1.230	416
570.00	0.934	0.170	0.600	1.267	416
573.00	0.959	0.169	0.627	1.290	404
581.00	0.985	0.171	0.649	1.320	411
586.00	1.014	0.174	0.674	1.355	408
604.00	1.060	0.185	0.697	1.422	405
621.00	1.119	0.208	0.712	1.525	392
635.00	1.181	0.207	0.776	1.587	396

Figure 18.35 *continued*

Recurrence Data Analysis

Age	Sample MCF	Standard Error	95% Confidence Limits Lower	Upper	Unit ID
640.00	1.244	0.226	0.800	1.687	394
646.00	1.320	0.229	0.873	1.768	389
653.00	1.432	0.252	0.937	1.926	328
653.00	1.543	0.312	0.932	2.154	328

Parametric modeling of the repair process requires more assumptions than nonparametric modeling, and considerable work has been done in this area. Ascher and Feingold (1984), Tobias and Trindade (1995), Crowder et al. (1991), Meeker and Escobar (1998), Cook and Lawless (2007), Abernethy (2006), and Rigdon and Basu (2000) describe parametric models for repair processes. Repairs are sometimes modeled as a nonhomogeneous Poisson process, and the RELIABILITY procedure provides several forms of Poisson process models for recurrent events data. See the section "Parametric Models for Recurrent Events Data" on page 1381 for details about the Poisson process models that the RELIABILITY procedure provides.

A nonparametric MCF plot might be a first step in modeling a repair process, and, in many cases, provide the required answers without further analysis. An estimate of the MCF for a sample of systems aids engineers in determining the repair rate at any age and the increase or decrease of repair rate with population age. The estimate is also useful for predicting the number of future repairs.

Comparison of Two Samples of Repair Data

Nelson (2002) and Doganaksoy and Nelson (1998) show how the difference of MCFs from two samples can be used to compare the populations from which they are drawn. The RELIABILITY procedure provides Doganaksoy and Nelson's confidence intervals for the pointwise difference of the two MCFs, which can be used to assess whether the difference is statistically significant.

Doganaksoy and Nelson (1998) give an example of two samples of locomotives with braking grids from two different production batches. Figure 18.36 contains a listing of the data. The variable ID is a unique identifier for individual locomotives. The variable Days provides the locomotive age in days. The variable Value is 1 if the age corresponds to a braking grid replacement or -1 if the age corresponds to the locomotive's latest age (the current end of its history). The variable Sample is a group variable that identifies the grid production batch.

```
data Grids;
   if _N_ < 40 then Sample = 'Sample1';
   else Sample = 'Sample2';
   input ID$ Days Value @@;
   datalines;
S1-01 462  1    S1-01 730 -1    S1-02 364  1    S1-02 391  1
S1-02 548  1    S1-02 724 -1    S1-03 302  1    S1-03 444  1
S1-03 500  1    S1-03 730 -1    S1-04 250  1    S1-04 730 -1
S1-05 500  1    S1-05 724 -1    S1-06  88  1    S1-06 724 -1
S1-07 272  1    S1-07 421  1    S1-07 552  1    S1-07 625  1
S1-07 719 -1    S1-08 481  1    S1-08 710 -1    S1-09 431  1
S1-09 710 -1    S1-10 367  1    S1-10 710 -1    S1-11 635  1
S1-11 650  1    S1-11 708 -1    S1-12 402  1    S1-12 700 -1
S1-13  33  1    S1-13 687 -1    S1-14 287  1    S1-14 687 -1
S1-15 317  1    S1-15 498  1    S1-15 657 -1    S2-01 203  1
S2-01 211  1    S2-01 277  1    S2-01 373  1    S2-01 511 -1
S2-02 293  1    S2-02 503 -1    S2-03 173  1    S2-03 470 -1
S2-04 242  1    S2-04 464 -1    S2-05  39  1    S2-05 464 -1
S2-06  91  1    S2-06 462 -1    S2-07 119  1    S2-07 148  1
S2-07 306  1    S2-07 461 -1    S2-08 382  1    S2-08 460 -1
S2-09 250  1    S2-09 434 -1    S2-10 192  1    S2-10 448 -1
S2-11 369  1    S2-11 448 -1    S2-12  22  1    S2-12 447 -1
S2-13  54  1    S2-13 441 -1    S2-14 194  1    S2-14 432 -1
S2-15  61  1    S2-15 419 -1    S2-16  19  1    S2-16 185  1
S2-16 419 -1    S2-17 187  1    S2-17 416 -1    S2-18  93  1
S2-18 205  1    S2-18 264  1    S2-18 415 -1
;
```

Figure 18.36 Partial Listing of the Braking Grids Data

Obs	Sample	ID	Days	Value
1	Sample1	S1-01	462	1
2	Sample1	S1-01	730	-1
3	Sample1	S1-02	364	1
4	Sample1	S1-02	391	1
5	Sample1	S1-02	548	1
6	Sample1	S1-02	724	-1
7	Sample1	S1-03	302	1
8	Sample1	S1-03	444	1
9	Sample1	S1-03	500	1
10	Sample1	S1-03	730	-1
11	Sample1	S1-04	250	1
12	Sample1	S1-04	730	-1
13	Sample1	S1-05	500	1
14	Sample1	S1-05	724	-1
15	Sample1	S1-06	88	1
16	Sample1	S1-06	724	-1
17	Sample1	S1-07	272	1
18	Sample1	S1-07	421	1
19	Sample1	S1-07	552	1
20	Sample1	S1-07	625	1

The following statements request the Nelson (1995) nonparametric estimate and confidence limits for the difference of the MCF functions shown in Figure 18.37 for the braking grids:

```
proc reliability data=Grids;
   unitid ID;
   mcfplot Days*Value(-1) = Sample / mcfdiff;
run;
```

The MCFPLOT statement requests a plot of each MCF estimate as a function of age (provided by Days), and it specifies that the end of history for each system is identified by Value equal to -1. The variable Sample identifies the two samples of braking grids. The option MCFDIFF requests that the difference between the MCFs of the two groups given in the variable Sample be computed and plotted. Confidence limits for the MCF difference are also computed and plotted. The UNITID statement specifies that the variable Id uniquely identify each system.

Figure 18.37 shows the plot of the MCF difference function and pointwise 95% confidence intervals. Since the pointwise confidence limits do not include zero for some system ages, the difference between the two populations is statistically significant. A listing of the tabular output is shown in Figure 18.38. It contains a summary of the repair data for the two samples, estimates, standard errors, and confidence intervals for the MCF difference. A statistical test for different MCFs is also computed and is displayed in the table "Tests for Equality of Mean Functions." The tests also indicate a significant difference between the two samples.

Figure 18.37 Mean Cumulative Function Difference

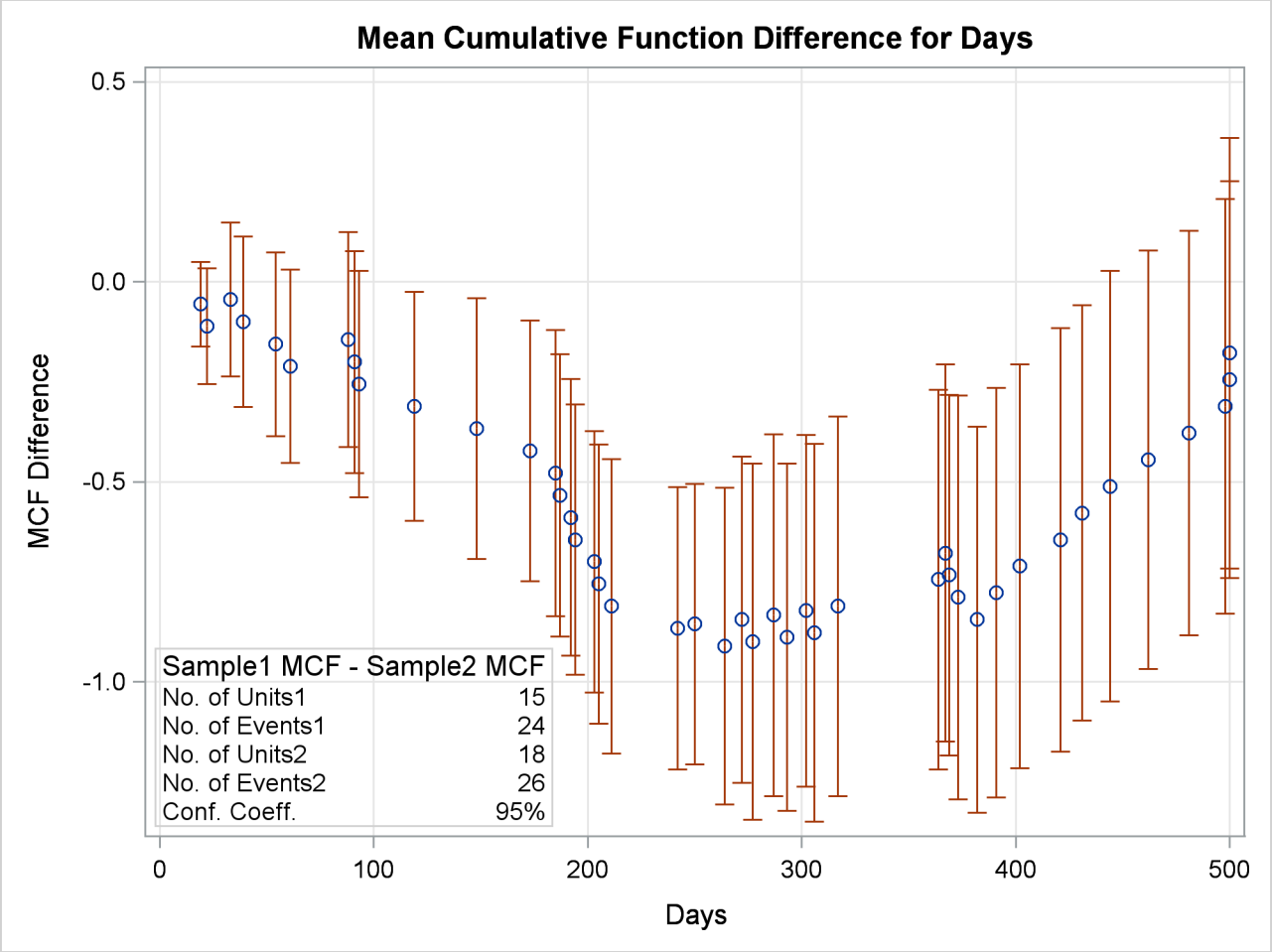


Figure 18.38 Listing of the Output for the Braking Grids Data

MCF Difference Data Summary	
Input Data Set	WORK.GRIDS
Group 1	Sample1
Observations Used	39
Number of Units	15
Number of Events	24
Group 2	Sample2
Observations Used	44
Number of Units	18
Number of Events	26

Figure 18.38 *continued*

Sample MCF Differences

Age	MCF Difference	Standard Error	95% Confidence Limits Lower	Upper	Unit ID
19.00	-0.056	0.054	-0.161	0.050	S2-16
22.00	-0.111	0.074	-0.256	0.034	S2-12
33.00	-0.044	0.098	-0.237	0.148	S1-13
39.00	-0.100	0.109	-0.313	0.113	S2-05
54.00	-0.156	0.117	-0.385	0.074	S2-13
61.00	-0.211	0.124	-0.453	0.031	S2-15
88.00	-0.144	0.137	-0.414	0.125	S1-06
91.00	-0.200	0.142	-0.478	0.078	S2-06
93.00	-0.256	0.145	-0.539	0.028	S2-18
119.00	-0.311	0.146	-0.598	-0.024	S2-07
148.00	-0.367	0.167	-0.693	-0.040	S2-07
173.00	-0.422	0.166	-0.748	-0.097	S2-03
185.00	-0.478	0.182	-0.835	-0.120	S2-16
187.00	-0.533	0.180	-0.886	-0.181	S2-17
192.00	-0.589	0.177	-0.935	-0.243	S2-10
194.00	-0.644	0.172	-0.982	-0.307	S2-14
203.00	-0.700	0.167	-1.027	-0.373	S2-01
205.00	-0.756	0.178	-1.105	-0.407	S2-18
211.00	-0.811	0.188	-1.179	-0.443	S2-01
242.00	-0.867	0.180	-1.219	-0.514	S2-04
250.00	-0.856	0.179	-1.207	-0.504	S1-04,S2-09
264.00	-0.911	0.202	-1.307	-0.515	S2-18
272.00	-0.844	0.208	-1.252	-0.437	S1-07
277.00	-0.900	0.227	-1.345	-0.455	S2-01
287.00	-0.833	0.231	-1.286	-0.380	S1-14
293.00	-0.889	0.222	-1.323	-0.455	S2-02
302.00	-0.822	0.224	-1.262	-0.383	S1-03
306.00	-0.878	0.241	-1.350	-0.406	S2-07
317.00	-0.811	0.242	-1.286	-0.337	S1-15
364.00	-0.744	0.242	-1.219	-0.270	S1-02
367.00	-0.678	0.241	-1.150	-0.206	S1-10
369.00	-0.733	0.230	-1.185	-0.282	S2-11
373.00	-0.789	0.257	-1.293	-0.284	S2-01
382.00	-0.844	0.246	-1.327	-0.362	S2-08
391.00	-0.778	0.261	-1.290	-0.266	S1-02
402.00	-0.711	0.258	-1.217	-0.206	S1-12
421.00	-0.644	0.270	-1.174	-0.115	S1-07
431.00	-0.578	0.265	-1.097	-0.059	S1-09
444.00	-0.511	0.275	-1.049	0.027	S1-03
462.00	-0.444	0.267	-0.968	0.079	S1-01
481.00	-0.378	0.258	-0.883	0.128	S1-08
498.00	-0.311	0.265	-0.830	0.208	S1-15
500.00	-0.244	0.253	-0.741	0.252	S1-05
500.00	-0.178	0.275	-0.716	0.360	S1-03

Figure 18.38 *continued*

Tests for Equality of Mean Cumulative Functions

Weight Function	Statistic	Variance	Chi-Square	DF	Pr > Chi Square
Constant	-3.673285	4.556053	2.961560	1	0.0853
Linear	-4.435032	1.424770	13.805393	1	0.0002

You can fit a parametric model that uses Sample as a classification variable. This results in a model with a common shape parameter for the two groups but with different scale parameters. Suppose you want estimates of the parametric mean and intensity functions at values of the time variable 500, 600, 700, 800, 900, and 1,000 days for each of the two groups. The following statements create a new input data set that has observations for the desired prediction times appended to it. The additional observations are not used in the analysis, because the censoring variable Value is set to missing for those observations. Values of the mean and intensity function are computed, however, in the table that is produced by specifying the OBSTATS option in the MODEL statement.

The following statements create the new data set by appending observations to the original Grids data set:

```
data Predict;
   Control=1;
   if _N_ < 7 then Sample = 'Sample1';
   else Sample = 'Sample2';

   input ID$ Days Value;
   datalines;
9999 500  .
9999 600  .
9999 700  .
9999 800  .
9999 900  .
9999 1000 .
9999 500  .
9999 600  .
9999 700  .
9999 800  .
9999 900  .
9999 1000 .
;

data Grids;
   set Predict Grids;
run;
```

The following statements fit a nonhomogeneous Poisson process with a power law mean function that uses Sample as a two-level covariate. The OBSTATS option requests that predicted values be computed for values of the variable Control equal to 1. The MCFPLOT statement plots the fitted model as well as the nonparametric estimates of the MCF. Parametric confidence limits are displayed by default.

```
proc reliability data=Grids;
   unitid ID;
   distribution nhpp(pow);
   class Sample;
   model   Days*Value(-1) = Sample /obstats(control=Control);
   mcfplot Days*Value(-1) = Sample /fit=model overlay;
run;
```

Figure 18.39 Predicted Mean and Intensity Function for the Braking Grids Data

The RELIABILITY Procedure

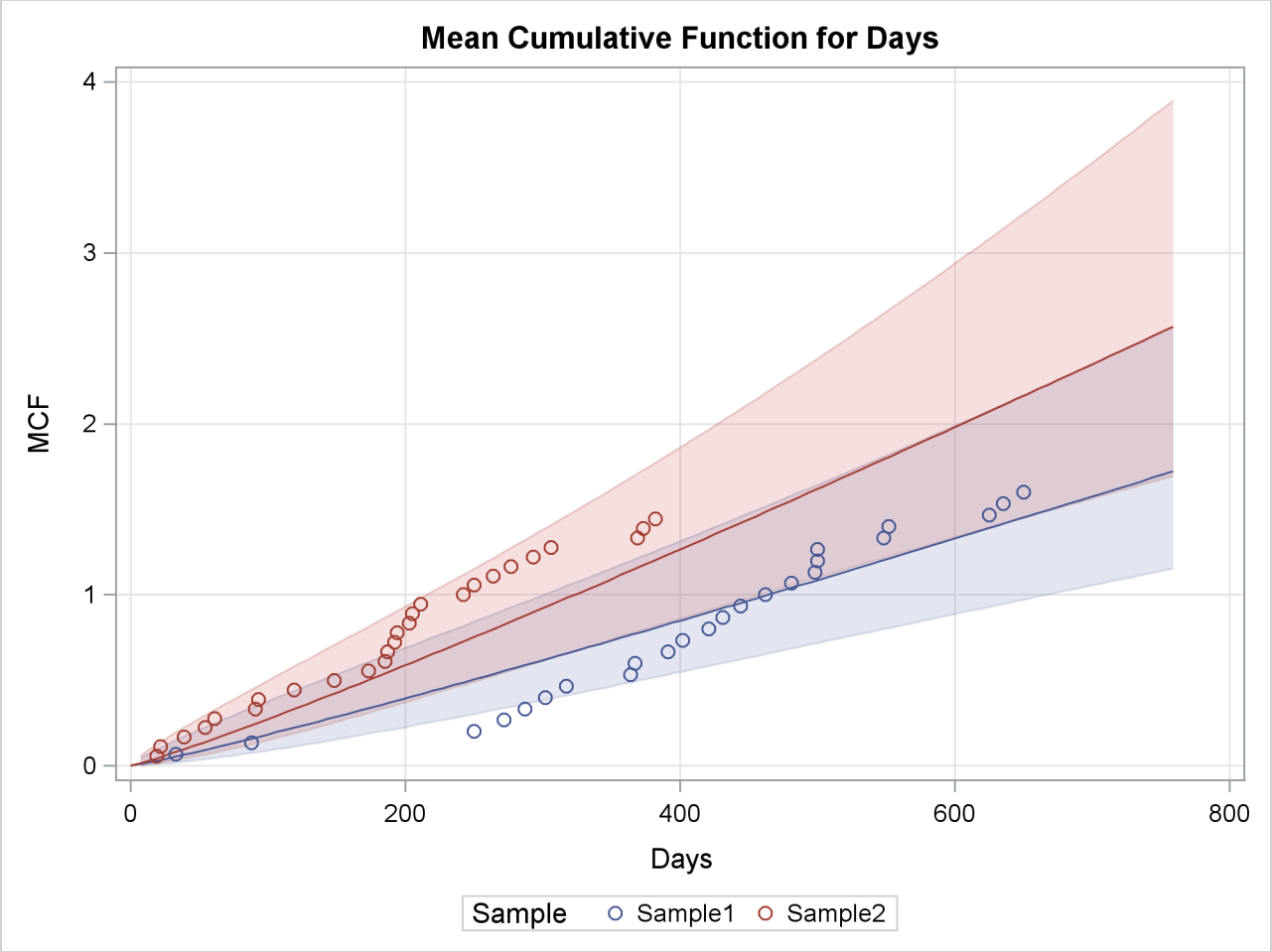
Observation Statistics

Days	Value	Sample	ID	Xbeta	Shape	MCF	MCF_Lower	MCF_Upper
500	.	Sample1	9999	464.04648	1.1050556	1.0859585	0.7176451	1.6432995
600	.	Sample1	9999	464.04648	1.1050556	1.3283512	0.8874029	1.9884054
700	.	Sample1	9999	464.04648	1.1050556	1.5750445	1.0556772	2.3499278
800	.	Sample1	9999	464.04648	1.1050556	1.8254803	1.2215469	2.7279987
900	.	Sample1	9999	464.04648	1.1050556	2.0792348	1.3846219	3.1223089
1000	.	Sample1	9999	464.04648	1.1050556	2.3359745	1.5448083	3.5323327
500	.	Sample2	9999	323.23791	1.1050556	1.6193855	1.1012136	2.3813813
600	.	Sample2	9999	323.23791	1.1050556	1.9808425	1.335699	2.9375905
700	.	Sample2	9999	323.23791	1.1050556	2.3487125	1.5633048	3.5287107
800	.	Sample2	9999	323.23791	1.1050556	2.7221634	1.7845232	4.1524669
900	.	Sample2	9999	323.23791	1.1050556	3.100563	2.0000301	4.8066733
1000	.	Sample2	9999	323.23791	1.1050556	3.4834144	2.2104841	5.4893747

Observation Statistics

Days	MCF_StdErr	Intensity	Int_Lower	Int_Upper	Int_StdErr
500	0.2295199	0.0024001	0.0015543	0.0037061	0.000532
600	0.2733977	0.0024465	0.0015458	0.0038719	0.0005731
700	0.3215248	0.0024864	0.0015325	0.0040343	0.000614
800	0.3741606	0.0025216	0.0015171	0.0041911	0.0006537
900	0.4313142	0.002553	0.0015011	0.0043418	0.0006917
1000	0.4928631	0.0025814	0.0014852	0.0044866	0.000728
500	0.3186232	0.003579	0.0021872	0.0058566	0.0008993
600	0.3982653	0.0036482	0.0021492	0.0061928	0.0009849
700	0.4878045	0.0037078	0.0021124	0.006508	0.0010643
800	0.5864921	0.0037602	0.002078	0.0068042	0.0011378
900	0.6935604	0.003807	0.002046	0.0070837	0.0012061
1000	0.8083117	0.0038494	0.0020164	0.0073484	0.0012699

Figure 18.40 Fitted Model



The predicted values of the mean and intensity functions at the desired values of Days, with standard errors and confidence limits, are shown in Figure 18.39.

A plot of the fitted mean function, along with nonparametric estimates for the two samples, is shown in Figure 18.40.

Analysis of Interval Age Recurrence Data

You can analyze recurrence data when the recurrence ages are grouped into intervals, instead of being exact ages. Figure 18.41 shows a listing of a SAS data set containing field data on replacements of defrost controls in 22,914 refrigerators, whose ages are grouped by months in service. Nelson (2002, problem 5.2, chapter 5) presents these data. Grouping the control data on the 22,914 refrigerators into age intervals enables you to represent the data by 29 data records, instead of requiring a single data record for each refrigerator, as required for exact recurrence data.

The variables Lower and Upper are the lower and upper monthly interval endpoints, Recurrences is the number of defrost control replacements in each month, and Censored is the number of refrigerator histories

censored in each month—that is, the number with current age in the monthly interval. Data are entered as shown in Figure 18.41.

Figure 18.41 Listing of the Defrost Controls Data

Obs	Lower	Upper	Recurrences	Censored
1	0	1	83	0
2	1	2	35	0
3	2	3	23	0
4	3	4	15	0
5	4	5	22	0
6	5	6	16	3
7	6	7	13	36
8	7	8	12	24
9	8	9	15	29
10	9	10	15	37
11	10	11	24	40
12	11	12	12	20041
13	12	13	7	14
14	13	14	11	17
15	14	15	15	13
16	15	16	6	28
17	16	17	8	22
18	17	18	9	27
19	18	19	9	64
20	19	20	5	94
21	20	21	6	119
22	21	22	6	118
23	22	23	6	138
24	23	24	5	1188
25	24	25	7	17
26	25	26	5	28
27	26	27	5	99
28	27	28	6	128
29	28	29	3	590

The following SAS statements create the plot of the sample MCF of defrost control replacement shown in Figure 18.42 and the tabular listing in Figure 18.43:

```
proc reliability data=defrost;
   mcfplot ( interval    = Lower Upper
             recurrences = Recurrences
             censor      = Censored ) / plotsymbol = X
                                        vaxis = 0 to .12 by .04
                                        interpolate = join;
run;
```

Pointwise confidence limits are included on the plot and in the tabular listing. These limits are approximate, and are usually shorter than the correct limits, which have not been developed for interval data.

Here, INTERVAL = LOWER UPPER specifies the input data set variables Lower and Upper as the age interval endpoints. The variable Recurrences identifies the number of recurrences (defrost control replacements) in each time interval, and Censored identifies the number of units censored in each interval (number in an age interval or removed from the sample in an age interval).

Figure 18.42 MCF Plot for the Defrost Controls

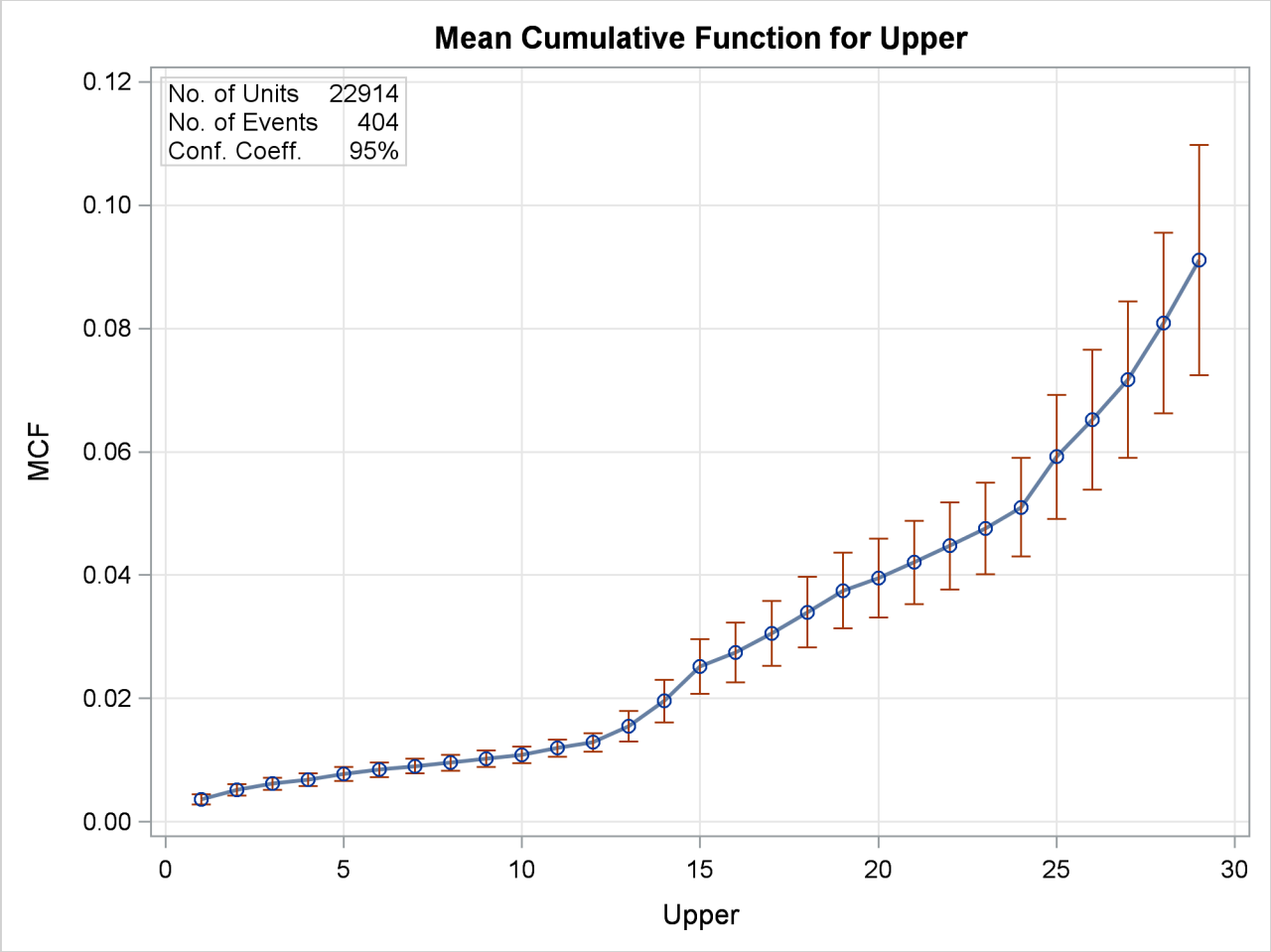


Figure 18.43 Listing of the Output for the Defrost Controls Data

Recurrence Data Summary	
Input Data Set	WORK.DEFROST
Observations Used	29
Number of Units	22914
Number of Events	404

Figure 18.43 *continued*

Recurrence Data Analysis						
	Endpoints				Naive 95% Confidence Limits	
	Lower	Upper	Sample MCF	Standard Error	Lower	Upper
	0.00	**1.00**	0.004	0.000	0.003	0.004
	1.00	**2.00**	0.005	0.000	0.004	0.006
	2.00	**3.00**	0.006	0.001	0.005	0.007
	3.00	**4.00**	0.007	0.001	0.006	0.008
	4.00	**5.00**	0.008	0.001	0.007	0.009
	5.00	**6.00**	0.008	0.001	0.007	0.010
	6.00	**7.00**	0.009	0.001	0.008	0.010
	7.00	**8.00**	0.010	0.001	0.008	0.011
	8.00	**9.00**	0.010	0.001	0.009	0.012
	9.00	**10.00**	0.011	0.001	0.010	0.012
	10.00	**11.00**	0.012	0.001	0.011	0.013
	11.00	**12.00**	0.013	0.001	0.011	0.014
	12.00	**13.00**	0.015	0.001	0.013	0.018
	13.00	**14.00**	0.020	0.002	0.016	0.023
	14.00	**15.00**	0.025	0.002	0.021	0.030
	15.00	**16.00**	0.027	0.002	0.023	0.032
	16.00	**17.00**	0.031	0.003	0.025	0.036
	17.00	**18.00**	0.034	0.003	0.028	0.040
	18.00	**19.00**	0.038	0.003	0.031	0.044
	19.00	**20.00**	0.040	0.003	0.033	0.046
	20.00	**21.00**	0.042	0.003	0.035	0.049
	21.00	**22.00**	0.045	0.004	0.038	0.052
	22.00	**23.00**	0.048	0.004	0.040	0.055
	23.00	**24.00**	0.051	0.004	0.043	0.059
	24.00	**25.00**	0.059	0.005	0.049	0.069
	25.00	**26.00**	0.065	0.006	0.054	0.077
	26.00	**27.00**	0.072	0.006	0.059	0.084
	27.00	**28.00**	0.081	0.007	0.066	0.096
*	**28.00**	**29.00**	0.091	0.010	0.072	0.110

*** The estimate and limits for this interval may not be appropriate.**

The last interval is always marked with a footnote indicating that estimates for the last interval may be biased since censoring ages often are not uniformly spread over that interval.

Analysis of Binomial Data

This example illustrates the analysis of binomial proportions of capacitor failures from nine circuit boards. The data are given by Nelson (1982, p. 451). The following statements create and list a SAS data set named BINEX containing the data:

```
data binex;
   input board sample fail;
   datalines;
1 84 2
2 72 3
3 72 5
4 119 19
5 538 21
6 51 2
7 517 9
8 462 18
9 143 2
;
```

Figure 18.44 displays a listing of the data. The variable Board identifies the circuit board, the variable Sample provides the number of capacitors on the boards, and the variable Fail provides the number of capacitors failing on the boards.

Figure 18.44 Listing of the Capacitor Data

Obs	board	sample	fail
1	1	84	2
2	2	72	3
3	3	72	5
4	4	119	19
5	5	538	21
6	6	51	2
7	7	517	9
8	8	462	18
9	9	143	2

The following statements analyze the proportion of capacitors failing:

```
proc reliability data=Binex;
   distribution binomial;
   analyze fail(sample) = board / predict(1000)
                                  tolerance(.05);
run;
```

The DISTRIBUTION statement specifies the binomial distribution. The analysis requested with the ANALYZE statement consists of tabular output only. Graphical output is not available for the binomial distribution. The variable Fail provides the number of capacitors failing on each board, the variable Sample provides the sample size (number of capacitors) for each board, and the variable Board identifies the individual boards. The statement option PREDICT(1000) requests the predicted number of capacitors failing and prediction limits in a future sample of size 1000. The option TOLERANCE(.05) requests the sample size required to estimate the binomial proportion to within 0.05. Figure 18.45 displays the results of the analysis.

The "Pooled Data Analysis" table displays the estimated binomial probability and exact binomial confidence limits when data from all boards are pooled. The chi-square value and p-value for a test of equality of the binomial probabilities for all of the boards are also shown. In this case, the p-value is less than 0.05, so you reject the test of equality at the 0.05 level.

The "Predicted Values and Limits" table provides the predicted failure count and prediction limits for the number of capacitors that would fail in a future sample of size 1000 for the pooled data, as requested with the PREDICT(1000) option. The "Sample Size for Estimation" table gives the sample size required to estimate the binomial probability to within 0.05 for the pooled data, as requested with the TOLERANCE(.05) option.

The "Estimates by Group" table supplies the estimated binomial probability, confidence limits, and the contribution to the total chi-square for each board. The pooled values are shown in the last line of the table.

The "Predicted Values by Group" table gives the predicted counts in a future sample of size 1000, prediction limits, and the sample size required to estimate the binomial probability to within the tolerance of 0.05 for each board. Values for the pooled data are shown in the last line of the table.

Figure 18.45 Analysis of the Capacitor Data

The RELIABILITY Procedure

Model Information - All Groups	
Input Data Set	WORK.BINEX
Events Variable	fail
Trials Variable	sample
Distribution	Binomial
Confidence Coefficient	95%
Observations Used	9

Binomial Data Analysis	
Pooled Events	81.0000
Pooled Trials	2058.0000
Estimate of Proportion	0.0394
Lower Limit For Proportion	0.0314
Upper Limit For Proportion	0.0487
ChiSquare	56.8504
Pr>ChiSquare	0.0000

Predicted Value and Limits	
Sample Size For Prediction	1000.0000
Predicted Count	39.3586
Lower Prediction Limit	24.8424
Upper Prediction Limit	56.3237

Sample Size For Estimation	
Tolerance	0.0500
Sample Size For Tolerance	58.0975

Figure 18.45 *continued*

Estimates By Group

Group	Events	Trials	Prop	95% Confidence Limits Lower	Upper	X2
1	2	84	0.0238	0.0029	0.0834	0.5371
2	3	72	0.0417	0.0087	0.1170	0.0101
3	5	72	0.0694	0.0229	0.1547	1.7237
4	19	119	0.1597	0.0990	0.2381	45.5528
5	21	538	0.0390	0.0243	0.0590	0.0015
6	2	51	0.0392	0.0048	0.1346	0.0000
7	9	517	0.0174	0.0080	0.0328	6.5884
8	18	462	0.0390	0.0233	0.0609	0.0019
9	2	143	0.0140	0.0017	0.0496	2.4348
Pooled	81	2058	0.0394	0.0314	0.0487	56.8504

Predicted/Tolerance Values By Group

Group	Predicted Count	95% Prediction Limits Lower	Upper	Tolerance Sample Size
1	23.81	1.5476	88.5824	35.71
2	41.67	6.9416	124.6142	61.36
3	69.44	20.4052	165.3499	99.30
4	159.66	91.9722	254.5444	206.17
5	39.03	20.1599	64.7140	57.64
6	39.22	3.3970	144.2494	57.90
7	17.41	5.3506	36.7531	26.28
8	38.96	19.3343	66.3850	57.53
9	13.99	0.3851	53.0715	21.19
Pooled	39.36	24.8424	56.3237	58.10

Three-Parameter Weibull

Meeker and Escobar (1998) give an example of the number of cycles to fatigue failure of specimens of a certain alloy. The first 67 specimens experienced failure, and the last five specimens had no failure at 300,000 cycles. The following statements create a SAS data set named Alloy that contains the number of cycles (in thousands) to failure or end of test for the specimens:

```
data Alloy;
   input kCycles@@;
   Cen = _n_ > 67;
   label kCycles = 'Fatigue Life in Thousands of Cycles';
   datalines;
94  96  99  99  104 108 112 114 117 117
118 121 121 123 129 131 133 135 136 139
139 140 141 141 143 144 149 149 152 153
159 159 159 159 162 168 168 169 170 170
171 172 173 176 177 180 180 184 187 188
189 190 196 197 203 205 211 213 224 226
227 256 257 269 271 274 291 300 300 300
300 300
;
```

The following SAS statements fit a three-parameter Weibull distribution to the specimen lifetimes, in thousands of cycles. The PROFILE option requests a profile likelihood plot for the threshold parameter. ODS Graphics must be enabled to create a profile likelihood plot with the PROFILE option.

```
proc Reliability data=Alloy;
   distribution Weibull3;
   Pplot kCycles*Cen(1) / Profile(noconf range=(50,100)) LifeUpper=500;
run;
```

Figure 18.46 shows the maximum likelihood estimates of the Weibull threshold, shape and scale parameters, and the corresponding extreme value location and scale parameter estimates.

Figure 18.46 Three-Parameter Weibull Parameter Estimates

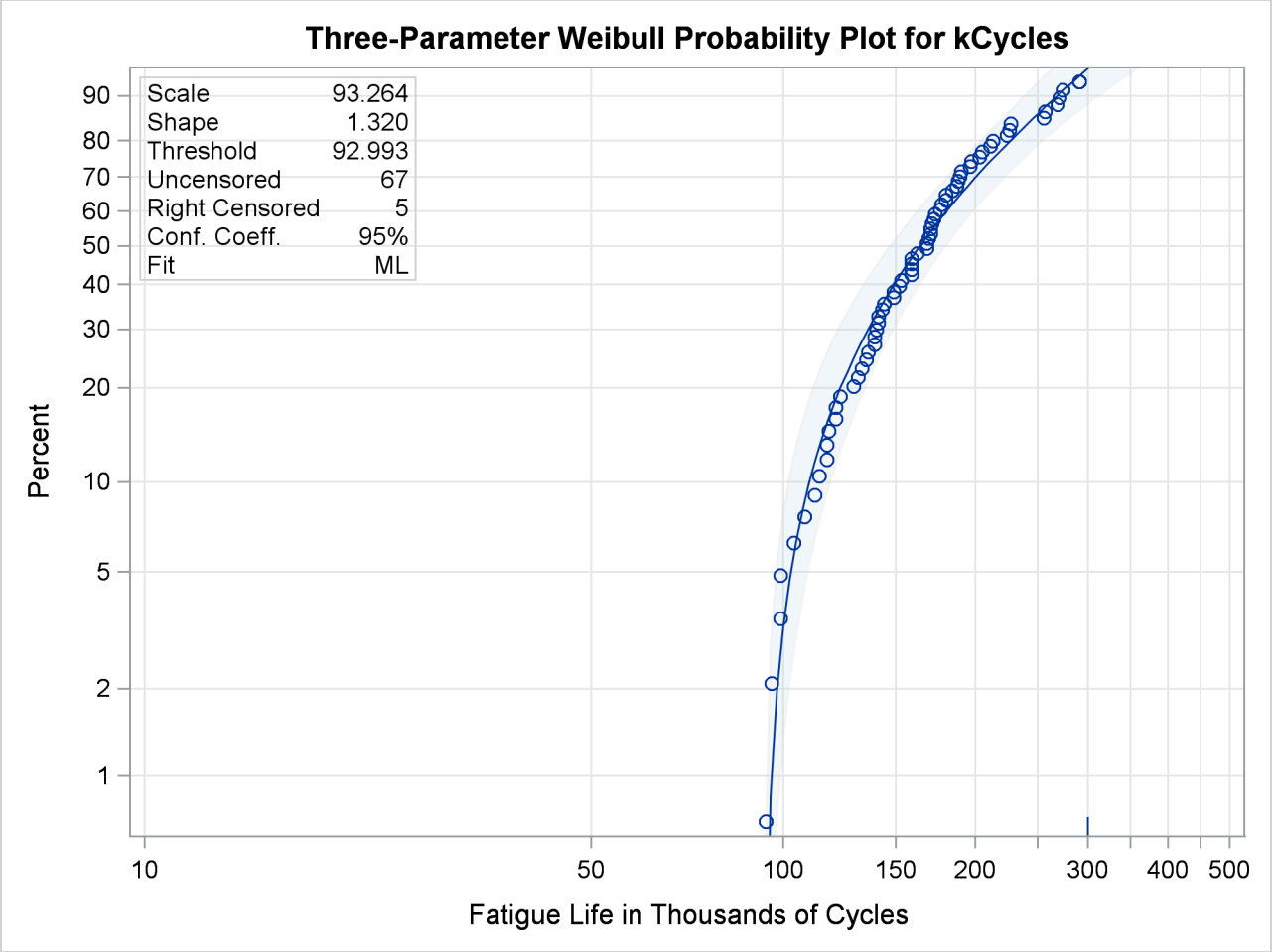
The RELIABILITY Procedure

Three-Parameter Weibull Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
EV Location	4.5354	0.1009	4.3377	4.7332
EV Scale	0.7575	0.0898	0.6005	0.9556
Weibull Scale	93.2642	9.4082	76.5329	113.6531
Weibull Shape	1.3202	0.1565	1.0465	1.6654
Weibull Threshold	92.9928	1.9516	89.1676	96.8179

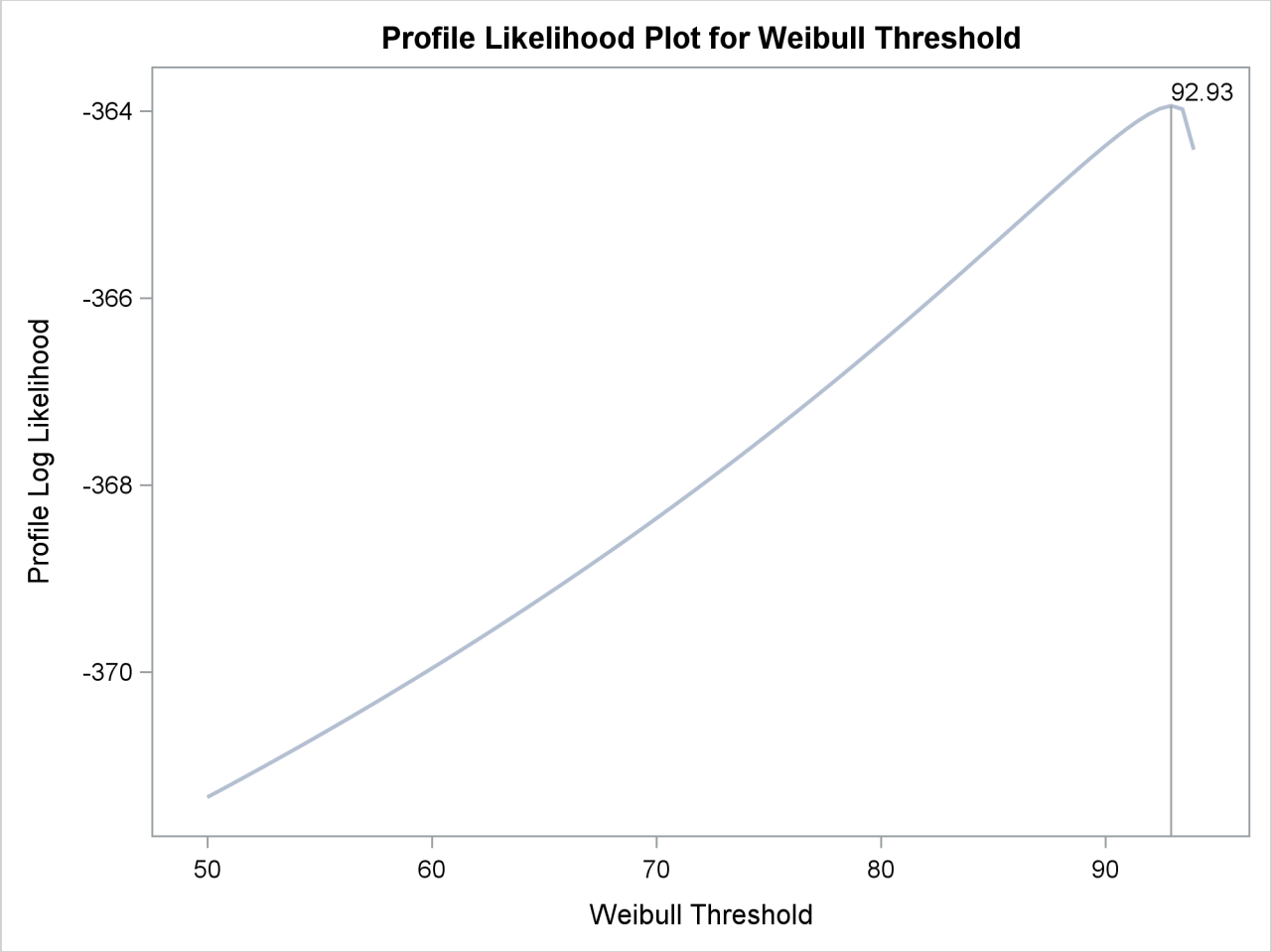
A probability plot of the failure lifetimes and the fitted three-parameter Weibull distribution is shown in Figure 18.47.

Figure 18.47 Three-Parameter Weibull Probability Plot



A profile likelihood plot for the threshold parameter is shown in Figure 18.48. The threshold value at the maximum log likelihood corresponds to the maximum likelihood estimate of the threshold parameter.

Figure 18.48 Profile Likelihood for Three-Parameter Weibull Threshold



Parametric Model for Recurrent Events Data

The following SAS statements fit a non-homogeneous Poisson process with a power intensity function model to the valve seat data described in the section "Analysis of Recurrence Data on Repairs" on page 1245. The FIT=MODEL option in the MCFPLOT statement requests that the fitted model be plotted on the plot with the nonparametric mean cumulative function estimates.

```
proc reliability data=Valve;
   unitid id;
   distribution Nhpp(Pow);
   model   Days*Value(-1);
   mcfplot Days*Value(-1) / Fit=Model Noconf;
run;
```

The model parameter estimates are shown in Figure 18.49.

Figure 18.49 Power Model Parameter Estimates for the Valve Seat Data

The RELIABILITY Procedure

NHPP-Power Parameter Estimates

Parameter	Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept	553.6430	57.8636	451.0941	679.5048
Shape	1.3996	0.2005	1.0570	1.8533

Figure 18.50 displays a plot of nonparametric estimates of the mean cumulative function and the fitted model mean function. The parametric model matches the data well except at the upper end of the range of repair times, where the parametric model does not capture the rapid increase in the number of replacements of the valve seats. For this reason, the parametric model might not be appropriate for predicting future repairs of the engines.

Figure 18.50 Mean Cumulative Function Plot for the Valve Seat Data

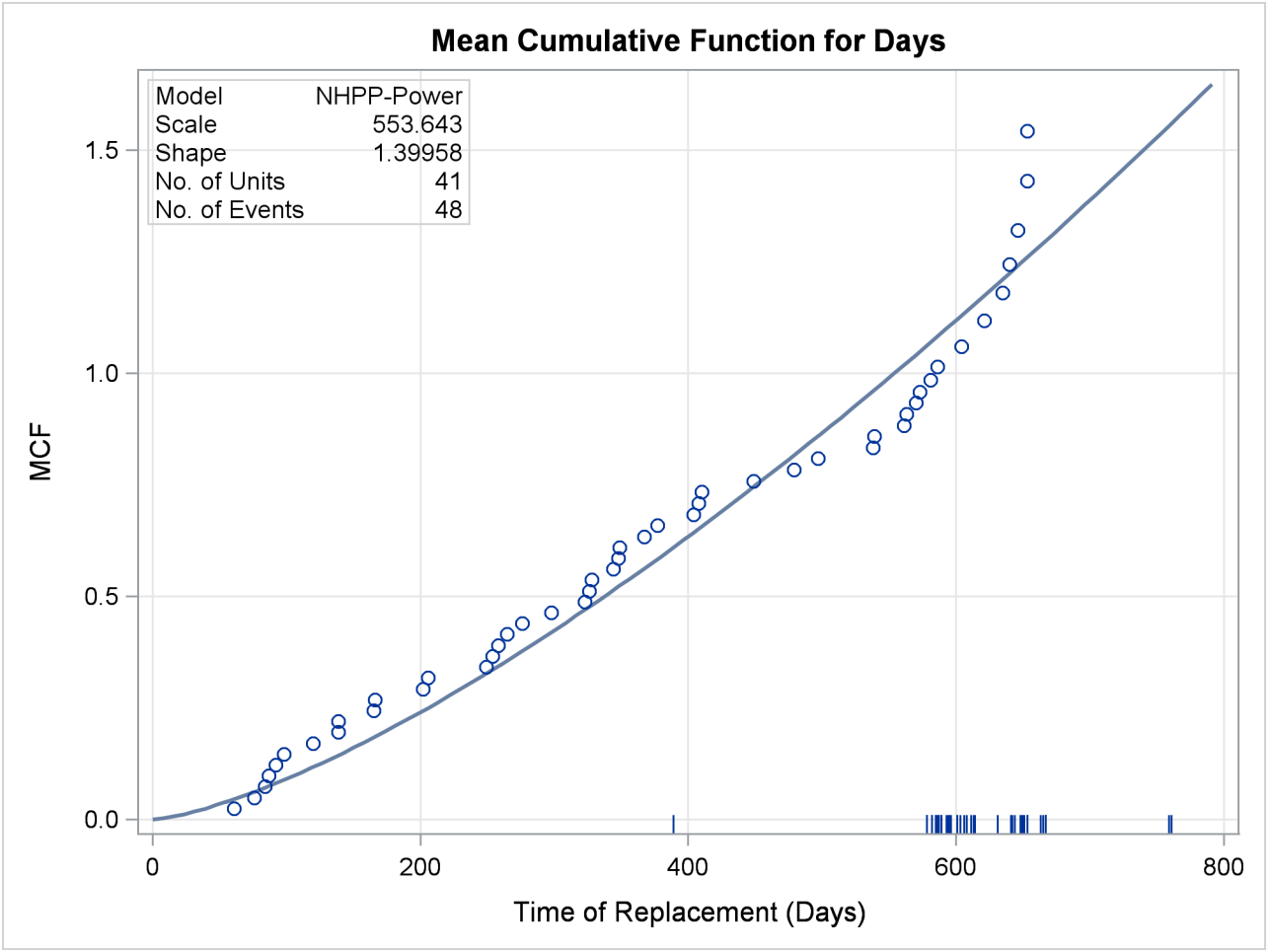
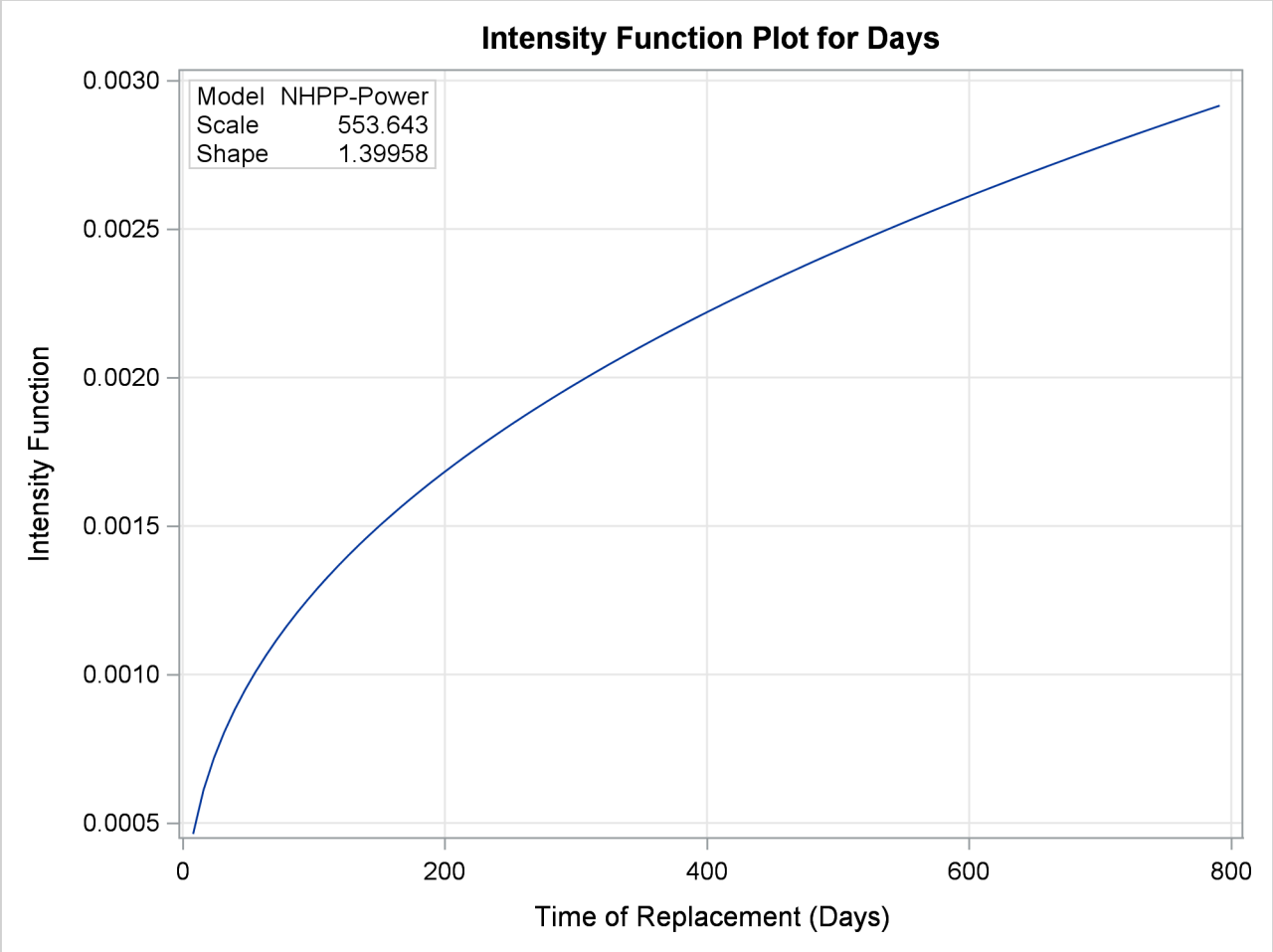


Figure 18.51 shows the parametric model intensity function. The intensity function increases with time, indicating an increasing rate of repairs. This is consistent with the parameter estimates in Figure 18.49, where a shape parameter significantly greater than 1 indicates an increasing failure rate.

Figure 18.51 Intensity Function Plot for the Valve Seat Data



Parametric Model for Interval Recurrent Events Data

Byar (1980) provides data for recurrences of bladder tumors in patients in a clinical trial. Figure 18.52 is a partial listing of data for 86 patients, of which 48 were given a placebo and 38 were treated with the drug Thiotepa. The data are here grouped into one month intervals.

Figure 18.52 Partial Listing of the Bladder Tumor Data

Obs	Group	Age	Age1	N	R
1	Placebo	0	1	48	0
2	Placebo	1	2	47	0
3	Placebo	2	3	46	1
4	Placebo	3	4	46	4
5	Placebo	4	5	46	7
6	Placebo	5	6	45	0
7	Placebo	6	7	45	2
8	Placebo	7	8	45	4
9	Placebo	8	9	44	1
10	Placebo	9	10	44	2
11	Placebo	10	11	44	4
12	Placebo	11	12	42	2
13	Placebo	12	13	42	1
14	Placebo	13	14	42	4
15	Placebo	14	15	42	1
16	Placebo	15	16	41	1
17	Placebo	16	17	41	5
18	Placebo	17	18	41	4
19	Placebo	18	19	41	4
20	Placebo	19	20	38	1

The following SAS statements fit a non-homogeneous Poisson process model with a power intensity function to the interval recurrence data. Some patients were lost to follow-up in each month, so the number of patients observed changes from month to month. The variable N provides the number of patients available at the beginning of each month and assumed to be observed throughout the month. The variable R is the number of recurrences of tumors in each month. Age represents the number of months after randomization into the trial (starting with month 0), and Age1=Age+1 is the end of a month. The variable Group represents the treatment, either Placebo or Thiotepa. The MODEL statement requests a maximum likelihood fit of the model with Group as a classification variable. The MCFPLOT statement requests a plot of the fitted model and nonparametric estimates of the mean cumulative function for each group.

```
proc reliability data=Tumor;
   distribution nhpp(pow);
   freq R;
   nenter N;
   class Group;
   model  (Age Age1) = Group;
   mcfplot(Age Age1) = Group / fit=Model;
run;
```

The resulting maximum likelihood parameter estimates are shown in Figure 18.53.

Figure 18.53 Power Model Parameter Estimates for the Bladder Tumor Data

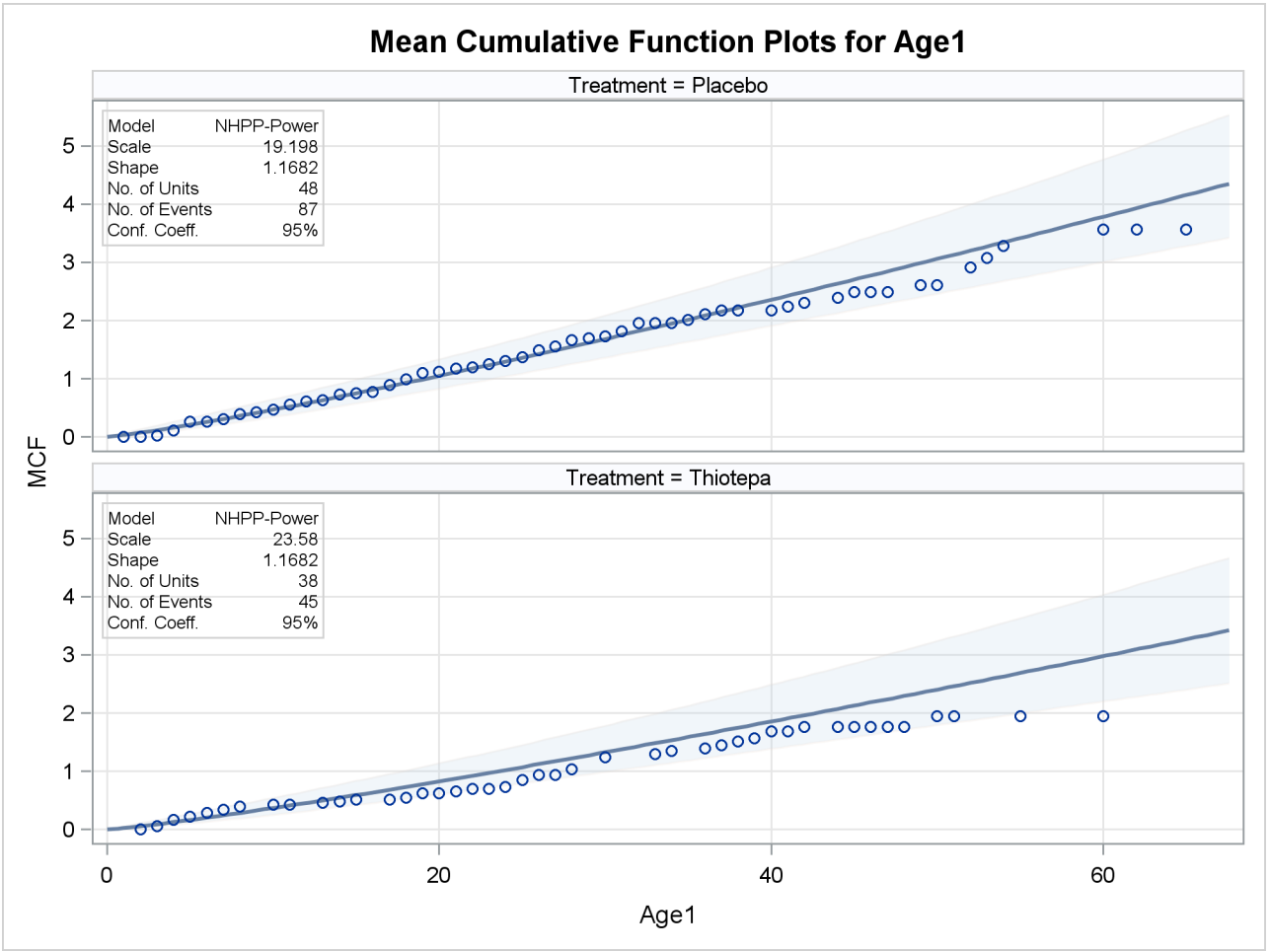
The RELIABILITY Procedure

NHPP-Power Parameter Estimates

Parameter		Estimate	Standard Error	Asymptotic Normal 95% Confidence Limits Lower	Upper
Intercept		23.5802	3.1567	17.3932	29.7671
Group	Placebo	-4.3826	3.4873	-11.2175	2.4523
Group	Thiotepa	0.0000	0.0000	0.0000	0.0000
Shape		1.1682	0.0960	0.9945	1.3723

Nonparametric estimates of the mean cumulative function are plotted as points, and the fitted model is plotted as the solid line in Figure 18.54. Pointwise parametric confidence intervals are plotted by default when the fit=Model option is used.

Figure 18.54 Mean Cumulative Function Plot for the Bladder Tumor Data



Syntax: RELIABILITY Procedure

Primary Statements

The following are the primary statements that control the RELIABILITY procedure:

PROC RELIABILITY < *options* > **;**
< *label:* >**ANALYZE** *variable* < ∗*censor-variable*(*values*) > < =*group-variables* >
< / *options* > **;**
< *label:* >**MCFPLOT** *variable* ∗ *cost/censor-variable*(*values*) < =*group-variables* >
< / *options* > **;**
MODEL *variable* < ∗*censor-variable*(*values*) > < =*independent-variables* >
< / *options* > **;**
< *label:* >**PROBPLOT** *variable* < ∗*censor-variable*(*values*) > < =*group-variables* >
< / *options* > **;**
< *label:* >**RELATIONPLOT** *variable* < ∗*censor-variable*(*values*) > < =*group-variables* >
< / *options* > **;**

The PROC RELIABILITY statement invokes the procedure.

The plot statements (PROBPLOT, RELATIONPLOT, and MCFPLOT) create graphical displays. Each of the plot statements has options that control the content and appearance of the plots they create. The default settings provide the best plots for many purposes; however, if you want to control specific details of the plots, such as axis limits or background colors, then you need to specify the options.

In addition to graphical output, each plot statement provides analysis results in tabular form. The tabular output also can be controlled with statement options.

The MODEL and ANALYZE statements produce only tabular analysis output, not graphical displays.

You can specify one or more of the plot and ANALYZE statements. If you specify more than one MODEL statement, only the last one specified is used.

Secondary Statements

You can specify the following statements in conjunction with the primary statements listed previously. These statements are used to modify the behavior of the primary statements or to specify additional variables.

BY *variables* **;**
CLASS *variables* **;**
DISTRIBUTION *distribution-name* **;**
EFFECTPLOT < *plot-type* < **(***plot-definition-options***)** > > < / *options* > **;**
ESTIMATE < *'label'* > *estimate-specification* < / *options* > **;**
FMODE *keyword* **=** *variable***(***'value1'* . . . *'valuen'***) ;**
FREQ *variable* **;**
INSET *keyword-list* < / *options* > **;**
LSMEANS < *model-effects* > < / *options* > **;**
LSMESTIMATE *model-effect* < *'label'* > *values* < *divisor***=***n* > < **,** . . . < *'label'* > *values* < *divisor***=***n* > >
< / *options* > **;**
MAKE *'table'* **OUT=***SAS-data-set* < *options* > **;**
NENTER *variable* **;**
NLOPTIONS < *options* > **;**
SLICE *model-effect* < / *options* > **;**
STORE < **OUT=** > *item-store-name* < / **LABEL=***'label'* > **;**
TEST < *model-effects* > < / *options* > **;**
UNITID *variable* **;**

The EFFECTPLOT, ESTIMATE, LSMEANS, LSMESTIMATE, SLICE, STORE, and TEST statements are used to provide further analysis of regression models that are fit by using a MODEL statement and are common to many SAS/STAT procedures. Summary descriptions of functionality and syntax for these statements appear after the PROC RELIABILITY statement in alphabetical order, and full documentation about them is available in Chapter 19, "Shared Concepts and Topics" (*SAS/STAT User's Guide*).

You can use the STORE statement to store the results of fitting a regression model with a MODEL statement for later analysis with the SAS/STAT procedure PROC PLM, if you have SAS/STAT software installed at your site.

The BY statement specifies variables in the input data set that are used for BY processing. A separate analysis is performed for each group of observations defined by the levels of the BY variables. The input data set must be sorted in order of the BY variables.

The CLASS statement specifies variables in the input data set that serve as *indicator*, *dummy*, or *classification* variables in the MODEL statement.

The DISTRIBUTION statement specifies a probability distribution name for those statements that require a probability distribution for proper operation (the ANALYZE, PROBPLOT, MODEL, and RELATIONPLOT statements). If you do not specify a distribution with the DISTRIBUTION statement, the normal distribution is used.

The FMODE statement specifies what failure-mode data to include in the analysis of data. Use this statement in conjunction with the ANALYZE, MODEL, PROBPLOT, or RELATIONPLOT statement.

The FREQ statement specifies a variable that provides frequency counts for each observation in the input data set.

The INSET statement specifies what information is printed in the inset box created by the PROBPLOT or MCFPLOT statement. The INSET statement also controls the appearance of the inset box.

The MAKE statement creates a SAS data set from any of the tables produced by the procedure. You specify a table and a SAS data set name for the data set you want to create. There is a unique table name that identifies each table printed; see the tables in the section "MAKE Statement" on page 1291.

The NENTER statement specifies interval-censored data having a special structure; these data are called *readout* data. Use the NENTER statement in conjunction with the FREQ statement.

The NLOPTIONS statement enables you to control aspects of the nonlinear optimizations used for maximum likelihood estimation of the parameters of the three-parameter Weibull distribution in the ANALYZE and PROBPLOT statements, and of parametric models for recurrent events data in the MODEL statement.

The UNITID statement specifies a variable in the input data set that is used to identify each individual unit in an MCFPLOT statement.

Graphical Enhancement Statements

You can use the TITLE, FOOTNOTE, and NOTE statements to enhance printed output. If you are creating plots, you can also use the LEGEND and SYMBOL statements to enhance your plots. For details, see the SAS/GRAPH documentation and the section for the plot statement that you are using.

PROC RELIABILITY Statement

PROC RELIABILITY < *options* > **;**

The PROC RELIABILITY statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*
specifies an input data set

GOUT=*graphics-catalog*
specifies a catalog for saving graphical output

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. The default length is 20 characters.

ANALYZE Statement

< *label:* >**ANALYZE** *variable* < ∗*censor-variable*(*values*) > < =*group-variables* > < / *options* > **;**

< *label:* >**ANALYZE** (*variable1 variable2*) < =*group-variables* > < / *options* > **;**

< *label:* >**ANALYZE** *variable1*(*variable2*) < =*group-variables* > < / *options* > **;**

You use the ANALYZE statement to estimate the parameters of the probability distribution specified in the DISTRIBUTION statement without producing any graphical output. The ANALYZE statement performs the same analysis as the PROBPLOT statement, but it does not produce any plots. In addition, you can use the ANALYZE statement to analyze data with the binomial and Poisson distributions. The third format for the preceding ANALYZE statement applies only to Poisson and binomial data. You can use any number of ANALYZE statements after a PROC RELIABILITY statement; each ANALYZE statement produces a separate analysis. You can specify an optional *label* to distinguish between multiple ANALYZE statements in the output.

You must specify one *variable*. If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values.

If you are using the binomial or Poisson distributions, you must specify *variable1* to represent a binomial or Poisson count and *variable2* to provide an exposure measure for the Poisson distribution or the binomial sample size for the binomial distribution.

You can optionally specify one or two *group-variables*. The ANALYZE statement produces an analysis for each level combination of the *group-variable* values. The observations in a given level are referred to as a *cell*.

The elements of the ANALYZE statement are described as follows.

variable
: represents the data for which an analysis is to be produced. A *variable* must be a numeric variable in the input data set.

censor-variable(values)
: indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you omit the specification of *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(variable1 variable2)
: is another method of specifying the data. You can use this syntax in a situation where uncensored, interval-censored, left-censored, and right-censored values occur in the same set of data. Table 18.31 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values. See the section "Lognormal Analysis with Arbitrary Censoring" on page 1224 for an example of using this syntax to create a probability plot.

variable1
: represents the count data for which a Poisson or binomial analysis is to be produced. A *variable1* must be a numeric variable in the input data set.

variable2
: provides either an exposure measure for a Poisson analysis or a binomial number of trials for a binomial analysis. A *variable2* must be a numeric variable in the input data set.

group-variables
: are one or two group variables. If no group variables are specified, a single analysis is produced. The *group-variables* can be numeric or character variables in the input data set.

  Note that the parentheses surrounding the textitgroup-variables are needed only if two group variables are specified.

options
: control the features of the analysis. All *options* are specified after a slash (/) in the ANALYZE statement.

Summary of Options

The following tables summarize the options available in the ANALYZE statement. You can specify one or more of these options to control the parameter estimation and provide optional analyses.

Table 18.4 Analysis Options for Distributions Other Than Poisson or Binomial

Option	Option Description
CONFIDENCE=*number*	Specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95.
CONVERGE=*number*	Specifies the convergence criterion for maximum likelihood fit. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CONVH=*number*	Specifies the convergence criterion for the relative Hessian convergence criterion. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CORRB	Requests the parameter correlation matrix.
COVB	Requests the parameter covariance matrix.
FITTYPE | FIT= *fit-specification*	Specifies the method of estimating distribution parameters. The available *fit-specifications* and their meanings are shown in the following table.

Fit Specification	Definition
LSYX	Least squares fit to the probability plot. The probability axis is the dependent variable.
LSXY	Least squares fit to the probability plot. The lifetime axis is the dependent variable.
MLE	Maximum likelihood (default).
NONE	No fit is computed.
WEIBAYES <(CONFIDENCE | CONF=*number*)>	Weibayes fit. *number* is the confidence coefficient for the Weibayes fit and is between 0 and 1. The default is 0.95.

Table 18.4 Analysis Options for Distributions Other Than Poisson or Binomial (continued)

Option	Option Description
INEST | IN=*SAS-data-set*	Specifies a SAS data set that can contain initial values, equality constraints, upper bounds, or lower bounds for the scale, shape, and threshold parameters in a three-parameter Weibull model for lifetime data. Applies only to three-parameter Weibull models. See the section "INEST Data Set for the Three-Parameter Weibull" on page 1356 for details.
ITPRINT	Requests the iteration history for maximum likelihood fit.
ITPRINTEM	Requests the iteration history for the Turnbull algorithm.
LRCL	Requests likelihood ratio confidence intervals for distribution parameters.
LRCLPER	Requests likelihood ratio confidence intervals for distribution percentiles.
LRCLSURV	Requests likelihood ratio confidence intervals for survival and cumulative distribution functions at times specified with the SURVTIME=*number-list* option.
LOCATION=*number* < LINIT >	Specifies fixed or initial value of location parameter.
MAKEHAM=*number* < MKINIT >	Specifies the fixed or initial value of the Makeham parameter for the three-parameter Gompertz distribution.
MAXIT=*number*	Specifies the maximum number of iterations allowed for maximum likelihood fit.
MAXITEREM | MAXITEM=*number1* < ,*number2* >	*number1* specifies the maximum number of iterations allowed for Turnbull algorithm. Iteration history is printed in increments of *number2* if requested with ITPRINTEM. See the section "Interval-Censored Data" on page 1347 for details.
NOPCTILES	Suppresses computation of percentiles.
NOPOLISH	Suppresses the setting of small interval probabilities to 0 in the Turnbull algorithm. See the section "Interval-Censored Data" on page 1347 for details.

Table 18.4 Analysis Options for Distributions Other Than Poisson or Binomial (continued)

Option	Option Description
PCTLIST=*number-list*	Specifies a list of percentages for which to compute percentile estimates. *number-list* must be a list of numbers separated by blanks or commas. Each number in the list must be between 0 and 100. If this option is not specified, percentiles are computed for a standard list of percentages.
PPOS=*plotting-position*	Specifies the *plotting-position* type used to compute nonparametric estimates of the probability distribution function. See the section "Probability Plotting" on page 1343 for details. The available *plotting-position* types are shown in the following table.

Plotting Position	Type
EXPRANK	Expected ranks
MEDRANK	Median ranks
MEDRANK1	Median ranks (exact formula)
KM	Kaplan-Meier
MKM	Modified Kaplan-Meier (default)
NA | NELSONAALEN	Nelson-Aalen

PPOUT	Requests a table of cumulative probabilities.
PRINTPROBS	Print intervals and associated probabilities for the Turnbull algorithm.
PROBLIST=*number-list*	Specifies a list of initial values for Turnbull algorithm. See the section "Interval-Censored Data" on page 1347 for details.
PSTABLE=*number*	Specifies stable parameterization. The *number* must be between 0 and 1. See the section "Stable Parameters" on page 1359 for further information.
READOUT	Analyzes readout data.
SCALE=*number* < SCINIT >	Specifies the fixed or initial value of scale parameter.
SHAPE=*number* < SHINIT >	Specifies the fixed or initial value of shape parameter.
SINGULAR=*number*	Specifies the singularity criterion for matrix inversion.

Table 18.4 Analysis Options for Distributions Other Than Poisson or Binomial (continued)

Option	Option Description
SURVTIME=*number-list*	Requests that the survival function, cumulative distribution function, and confidence limits be computed for values in *number-list*. See the section "Reliability Function" on page 1364 for details.
THRESHOLD=*number*	Specifies a fixed threshold parameter. See Table 18.57 for the distributions with a threshold parameter.
TOLLIKE=*number*	Specifies the criterion for convergence in the Turnbull algorithm. The default is 10^{-8}. See the section "Interval-Censored Data" on page 1347 for details.
TOLPROB=*number*	Specifies the criterion for setting interval probability to 0 in the Turnbull algorithm. Default is 10^{-6}. See the section "Interval-Censored Data" on page 1347 for details.
WALDCL | NORMALCL	Requests Wald type confidence intervals for distribution parameters. See Table 18.68 and Table 18.74 for details about the computation of Wald confidence intervals. Wald confidence intervals are provided by default, but this option can be combined with LRCL to obtain both types of intervals.

Table 18.5 Analysis Options for Poisson and Binomial Distributions

Option	Option Description
CONFIDENCE=*number*	Specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95.
PREDICT(*number*)	Requests predicted counts for exposure *number* for Poisson or sample size *number* for binomial.
TOLERANCE(*number*)	Requests exposure for Poisson or sample size for binomial to estimate Poisson rate or binomial probability within *number* with probability given by the CONFIDENCE= option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC RELIABILITY to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the RELIABILITY procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable-names*< */ options*> **;**

The CLASS statement specifies variables in the input data set that serve as *indicator*, *dummy*, or *classification* variables in the MODEL statement. If a CLASS variable is specified as an independent variable in the MODEL statement, the RELIABILITY procedure automatically generates an indicator variable for each level of the CLASS variable. The indicator variables generated are used as independent variables in the regression model specified in the MODEL statement. An indicator variable for a level of a CLASS variable is a variable equal to 1 for those observations corresponding to the level and equal to 0 for all other observations.

You can specify the following option in the CLASS statement.

TRUNCATE< *=n*>
specifies the length *n* of CLASS variable values to use in determining CLASS variable levels. If you specify TRUNCATE without the length *n*, the first 16 characters of the formatted values are used. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases before SAS 9. The default is to use the full formatted length of the CLASS variable.

DISTRIBUTION Statement

DISTRIBUTION *probability distribution-name* **;**

The ANALYZE, PROBPLOT, RELATIONPLOT, and MODEL statements require you to specify the probability distribution that describes your data. You can specify a probability distribution for lifetime data by using the DISTRIBUTION statement anywhere after the PROC RELIABILITY statement and before the RUN statement. If you do not specify a distribution in a DISTRIBUTION statement, the normal distribution is assumed. In addition, you can specify a parametric non-homogeneous Poisson process model for recurrent events data in a DISTRIBUTION statement. The probability distribution for lifetime data or the model for recurrent events data specified determines the distribution for which parameters are estimated from your data. The valid distributions and the statements to which they apply are shown in Table 18.6 and Table 18.7.

Table 18.6 Probability Distributions for Lifetime Data

Distribution	Distribution-Name Specified	Statement
Binomial	BINOMIAL	ANALYZE
Exponential	EXPONENTIAL	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Extreme value	EXTREME | EV	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Generalized gamma	GAMMA	MODEL
Gompertz	GOMPERTZ | G2	ANALYZE, PROBPLOT
Three-parameter Gompertz	GOMPERTZ3 | G3	ANALYZE, PROBPLOT
Logistic	LOGISTIC | LOGIT	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Log-logistic	LLOGISTIC | LLOGIT	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Lognormal (base *e*)	LOGNORMAL | LNORM	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Lognormal (base 10)	LOGNORMAL10 | LNORM10	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Normal	NORMAL	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Poisson	POISSON	ANALYZE
Weibull	WEIBULL | W2	ANALYZE, PROBPLOT, RELATIONPLOT, MODEL
Three-parameter Weibull	WEIBULL3 | W3	ANALYZE, PROBPLOT

Table 18.7 Poisson Process Models for Recurrence Data

NHPP Model	Distribution-Name Specified	Statement
Homogeneous	HPP	MODEL
Crow-AMSAA	NHPP(CA) | NHPP(CROWAMSAA)	MODEL
Log-linear	NHPP(LOG) | NHPP(LOGLINEAR)	MODEL
Power	NHPP(POW) | NHPP(POWER)	MODEL
Proportional intensity	NHPP(PROP) | NHPP(PROPORTIONAL)	MODEL

EFFECTPLOT Statement

EFFECTPLOT < *plot-type* < *(plot-definition-options)* > > < / *options* > **;**

The EFFECTPLOT statement produces a display of the fitted model and provides options for changing and enhancing the displays. Table 18.8 describes the available *plot-types* and their *plot-definition-options*.

Table 18.8 *Plot-Types* and *Plot-Definition-Options*

Plot-Type and Description	Plot-Definition-Options
BOX Displays a box plot of continuous response data at each level of a CLASS effect, with predicted values superimposed and connected by a line. This is an alternative to the INTERACTION *plot-type*.	PLOTBY= variable or CLASS effect X= CLASS variable or effect
CONTOUR Displays a contour plot of predicted values against two continuous covariates.	PLOTBY= variable or CLASS effect X= continuous variable Y= continuous variable
FIT Displays a curve of predicted values versus a continuous variable.	PLOTBY= variable or CLASS effect X= continuous variable
INTERACTION Displays a plot of predicted values (possibly with error bars) versus the levels of a CLASS effect. The predicted values are connected with lines and can be grouped by the levels of another CLASS effect.	PLOTBY= variable or CLASS effect SLICEBY= variable or CLASS effect X= CLASS variable or effect
MOSAIC Displays a mosaic plot of predicted values using up to three CLASS effects.	PLOTBY= variable or CLASS effect X= CLASS effects
SLICEFIT Displays a curve of predicted values versus a continuous variable grouped by the levels of a CLASS effect.	PLOTBY= variable or CLASS effect SLICEBY= variable or CLASS effect X= continuous variable

For full details about the syntax and options of the EFFECTPLOT statement, see the section "EFFECTPLOT Statement" (Chapter 19, *SAS/STAT User's Guide*) in Chapter 19, "Shared Concepts and Topics" (*SAS/STAT User's Guide*).

ESTIMATE Statement

ESTIMATE < '*label*' > *estimate-specification* < **(***divisor***=***n***)** >
< **,** . . . < '*label*' > *estimate-specification* < **(***divisor***=***n***)** > >
< **/** *options* > **;**

The ESTIMATE statement provides a mechanism for obtaining custom hypothesis tests. Estimates are formed as linear estimable functions of the form $\mathbf{L}\boldsymbol{\beta}$. You can perform hypothesis tests for the estimable functions, construct confidence limits, and obtain specific nonlinear transformations.

Table 18.9 summarizes the *options* available in the ESTIMATE statement.

Table 18.9 ESTIMATE Statement Options

Option	Description
Construction and Computation of Estimable Functions	
DIVISOR=	Specifies a list of values to divide the coefficients
NOFILL	Suppresses the automatic fill-in of coefficients for higher-order effects
SINGULAR=	Tunes the estimability checking difference
Degrees of Freedom and *p*-values	
ADJUST=	Determines the method for multiple comparison adjustment of estimates
ALPHA=α	Determines the confidence level $(1-\alpha)$
LOWER	Performs one-sided, lower-tailed inference
STEPDOWN	Adjusts multiplicity-corrected *p*-values further in a step-down fashion
TESTVALUE=	Specifies values under the null hypothesis for tests
UPPER	Performs one-sided, upper-tailed inference
Statistical Output	
CL	Constructs confidence limits
CORR	Displays the correlation matrix of estimates
COV	Displays the covariance matrix of estimates
E	Prints the **L** matrix
JOINT	Produces a joint *F* or chi-square test for the estimable functions
PLOTS=	Requests ODS statistical graphics if the analysis is sampling-based
SEED=	Specifies the seed for computations that depend on random numbers
Generalized Linear Modeling	
CATEGORY=	Specifies how to construct estimable functions with multinomial data

Table 18.9 *continued*

Option	Description
EXP	Exponentiates and displays estimates
ILINK	Computes and displays estimates and standard errors on the inverse linked scale

For details about the syntax of the ESTIMATE statement, see the section "ESTIMATE Statement" (Chapter 19, *SAS/STAT User's Guide*) in Chapter 19, "Shared Concepts and Topics" (*SAS/STAT User's Guide*).

FMODE Statement

FMODE keyword=*variable ('value1' . . . 'valuen')* < */ options* > **;**

Use the FMODE statement with data that have failures attributable to multiple causes (*failure modes*). You can analyze data by either keeping, eliminating, or combining specific failure modes with the FMODE statement. Use this statement with the KEEP or ELIMINATE keyword in conjunction with the ANALYZE, MODEL, PROBPLOT, or RELATIONPLOT statement. Use this statement with the COMBINE keyword with the ANALYZE or PROBPLOT statement. You can place an FMODE statement anywhere after the PROC RELIABILITY statement and before the RUN statement.

If you specify the keyword KEEP, the life distribution for only the identified failure modes is estimated, with all other failure modes treated as right-censored data. If you specify the keyword ELIMINATE, the life distribution that results if the failure modes identified are completely eliminated is estimated. The keyword ELIMINATE causes the failure modes identified to be treated as right-censored data and causes a single life distribution to be estimated for the remaining data. If you specify the keyword COMBINE, the data are analyzed with all the specified failure modes combined acting. See the section "Weibull Probability Plot for Two Combined Failure Modes" on page 1241 for an example of a Weibull plot of data with two combined failure modes. The failure mode for an observation in the input data set is identified by the value of *variable*, where *variable* is any numeric or character variable in the input data set. You must identify a failure mode for each observation that is not right-censored. You specify failure modes to keep, eliminate, or combine by listing variable values *(value1 . . . valuen)* in parentheses after the failure mode variable name. The list of variable values must have entries separated by blanks or commas. You can specify the following *options* after the slash (/). These options will affect the analysis only when you use the COMBINE keyword.

Table 18.10 FMODE Statement Options

Option	Description
LEGEND=	Specifies a LEGEND statement for individual mode fit lines.
NOLEGEND	Suppresses legend for individual mode fit lines.
PLOTMODES	Plots individual failure distribution lines on probability plot.

FREQ Statement

FREQ *variable-name* **;**

The FREQ statement specifies a variable that provides frequency counts for each observation in the input data set. If *n* is the value of the FREQ variable in the input data set for an observation, then that observation is weighted by *n*. The log-likelihood function for maximum likelihood estimation is multiplied by *n*. If *n* is not an integer, the integer part of *n* is used in creating probability plots.

You can also use the FREQ statement in conjunction with the NENTER statement to specify interval-censored data having a special structure; these data are called *readout* data. The FREQ statement specifies a variable in the input data set that determines the number of units failing in each interval. See the section "Weibull Analysis of Interval Data with Common Inspection Schedule" on page 1219 for an example that uses the FREQ statement with readout data.

You can also use the FREQ statement in conjunction with the NENTER statement to specify recurrent events data when the event times are grouped into intervals, rather than being observed exactly. The FREQ statement specifies a variable in the input data set that determines the number of events in each interval.

You can use the FREQ statement with the MCFPLOT and MODEL statements for exact age data to provide frequency counts for entire recurrence histories. If *n* is the value of the FREQ variable at a censor time, the history of recurrences for the corresponding system is replicated independently *n* times. Values of the FREQ variable at times other than censor times are not used; they can be any value or missing without affecting the analysis.

INSET Statement

INSET *keyword-list* < */ options* > **;**

The box or table of summary information produced on plots made with the PROBPLOT or MCFPLOT statement is called an *inset*. You can use the INSET statement to customize the appearance of the inset box and the information that is printed in the inset box. To supply the information that is displayed in the inset box, you specify a *keyword* that corresponds to the information you want shown. For example, the following statements produce a Weibull plot with the sample size, the number of failures, and the Weibull mean displayed in the inset:

```
proc reliability data=fan;
   distribution Weibull;
   pplot lifetime*censor(1);
   inset n nfail weibull(mean);
run;
```

By default, inset entries are identified with appropriate labels. However, you can provide a customized label by specifying the *keyword* for that entry followed by the equal sign (=) and the label in quotes. For example, the following INSET statement produces an inset that contains the sample size and Weibull mean, labeled "Sample Size" and "Weibull Mean" in the inset:

```
inset n='Sample Size' weibull(mean='Weibull Mean');
```

If you specify a keyword that does not apply to the plot you are creating, then the keyword is ignored.

The *options* control the appearance of the box.

If you specify more than one INSET statement, only the last one is used.

Keywords Used in the INSET Statement

Table 18.11 through Table 18.13 list keywords available in the INSET statement to display summary statistics, distribution parameters, and distribution fitting information.

Table 18.11 Summary Statistics

Keyword	Description
N	Sample size
NFAIL	Number of failures for probability plots.
NEVENTS	Number of events or repairs for MCF plots.
NEVENTS1	Number of events or repairs in the first group for MCF difference plots.
NEVENTS2	Number of events or repairs in the second group for MCF difference plots.
NUNITS	Number of units or systems for MCF plots.
NUNITS1	Number of units or systems in the first group for MCF difference plots.
NUNITS2	Number of units or systems in the second group for MCF difference plots.

Table 18.12 General Information

Keyword	Description
CONFIDENCE	Confidence coefficient for all confidence intervals or for the Weibayes fit
FIT	Method used to estimate distribution parameters for probability plots
RSQUARE	R square for least squares distribution fit to probability plots

Distribution parameters are specified as *distribution-name*(*distribution-parameters*). The following table lists the keywords available.

Table 18.13 Distribution Parameters

Keyword	Secondary Keyword	Description
EXPONENTIAL	SCALE	Scale parameter
	THRESHOLD	Threshold parameter
	MEAN	Expected value
EXTREME | EV	LOCATION	Location parameter
	SCALE	Scale parameter
	MEAN	Expected value
GOMPERTZ | GOMP	SCALE	Scale parameter
	SHAPE	Shape parameter
	MEAN	Expected value
GOMPERTZ3 | GOMP3	SCALE	Scale parameter
	SHAPE	Shape parameter
	MAKEHAM	Makeham mortality component
	MEAN	Expected value
LOGISTIC | LOGIT	LOCATION	Location parameter
	SCALE	Scale parameter
	MEAN	Expected value
LOGLOGISTIC | LLOGIT	LOCATION	Location parameter
	SCALE	Scale parameter
	THRESHOLD	Threshold parameter
	MEAN	Expected value
LOGNORMAL	LOCATION	Location parameter
	SCALE	Scale parameter
	THRESHOLD	Threshold parameter
	MEAN	Expected value
LOGNORMAL10	LOCATION	Location parameter
	SCALE	Scale parameter
	THRESHOLD	Threshold parameter
	MEAN	Expected value
NORMAL	LOCATION	Location parameter
	SCALE	Scale parameter
	MEAN	Expected value
WEIBULL	SCALE	Scale parameter
	SHAPE	Shape parameter
	THRESHOLD	Threshold parameter
	MEAN	Expected value

Options Used in the INSET Statement

Table 18.14 through Table 18.17 list INSET statement options that control the appearance of the inset box.

Table 18.14 lists options that control the appearance of the box when you use traditional graphics.

Table 18.14 General Appearance Options (Traditional Graphics)

Option	Option Description
HEADER=*'quoted-string'*	Specifies text for header or box title.
NOFRAME	Omits frame around box.
POS=*value* < DATA | PERCENT >	Determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW) or a pair of coordinates (x, y) enclosed in parentheses. The coordinates can be specified in axis percent units or axis data units.
REFPOINT=*name*	Specifies the reference point for an inset that is positioned by a pair of coordinates with the POS= option. You use the REFPOINT= option in conjunction with the POS= coordinates. *name* specifies which corner of the inset frame you have specified with coordinates (x, y); it can take the value of BR (bottom right), BL (bottom left), TR (top right), or TL (top left). The default is REFPOINT=BL. If the inset position is specified as a compass point, then the REFPOINT= option is ignored.

Table 18.15 lists options that control the appearance of the box when you use ODS Graphics.

Table 18.15 General Appearance Options (ODS Graphics)

Option	Option Description
HEADER=*'quoted-string'*	Specifies text for header or box title.
NOFRAME	Omits frame around box.
POS=*value*	Determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW).

Table 18.16 lists options that control the appearance of the text within the box when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.16 Text Enhancement Options (Traditional Graphics)

Option	Option Description
FONT=*font*	Software font for text
HEIGHT=*value*	Height of text

Table 18.17 lists options that control the colors and patterns used in the box when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.17 Color and Pattern Options (Traditional Graphics)

Option	Option Description
CFILL=*color*	Color for filling box
CFILLH=*color*	Color for filling box header
CFRAME=*color*	Color for frame
CHEADER=*color*	Color for text in header
CTEXT=*color*	Color for text

LOGSCALE Statement

LOGSCALE *effect-list* < */options* > **;**

You use the LOGSCALE statement to model the logarithm of the distribution scale parameter as a function of explanatory variables. A MODEL statement must also be present to specify the model for the distribution location parameter. *effect-list* is a list of variables in the input data set representing the values of the independent variables in the model for each observation, and combinations of variables representing interaction terms. It can contain any variables or combination of variables in the input data set. It can contain the same variables as the MODEL statement, or it can contain different variables. The variables in the *effect-list* can be any combination of indicator variables named in a CLASS statement and continuous variables. The coefficients of the explanatory variables are estimated by maximum likelihood.

Table 18.18 lists the *options* available for the LOGSCALE statement.

Table 18.18 LOGSCALE Statement Options

Option	Option Description
INITIAL=*number-list*	Specifies initial values for log-scale regression parameters other than the intercept term.
INTERCEPT=*number* < INTINIT >	Specifies an initial or fixed value of the intercept parameter, depending on whether INTINIT is present.

LSMEANS Statement

LSMEANS < *model-effects* > < / *options* > **;**

The LSMEANS statement computes and compares least squares means (LS-means) of fixed effects. LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs.

Table 18.19 summarizes the *options* available in the LSMEANS statement.

Table 18.19 LSMEANS Statement Options

Option	Description
Construction and Computation of LS-Means	
AT	Modifies the covariate value in computing LS-means
BYLEVEL	Computes separate margins
DIFF	Requests differences of LS-means
OM=	Specifies the weighting scheme for LS-means computation as determined by the input data set
SINGULAR=	Tunes estimability checking
Degrees of Freedom and *p*-values	
ADJUST=	Determines the method for multiple-comparison adjustment of LS-means differences
ALPHA=α	Determines the confidence level $(1 - \alpha)$
STEPDOWN	Adjusts multiple-comparison *p*-values further in a step-down fashion
Statistical Output	
CL	Constructs confidence limits for means and mean differences
CORR	Displays the correlation matrix of LS-means
COV	Displays the covariance matrix of LS-means
E	Prints the $\mathbf{L}$ matrix
LINES	Produces a "Lines" display for pairwise LS-means differences
MEANS	Prints the LS-means
PLOTS=	Requests graphs of means and mean comparisons
SEED=	Specifies the seed for computations that depend on random numbers
Generalized Linear Modeling	
EXP	Exponentiates and displays estimates of LS-means or LS-means differences
ILINK	Computes and displays estimates and standard errors of LS-means (but not differences) on the inverse linked scale
ODDSRATIO	Reports (simple) differences of least squares means in terms of odds ratios if permitted by the link function

For details about the syntax of the LSMEANS statement, see the section "LSMEANS Statement" (Chapter 19, *SAS/STAT User's Guide*).

LSMESTIMATE Statement

LSMESTIMATE *model-effect* < '*label*' > *values* < *divisor*=*n* >
< , . . . < '*label*' > *values* < *divisor*=*n* > >
< / *options* > ;

The LSMESTIMATE statement provides a mechanism for obtaining custom hypothesis tests among least squares means.

Table 18.20 summarizes the *options* available in the LSMESTIMATE statement.

Table 18.20 LSMESTIMATE Statement Options

Option	Description
Construction and Computation of LS-Means	
AT	Modifies covariate values in computing LS-means
BYLEVEL	Computes separate margins
DIVISOR=	Specifies a list of values to divide the coefficients
OM=	Specifies the weighting scheme for LS-means computation as determined by a data set
SINGULAR=	Tunes estimability checking
Degrees of Freedom and *p*-values	
ADJUST=	Determines the method for multiple-comparison adjustment of LS-means differences
ALPHA=α	Determines the confidence level $(1 - \alpha)$
LOWER	Performs one-sided, lower-tailed inference
STEPDOWN	Adjusts multiple-comparison p-values further in a step-down fashion
TESTVALUE=	Specifies values under the null hypothesis for tests
UPPER	Performs one-sided, upper-tailed inference
Statistical Output	
CL	Constructs confidence limits for means and mean differences
CORR	Displays the correlation matrix of LS-means
COV	Displays the covariance matrix of LS-means
E	Prints the **L** matrix
ELSM	Prints the **K** matrix
JOINT	Produces a joint F or chi-square test for the LS-means and LS-means differences
PLOTS=	Requests graphs of means and mean comparisons
SEED=	Specifies the seed for computations that depend on random numbers
Generalized Linear Modeling	
CATEGORY=	Specifies how to construct estimable functions with multinomial data
EXP	Exponentiates and displays LS-means estimates

Table 18.20 *continued*

Option	Description
ILINK	Computes and displays estimates and standard errors of LS-means (but not differences) on the inverse linked scale

For details about the syntax of the LSMESTIMATE statement, see the section "LSMESTIMATE Statement" (Chapter 19, *SAS/STAT User's Guide*) in Chapter 19, "Shared Concepts and Topics" (*SAS/STAT User's Guide*).

MAKE Statement

MAKE '*table*' **OUT=***SAS-data-set*< *SAS-data-set options* > **;**

The MAKE statement creates a SAS data set from any of the tables produced by the RELIABILITY procedure. You can specify SAS data set options in parentheses after the data set name. You can specify one MAKE statement for each table that you want to save to a SAS data set.

The ODS statement also creates SAS data sets from tables, in addition to providing an extensive and flexible method of controlling output created by the RELIABILITY procedure. The ODS statement is the recommended method of controlling procedure output; however, the MAKE statement is provided for compatibility with earlier releases of the SAS System.

The valid values for *table* are shown in the section "ODS Table Names" on page 1388, organized by the RELIABILITY procedure statement that produces the tabular output. The *table* names are not case sensitive, but they must be enclosed in single quotes.

MCFPLOT Statement

< *label:* >**MCFPLOT** *variable*∗*cost/censor-variable*(*values*) < =*group-variables* > < / *options* > **;**

< *label:* >**MCFPLOT** (< **INTERVAL=** >*variable1 variable2* < **RECURRENCES=** > *variable3* < **CENSOR=** > *variable4*) < =*group-variables* > < / *options* > **;**

< *label:* >**MCFPLOT** (*variable1 variable2*) < =*group-variables* > < / *options* > **;**

You can specify any number of MCFPLOT statements after a PROC RELIABILITY statement. Each MCFPLOT statement creates a separate MCF plot and associated analysis. See the section "Analysis of Recurrence Data on Repairs" on page 1245, the section "Comparison of Two Samples of Repair Data" on page 1250, and the section "Analysis of Interval Age Recurrence Data" on page 1257 for examples that use the MCFPLOT statement. You can specify an optional *label* to distinguish between multiple MCFPLOT statements in the output.

To create a plot of the mean cumulative function for cost or number of repairs with exact age data, you specify a *variable* that represents the times of repairs. You must also specify a *cost/censor-variable* and the *values*, in parentheses, of the *cost/censor-variable* that correspond to end-of-history data values (also referred to as *censored* data values).

To create a plot of the mean cumulative function for cost or number of repairs with interval age data, you specify *variable1 variable2* that represents the age intervals. You must also specify either *variable3* that represents the number of recurrences in the intervals and *variable4* that represents the number censored in the intervals, or a FREQ statement that represents the number of recurrent events in the intervals and a NENTER statement that represents the number of units observed in the intervals.

You can optionally specify one or two *group-variables* (also referred to as *classification variables*). The MCFPLOT statement displays a component plot for each level of the *group-variables*. The observations in a given level are called a *cell*.

For exact data, you must also specify a *unit-identification* variable in conjunction with the MCFPLOT statement to identify the individual unit name for each instance of repair or end of history on the unit. Specify the *unit-identification* variable in the UNITID statement.

Add the EVENTPLOT option to any MCFPLOT statement to obtain a horizontal plot of failure and censoring times for each system.

The elements of the MCFPLOT statement are described as follows.

variable
: represents the time of repair. A *variable* must be a numeric variable in the input data set.

variable1 variable2
: represents time intervals for grouped data. *variable1* and *variable2* must be numeric variables in the input data set.

variable3
: represents the number of recurrences in an interval. A *variable3* must be a numeric variable in the input data set.

variable4
: represents the number censored in an interval. A *variable4* must be a numeric variable in the input data set.

cost/censor-variable(values)
: indicates the cost of each repair or the number of repairs. This variable also indicates which observations in the input data set are end-of-history (censored) data points. You specify the values of *cost/censor-variable* that represent censored observations by placing those values in parentheses after the variable name. A *cost/censor-variable* must be a numeric variable in the input data set.

group-variables
: are one or two group variables. If no group variables are specified, a single plot is produced. The *group-variables* can be any numeric or character variables in the input data set. For exact data, if a single group variable is specified, and the group variable has two levels, then a statistical test for equality of the groups represented by the two levels is computed and displayed in the "Tests for Equality of Mean Functions" table. Refer to "Comparison of Two Groups of Recurrent Events Data" on page 1381 for more details.

  Note that the parentheses surrounding the *group-variables* are needed only if two group variables are specified.

options

control the features of the mean cumulative function plot. All *options* are specified after a slash (/) in the MCFPLOT statement. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

Table 18.21 lists available analysis options.

Table 18.21 Analysis Options

Option	Option Description
CONFIDENCE=*number*	Specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95.
EVENTPLOT <(SORT=*sort-order*)>	Specifies a separate horizontal plot of failure and censoring times for each system. The following sort orders are available:

Sort Order	Definition
ASCENDINGTIME	sorts by increasing censoring times (default)
DESCENDINGTIME	sorts by decreasing censoring times
ASCENDINGFORMATTED	sorts alphabetically by system name or label
DESCENDINGFORMATTED	sorts alphabetically in reverse by system name or label

INDINC	Requests variance estimates of the MCF using the Nelson (2002) estimator under the independent increments assumption.
LOGINTERVALS	Requests that confidence intervals be computed based on the asymptotic normality of log(MCF). This is appropriate only when the MCF estimate is positive, so does not apply to MCF differences or when negative costs are specified.
MCFDIFF	Requests a plot of differences of MCFS of two groups specified by a single group variable.
NOVARIANCE	Suppresses MCF variance computation.
VARIANCE=*variance-specification*	Specifies the method of variance calculation. The following methods are available.

Table 18.21 Analysis Options (continued)

Option	Option Description

Method	Definition
INDINC	The method of Nelson (2002) assuming independent increments
LAWLESS | VARMETHOD2	The method of Lawless and Nadeau (1995) (the default method)
NELSON	The method of Nelson (2002)
POISSON	Poisson process method

Table 18.22 lists plot layout options that are available when you use traditional graphics.

Table 18.22 Plot Layout Options for Traditional Graphics

Option	Option Description
CENBIN	Plots censored data as frequency counts rather than as individual points.
CENSYMBOL=*symbol* | (*symbol-list*)	Specifies symbols for censored values. *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z). If you are creating overlaid plots for groups of data, you can specify different symbols for the groups with a list of symbols or letters, separated by blanks, enclosed in parentheses. If no CENSYMBOL option is specified, the symbol used for censored values is the same as for repairs.
HOFFSET=*value*	Specifies offset for horizontal axis.
INBORDER	Requests a border around MCF plots.
INTERPOLATE=JOIN | STEP | NONE	Requests that symbols in an MCF plot be connected with a straight line, step function, or not connected.
INTERTILE=*value*	Specifies the distance between tiles.
MCFLEGEND=*legend-statement-name* | NONE	Identifies a legend statement to specify legend for overlaid MCF plots.
MISSING1	Requests that missing values of first GROUP= variable be treated as a level of the variable.
MISSING2	Requests that missing values of second GROUP= variable be treated as a level of the variable.

Table 18.22 Plot Layout Options (continued)

Option	Option Description
NCOLS=*n*	Specifies the number of columns plotted on a page.
NOCENPLOT	Suppresses plotting of censored data points.
NOCONF	Suppresses plotting of confidence intervals.
NOFRAME	Suppresses the frame around the plotting area.
NOINSET	Suppresses the inset.
NOLEGEND	Suppresses the legend for overlaid MCF plots.
NROWS=*n*	Specifies the number of rows plotted on a page.
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	Specifies display order for values of the first GROUP= variable.
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	Specifies display order for values of the second GROUP= variable.
OVERLAY	Requests that plots with group variables be overlaid on a single page.
PLOTSYMBOL=*symbol* | (*symbol-list*)	Specifies symbols that represent events in an MCF plot.
PLOTCOLOR=*color* | (*color-list*)	Specifies colors of symbols that represent events in an MCF plot.
TURNVLABELS	Vertically strings out characters in labels for vertical axis.
VOFFSET=*value*	Specifies length of offset at upper end of vertical axis.

Table 18.23 lists plot layout options available when you use ODS Graphics.

Table 18.23 Plot Layout Options for ODS Graphics

Option	Option Description
DUANE	Requests that a Duane plot be created in addition to an MCF plot. If you specify the FIT=MODEL option, the fitted parametric model is included in the Duane plot. See the section "Duane Plots" on page 1387 for a description of Duane plots.

Table 18.23 Plot Layout Options (continued)

Option	Option Description
FIT=MODEL	Requests that a parametric cumulative mean function fit with a MODEL statement be plotted on the same plot with nonparametric estimates of the MCF. This option is valid only if the response specification in the MCFPLOT statement matches the response specification in the MODEL statement. If this option is specified, the fit parametric intensity function is plotted on a separate graph.
INTERPOLATE=JOIN | STEP | NONE	Requests that symbols in an MCF plot be connected with a straight line, step function, or not connected.
MISSING1	Requests that missing values of first GROUP= variable be treated as a level of the variable.
MISSING2	Requests that missing values of second GROUP= variable be treated as a level of the variable.
NCOLS=*n*	Specifies the number of columns plotted on a page.
NOCENPLOT	Suppresses plotting of censored data points.
NOCONF	Suppresses plotting of confidence intervals.
NOINSET	Suppresses the inset.
NROWS=*n*	Specifies the number of rows plotted on a page.
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	Specifies the display order for values of the first GROUP= variable.
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	Specifies the display order for values of the second GROUP= variable.
OVERLAY	Requests that plots with group variables be overlaid on a single page.

Table 18.24 lists reference line options that are available when you use traditional graphics.

Table 18.24 Reference Line Options for Traditional Graphics

Option	Option Description
HREF=*value-list*	Specifies reference lines perpendicular to horizontal axis.
HREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for HREF= lines.

Table 18.24 Reference Line Options (continued)

Option	Option Description
HREFLABPOS=*n*	Specifies the vertical position of labels for HREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table: *n* **Label Placement** 1 Top 2 Staggered from top 3 Bottom 4 Staggered from bottom 5 Alternating from top 6 Alternating from bottom
LHREF=*linetype*	Specifies the line style for HREF= lines.
LVREF=*linetype*	Specifies the line style for VREF= lines.
VREF=*value-list*	Specifies reference lines perpendicular to vertical axis.
VREFLABELS=*('label1' ... 'labeln')*	Specifies labels for VREF= lines.
VREFLABPOS=*n*	Specifies the horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table: *n* **Label Placement** 1 Left 2 Right

Table 18.25 lists reference line options that are available when you use ODS Graphics.

Table 18.25 Reference Line Options for ODS Graphics

Option	Option Description
LREF<(INTERSECT)>=*value-list*	Specifies reference lines perpendicular to the lifetime axis. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the MCF axis and intersects the fit line at the same point as the lifetime axis reference line. If a lifetime axis reference line label is specified, the intersecting MCF axis reference line is labeled with the MCF axis value.
LREFLABELS=*('label1' ... 'labeln')*	Specifies labels for LREF= lines.

Table 18.25 Reference Line Options (continued)

Option	Option Description
MREF< (INTERSECT) >=*value-list*	Specifies reference lines perpendicular to the MCF axis. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the lifetime axis and intersects the fit line at the same point as the MCF axis reference line. If an MCF axis reference line label is specified, the intersecting lifetime axis reference line is labeled with the lifetime axis value.
MREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for MREF= lines.

Table 18.26 lists the options that control the appearance of the text when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.26 Text Enhancement Options

Option	Option Description
FONT=*font*	Software font for text.
HEIGHT=*value*	Height of text used outside framed areas.
INFONT=*font*	Software font for text inside framed areas.
INHEIGHT=*value*	Height of text inside framed areas.

Table 18.27 lists options to control the appearance of the axes when you use traditional graphics.

Table 18.27 Axis Options for Traditional Graphics

Option	Option Description

HAXIS=*value1* TO *value2*< BY *value3* >	Specifies tick mark values for the horizontal axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of HAXIS= lists follow:
	`haxis = 0 to 10`
	`haxis = 2 to 10 by 2`
	`haxis = 0 to 200 by 10`

Table 18.27 Axis Options (continued)

Option	Option Description
HLOWER=*number*	Specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *number* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS option is used.
HUPPER=*number*	Specifies the upper limit on the horizontal axis scale. The HUPPER= option specifies *number* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.
LGRID=*number*	Specifies a line style for all grid lines. *number* is between 1 and 46 and specifies a linestyle for grids.
LOGLOG	Requests log scales on both axes.
MINORLOGGRID	Adds a minor grid for log axes.
NOGRID	Suppresses grid lines.
NOHLABEL	Suppresses label for horizontal axis.
NOVLABEL	Suppresses label for vertical axis.
NOVTICK	Suppresses tick marks and tick mark labels for vertical axis.
NOHTICK	Suppresses tick marks and tick mark labels for horizontal axis.
NHTICK=*number*	Specifies the number of tick intervals for the horizontal axis. This option has no effect if the HAXIS= option is used.
NVTICK=*number*	Specifies the number of tick intervals for the vertical axis. This option has no effect if the VAXIS= option is used.

Table 18.27 Axis Options (continued)

Option	Option Description

VAXIS=*value1* TO *value2*< BY *value3* >	Specifies tick mark values for the vertical axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.
	`vaxis = 0 to 10`
	`vaxis = 0 to 2 by .1`

VAXISLABEL='*string*'	Specifies a label for the vertical axis
VLOWER=*number*	Specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *number* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.
VUPPER=*number*	Specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *number* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.
WAXIS=*n*	Specifies the line thickness for axes and frame.

Table 18.28 lists options that control the appearance of the axes when you use ODS Graphics.

Table 18.28 Axis Options for ODS Graphics

Option	Option Description
HLOWER=*number*	Specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *number* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS option is used.

Table 18.28 Axis Options (continued)

Option	Option Description
HUPPER=*number*	Specifies the upper limit on the horizontal axis scale. The HUPPER= option specifies *number* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.
LOGLOG	Requests log scales on both axes.
MINORLOGGRID	Adds a minor grid for log axes.
NOGRID	Suppresses grid lines.
VLOWER=*number*	Specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *number* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.
VUPPER=*number*	Specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *number* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

Table 18.29 lists options that control colors and patterns used in the graph when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.29 Color and Pattern Options

Option	Option Description
CAXIS=*color*	Color for axis.
CCENSOR=*color*	Color for filling censor plot area.
CENCOLOR=*color*	Color for censor symbol.
CFRAME=*color*	Color for frame.
CFRAMESIDE=*color*	Color for filling frame for row labels.
CFRAMETOP=*color*	Color for filling frame for column labels.
CGRID=*color*	Color for grid lines.
CHREF=*color*	Color for HREF= lines.
CTEXT=*color*	Color for text.
CVREF=*color*	Color for VREF= lines.

Table 18.30 lists options that control the use of a graphics catalog to store graphs if you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.30 Graphics Catalog Options

Option	Option Description
DESCRIPTION='*string*'	Description for graphics catalog member.
NAME='*string*'	Name for plot in graphics catalog.

MODEL Statement

MODEL *variable* < ∗*censor-variable*(*values*) > < **=***effect-list* > < / *options* > **;**

MODEL (*variable1 variable2*) < **=***effect-list* > < / *options* > **;**

You use the MODEL statement to fit regression models, where life is modeled as a function of explanatory variables.

You can use only one MODEL statement after a PROC RELIABILITY statement. If you specify more than one MODEL statement, only the last is used.

The MODEL statement does not produce any plots, but it enables you to analyze more complicated regression models than the ANALYZE, PROBPLOT, or RELATIONPLOT statement does. The probability distribution specified in the DISTRIBUTION statement is used in the analysis. The following are examples of MODEL statements:

```
model time = temp voltage;
model life*censor(1) = voltage width;
```

See the section "Analysis of Accelerated Life Test Data" on page 1214 and the section "Regression Modeling" on page 1229 for examples that use the MODEL statement to fit regression models.

If your data are right censored lifetime data, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values.

If your data are recurrent events data with exact event times, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to the end-of-service times for each unit under observation. In this case, you must also specify a UNITID statement to identify the specific unit that corresponds to each observation.

If your lifetime data contain any interval-censored or left-censored values, you must specify *variable1* and *variable2* in parentheses to provide the endpoints of the interval for each observation.

If your data are recurrent events data, and event times are not known exactly, but are known only to have occurred in intervals, you must specify *variable1* and *variable2* in parentheses to provide the endpoints of the interval for each observation. In this case, you must also specify a variable that determines the number of events observed in each interval with a FREQ statement, and a variable that determines the number of units under observation in each interval with a NENTER statement.

The independent variables in your regression model are specified in the *effect-list*. The *effect-list* is any combination of continuous variables, classification variables, and interaction effects.

See the section "Regression Models" on page 1357 for further information on specifying the independent variables.

The elements of the MODEL statement are described as follows.

variable
is the dependent, or response, variable. The *variable* must be a numeric variable in the input data set.

censor-variable(values)
for lifetime data, indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you can omit the specification of a *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

If your data are recurrent events data and exact event times are known, then you must specify *censor-variable*. If *censor-variable* is equal to one of the *values*, then the value of *variable* is the end of observation time for a unit. Otherwise, you use *censor-variable* to assign a cost to the event that occurs at the value of *variable*. If all events have unit cost, then *censor-variable* should be set to one for all observations that do not correspond to end of observation times. The *censor-variable* plays the same role as the *cost/censor-variable* in the MCFPLOT statement in this case.

(variable1 variable2)
is another method of specifying the dependent variable in a regression model for lifetime data. You can use this syntax in a situation where uncensored, interval-censored, left-censored, and right-censored values occur in the same set of data. Table 18.31 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values.

Table 18.31 Specifying Censored Values

Variable1	Variable2	Type of Censoring
Nonmissing	Nonmissing	Uncensored if *variable1* = *variable2*
Nonmissing	Nonmissing	Interval censored if *variable1* < *variable2*
Nonmissing	Missing	Right censored at *variable1*
Missing	Nonmissing	Left censored at *variable2*

For example, if T1 and T2 represent time in hours in the input data set

```
OBS    T1    T2

 1      .     6
 2      6    12
 3     12    24
 4     24     .
 5     24    24
```

then the statement

```
model (t1 t2);
```

specifies a model in which observation 1 is left censored at 6 hours, observation 2 is interval censored in the interval (6, 12), observation 3 is interval censored in (12,24), observation 4 is right censored at 24 hours, and observation 5 is an uncensored lifetime of 24 hours.

You can also use this method to specify a model for recurrent events data when exact recurrence times are not known. In this case, events are observed to have occurred in intervals specified by *(variable1 variable2)*. The values of the variable specified in a FREQ statement determine the number of events that occurred in each interval, and the values of the variable specified in a NENTER statement determine the number of units under observation in each interval.

effect-list
is a list of variables in the input data set representing the values of the independent variables in the model for each observation, and combinations of variables representing interaction terms. If a variable in the *effect-list* is also listed in a CLASS statement, an indicator variable is generated for each level of the variable. An indicator variable for a particular level is equal to 1 for observations with that level, and equal to 0 for all other observations. This type of variable is called a *classification* variable. Classification variables can be either character or numeric. If a variable is not listed in a CLASS statement, it is assumed to be a continuous variable, and it must be numeric.

options
control how the model is fit and what output is produced. All *options* are specified after a slash (/) in the MODEL statement. The "Summary of Options" section, which follows, lists all options by function.

Summary of Options

Table 18.32 Model Statement Options

Option	Option Description
CONFIDENCE=*number*	Specifies the confidence coefficient for all confidence intervals. Specify a *number* between 0 and 1. The default value is 0.95.

Table 18.32 Model Statement Options (continued)

Option	Option Description
CONVERGE=*number*	Specifies the convergence criterion for maximum likelihood fit. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CONVH=*number*	Specifies the convergence criterion for the relative Hessian convergence criterion. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CORRB	Requests parameter correlation matrix.
COVB	Requests parameter covariance matrix.
HPPTEST	Applies only to models for recurrent events data. This option requests a likelihood ratio test for a homogeneous Poisson process.
INEST | IN=*SAS-data-set*	Applies only to models for recurrent events data. This option specifies a SAS data set that can contain initial values, equality constraints, upper bounds, or lower bounds for the intercept and shape parameters in a model for recurrents events data. See the section "INEST Data Set for Recurrent Events Models" on page 1386 for details.
INITIAL=*number list*	Specifies initial values for regression parameters other than the location (intercept) term.
ITPRINT	Requests the iteration history for maximum likelihood fit.
LRCL	Requests likelihood ratio confidence intervals for distribution parameters.
LOCATION=*number* <LINIT>	Specifies the fixed or initial value of the location, or intercept parameter.
MAXIT=*number*	Specifies the maximum number of iterations allowed for maximum likelihood fit.

Table 18.32 Model Statement Options (continued)

Option	Option Description
OBSTATS	Requests a table that contains the XBETA, SURV, SRESID, and ADJRESID statistics in Table 18.33 or the XBETA, MCF, and INTENSITY statistics in Table 18.34. The table also contains the dependent and independent variables in the model. You can use this option to compute statistics such as survival function estimates for lifetime data or mean function estimates for recurrent events data for dependent variable values not included in the analysis. Refer to "Comparison of Two Samples of Repair Data" on page 1250 for an example of computing predicted values for recurrent events data.
OBSTATS(*statistics*)	Requests a table that contains the model variables and the statistics in the specified list of *statistics*. Available statistics are shown in Table 18.33.
ORDER=DATA | FORMATTED | FREQ | INTERNAL	Specifies the sort order for values of the classification variables in the *effect-list*.
PSTABLE=*number*	Specifies stable parameterization. The *number* must be between zero and one. See the section "Stable Parameters" on page 1359 for further information.
READOUT	Analyzes data in readout structure. The FREQ statement must be used to specify the number of units that fail in each interval, and the NENTER statement must be used to specify the number of unfailed units that enter each interval.

Table 18.32 Model Statement Options (continued)

Option	Option Description
RELATION=*transformation-keyword* RELATION=*(transformation-keyword1< , >transformation-keyword2)*	Specifies the type of relationship between independent and dependent variables. In the first form, the transformation specified is applied to the first continuous independent variable in the model. In the second form, the transformations specified within parentheses are applied to the first two continuous independent variables in the model, in the order listed. *transformation-keyword*, *transformation-keyword1*, and *transformation-keyword2* can be any of the transformations listed in the following table. See Table 18.67 for definitions of the transformations.

Transformation Keyword	Type of Transformation
ARRHENIUS	Arrhenius (Nelson parameterization)
ARRHENIUS2	Arrhenius (activation energy)
POWER	Logarithmic
LINEAR	Linear
LOGISTIC	Logistic

SCALE=*number* < SCINIT >	Specifies a fixed or initial value of scale parameter.
SHAPE=*number* < SHINIT >	Specifies a fixed or initial value of shape parameter.
SINGULAR=*number*	Specifies the singularity criterion for matrix inversion.
THRESHOLD=*number*	Specifies a fixed threshold parameter. See Table 18.57 for the distributions with a threshold parameter.
TREND=*trend-test keyword* | *(trend-test keywords)*	Applies only to models for recurrent events data. This option requests one or more tests of trend for a Poisson process. TREND=LRHPP is equivalent to the HPPTEST option. See the section "Tests of Trend" on page 1385 for more information about the tests. The available tests are shown in the following table.

Table 18.32 Model Statement Options (continued)

Option	Option Description

Trend-Test Keyword	Description of Test
MH | HDBK | MIL-HDBK	Military handbook test
LA | LAPLACE	Laplace's test
LR | LEWIS-ROBINSON	Lewis-Robinson test
LRHPP | LIKELIHOOD	Likelihood ratio tset
ALL	All available tests

WALDCL | NORMALCL

Requests Wald type confidence intervals for distribution parameters. See Table 18.68 and Table 18.74 for details about the computation of Wald confidence intervals. Wald confidence intervals are provided by default, but this option can be combined with LRCL to obtain both types of intervals.

Table 18.33 Available Statistics Computed for Each Observation with the OBSTATS Option for LIfetime Data

Option	Option Description

CENSOR

Is a variable that indicates the type of censoring for each observation in the input data set. The possible values for CENSOR and their interpretations are listed in the following table.

Type of Response	CENSOR Variable Value
Uncensored	0
Right-censored	1
Left-censored	2
Interval-censored	3

CONTROL=*variable*

Specifies a control variable in the input data set that allows the computation of statistics for a subset of observations in the input data set. If the value of *variable* is 1, the statistics are computed for that observation. If the value of the control variable is not equal to 1, the statistics are not computed for that observation.

Table 18.33 Available Statistics Computed for Each Observation with the OBSTATS Option (continued)

Option	Option Description
QUANTILES | QUANTILE | Q=*number-list*	Requests distribution quantiles for each number in *number-list* for each observation. The numbers must be between 0 and 1. Estimated quantile standard errors, and upper and lower confidence limits are also tabulated.
XBETA	Specifies the linear predictor.
SURVIVAL | SURV	Specifies the fitted survival function, evaluated at the value of the dependent variable.
RESID	Specifies the raw residual.
SRESID	Specifies the standardized residual.
GRESID	Specifies the modified Cox-Snell residual.
DRESID	Specifies the deviance residual.
ADJRESID	Specifies the adjusted standardized residuals. These are adjusted for right-censored observations by adding the median of the lifetime greater than the right-censored values to the residuals.
RESIDADJ=*number*	Specifies the adjustment to be added to Cox-Snell residual for right-censored data values. The default of *number* is 1.0, the mean of the standard exponential distribution.
RESIDALPHA | RALPHA=*number*	Specifies that the *number*$\times 100\%$ percentile residual lifetime be used to adjust right-censored standardized residuals. The *number* must be between 0 and 1. The default value is 0.5, which corresponds to the median.

Table 18.34 Available Statistics Computed for Each Observation with the OBSTATS Option for Recurrent Events Data

Option	Option Description
CONTROL=*variable*	Specifies a control variable in the input data set that allows the computation of statistics for a subset of observations in the input data set. If the value of *variable* is 1, the statistics are computed for that observation. If the value of *variable* is not equal to 1, the statistics are not computed for that observation.

Table 18.34 Available Statistics Computed for Each Observation with the OBSTATS Option (continued)

Option	Option Description
MCF	Specifies the mean function, which is evaluated at the value of time for each observation. Standard errors and confidence limits are also computed.
INTENSITY	Specifies the intensity function, which is evaluated at the value of time for each observation. Standard errors and confidence limits are also computed.
XBETA	Specifies the linear predictor.

NENTER Statement

NENTER *variable* **;**

Use the NENTER statement in conjunction with the FREQ statement to specify interval-censored lifetime data having a special structure; these data are called *readout* data. The NENTER statement specifies a *variable* in the input data set that determines the number of unfailed units entering each interval. See the section "Weibull Analysis of Interval Data with Common Inspection Schedule" on page 1219 for an example that uses the NENTER statement with readout data.

You can also use the NENTER statement in conjunction with the FREQ statement to specify recurrent events data when the event times are grouped into intervals, rather than being observed exactly. The NENTER statement specifies a variable in the input data set that determines the number of units observed in each interval.

NLOPTIONS Statement

NLOPTIONS < *options* > **;**

You use the NLOPTIONS statement to control aspects of the optimization system that is used to compute maximum likelihood estimates of the parameters of the three-parameter Weibull distribution with an ANALYZE or PROBPLOT statement, and of the parameters of models for recurrent events data with a MODEL statement. The syntax and options of the NLOPTIONS statement are described in Chapter 19, "Shared Concepts and Topics" (*SAS/STAT User's Guide*).

PROBPLOT Statement

< *label:* **>PROBPLOT** *variable* **<** ∗*censor-variable*(*values*) **>** **<** **=***group-variables* **>** **<** / *options* **>** **;**

< *label:* **>PROBPLOT (***variable1 variable2***)** **<** **=***group-variables* **>** **<** / *options* **>** **;**

You use the PROBPLOT statement to create a probability plot from complete, left-censored, right-censored, or interval-censored data.

You can specify the keyword PPLOT as an alias for PROBPLOT. You can specify any number of PROBPLOT statements after a PROC RELIABILITY statement. Each PROBPLOT statement creates a probability plot and an associated analysis. The probability distribution used in creating the probability plot and performing the analysis is determined by the DISTRIBUTION statement. You can specify an optional *label* to distinguish between multiple PROBPLOT statements in the output.

See the section "Analysis of Right-Censored Data from a Single Population" on page 1206 and the section "Weibull Analysis Comparing Groups of Data" on page 1210 for examples that create probability plots with the PROBPLOT statement.

To create a probability plot, you must specify one *variable*. If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values.

You can optionally specify one or two *group-variables* (also referred to as *classification variables*). The PROBPLOT statement displays a probability plot for each level of the *group-variables*. The observations in a given level are called a *cell*.

The elements of the PROBPLOT statement are described as follows.

variable
: represents the data for which a probability plot is to be produced. The *variable* must be a numeric variable in the input data set.

censor-variable(values)
: indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you can omit the specification of *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(variable1 variable2)
: is another method of specifying the data for which a probability plot is to be produced. You can use this syntax in a situation where uncensored, interval-censored, left-censored, and right-censored values occur in the same set of data. Table 18.31 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values. See the section "Lognormal Analysis with Arbitrary Censoring" on page 1224 for an example that uses this syntax to create a probability plot.

group-variables
: are one or two group variables. If no group variables are specified, a single probability plot is produced. The *group-variables* can be numeric or character variables in the input data set.

  Note that the parentheses surrounding the *group-variables* are needed only if two group variables are specified.

options

control the features of the probability plot. All *options* are specified after the slash (/) in the PROBPLOT statement. See the section "Summary of Options" on page 1312, which follows, for a list of all options by function.

Summary of Options

Table 18.35 lists analysis options that are available when you use either traditional graphics or ODS Graphics.

Table 18.35 Analysis Options

Option	Option Description
CONFIDENCE=*number*	Specifies the confidence coefficient for all confidence intervals. The *number* must be between 0 and 1. The default value is 0.95
CONVERGE=*number*	Specifies the convergence criterion for maximum likelihood fit. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CONVH=*number*	Specifies the convergence criterion for the relative Hessian convergence criterion. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CORRB	Requests the parameter correlation matrix.
COVB	Requests the parameter covariance matrix.
FITTYPE | FIT= *fit-specification*	Specifies the method of estimating distribution parameters. The available *fit-specifications* and their meanings are shown in the following table.

Fit Specification	Definition
LSYX	Least squares fit to the probability plot. The probability axis is the dependent variable.
LSXY	Least squares fit to the probability plot. The lifetime axis is the dependent variable.
MLE	Maximum likelihood (default).
MODEL	Use the fit from the preceding MODEL statement.
NONE	No fit is computed.
WEIBAYES <(CONFIDENCE | CONF=*number*)>	Weibayes fit. *number* is the confidence coefficient for the Weibayes fit and is between 0 and 1. The default is 0.95.

Table 18.35 Analysis Options (continued)

Option	Option Description
INEST | IN=*SAS-data-set*	Specifies a SAS data set that can contain initial values, equality constraints, upper bounds, or lower bounds for the scale, shape, and threshold parameters in a three-parameter Weibull model for lifetime data, and applies only to three-parameter Weibull models. See the section "INEST Data Set for the Three-Parameter Weibull" on page 1356 for details.
ITPRINT	Requests the iteration history for maximum likelihood fit.
ITPRINTEM	Requests the iteration history for the Turnbull algorithm.
LRCL	Requests likelihood ratio confidence intervals for distribution parameters.
LRCLPER	Requests likelihood ratio confidence intervals for distribution percentiles.
LRCLSURV	Requests likelihood ratio confidence intervals for survival and cumulative distribution functions at times specified with the SURVTIME=*number-list* option.
LOCATION=*number* < LINIT >	Specifies a fixed or initial value of location parameter.
MAKEHAM=*number* < MKINIT >	Specifies the fixed or initial value of the Makeham parameter for the three-parameter Gompertz distribution.
MAXIT=*number*	Specifies the maximum number of iterations allowed for a maximum likelihood fit.
MAXITEM=*number1* < ,*number2* >	*number1* Specifies the maximum number of iterations allowed for the Turnbull algorithm. Iteration history will be printed in increments of *number2* if an iteration history is requested with ITPRINTEM. See the section "Interval-Censored Data" on page 1347 for details.
NOPCTILES	Suppresses computation of percentiles for standard list of percentage points.
NOPOLISH	Suppresses the setting of small interval probabilities to 0 in the Turnbull algorithm. See the section "Interval-Censored Data" on page 1347 for details.
NPINTERVALS=*interval-type*	Specifies the type of nonparametric confidence interval displayed in a probability plot. The available types of intervals are listed in the following table.

Table 18.35 Analysis Options (continued)

Option	Option Description
	Interval Type / **Definition** POINTWISE | POINT — Pointwise confidence intervals for the CDF. See the section "Pointwise Confidence Intervals" on page 1353 for details. SIMULTANEOUS | SIMUL< *number1, number2* > — Simultaneous confidence intervals for the CDF. *number1* and *number2* are constants that control the time interval for which simultaneous intervals are computed. The default time intervals are the lowest and highest times corresponding to failures in the case of right-censored data, or to the lowest and highest intervals for which probabilities are computed for interval-censored data. See the section "Simultaneous Confidence Intervals" on page 1353 for details.
PCTLIST=*number-list*	Specifies a list of percentages for which to compute percentile estimates. The *number-list* must be a list of numbers separated by blanks or commas. Each number in the list must be between 0 and 100. If this option is not specified, percentiles are computed for a standard list of percentages.
PINTERVALS=*interval-type*	Specifies the type of parametric pointwise confidence interval displayed in a probability plot. The available types of intervals are listed in the following table. The default type is PROBABILITY, pointwise confidence intervals on cumulative failure probability.

Table 18.35 Analysis Options (continued)

Option	Option Description

Interval Type	Definition
LIKELIHOOD | LRCI	Likelihood ratio confidence intervals
PERCENTILES | PER	Pointwise parametric confidence intervals for the percentiles of the fitted CDF
PROBABILITY | CDF	Pointwise parametric confidence intervals for the cumulative failure probabilities. See the section "Reliability Function" on page 1364 for details.

PPOS=*plotting-position* Specifies the *plotting-position* type used to compute nonparametric estimates of the probability distribution function. See the section "Probability Plotting" on page 1343 for details. The plotting position types available are shown in the following table.

Plotting Position	Type
EXPRANK	Expected ranks
MEDRANK	Median ranks
MEDRANK1	Median ranks (exact formula)
KM	Kaplan-Meier
MKM	Modified Kaplan-Meier (default)
NA | NELSONAALEN	Nelson-Aalen

PPOUT Requests a table of nonparametric cumulative probabilities in the printed output.

PRINTPROBS Specifies that intervals and associated probabilities for the Turnbull algorithm be printed.

PROBLIST=*number-list* Specifies a list of initial values for Turnbull algorithm. See the section "Interval-Censored Data" on page 1347 for details.

PSTABLE=*number* Specifies a stable parameterization. The *number* must be between 0 and 1. See the section "Stable Parameters" on page 1359 for further information.

READOUT Specifies the data have the readout structure.

SCALE=*number* < SCINIT > Specifies a fixed or initial value of the scale parameter.

SHAPE=*number* < SHINIT > Specifies a fixed or initial value of the shape parameter.

SINGULAR=*number* Specifies the singularity criterion for matrix inversion.

Table 18.35 Analysis Options (continued)

Option	Option Description
SURVTIME=*number-list*	Requests that the survival function, cumulative distribution function, and confidence limits be computed for values in *number-list*. See the section "Reliability Function" on page 1364 for details.
THRESHOLD=*number*	Specifies a fixed threshold parameter. See Table 18.57 for the distributions with a threshold parameter.
TOLLIKE=*number*	Specifies the criterion for convergence in the Turnbull algorithm. The default is 10^{-8}. See the section "Interval-Censored Data" on page 1347 for details.
TOLPROB=*number*	Specifies the criterion for setting interval probabilities to 0 in the Turnbull algorithm. The default is 10^{-6}. See the section "Interval-Censored Data" on page 1347 for details.

Table 18.36 lists analysis options that are available when ODS Graphics is enabled.

Table 18.36 Analysis Options for ODS Graphics

Option	Option Description
PROFILE<(*options*)>	Requests a profile log-likelihood plot of the threshold parameter for a three-parameter Weibull distribution. The *options* listed in the following table are available; they are specified by enclosing them in parentheses after the PROFILE option.

Table 18.36 Analysis Options (continued)

Option	Option Description
Profile Option	**Option Description**
NOCONF	Specifies that a reference line on the vertical, log-likelihood axis not be drawn. If this option is not specified, a reference line is drawn at a log-likelihood value that corresponds to the profile likelihood confidence limits on the horizontal axis.
NPROFILE=*n*	Specifies that the profile log likelihood be computed and plotted at *n* threshold points. If this option is not specified, the profile log likelihood is computed and plotted at 100 points.
RANGE=(*value1*, *value2*)	Specifies the range of threshold values for which the profile log likelihood is computed and plotted as (*value1*, *value2*). If this option is not specified, the range of threshold values for which the profile log likelihood is computed is from 0 to the minimum failure time.

Table 18.37 lists plot layout options that are available when you use traditional graphics.

Table 18.37 Probability Plot Layout Options for Traditional Graphics

Option	Option Description
CENBIN	Specifies that censored data be plotted as frequency counts rather than as individual points.
CENSYMBOL= *symbol* | (*symbol-list*)	Specifies symbols for censored values. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z). For overlaid plots for groups of data, you can specify different symbols for the groups with a list of symbols or letters, separated by blanks, enclosed in parentheses. If no CENSYMBOL option is specified, the symbol used for censored values is the same as the symbol used for failures.

Table 18.37 Probability Plot Layout Options (continued)

Option	Option Description
HOFFSET=*value*	Specifies the offset for the horizontal axis.
INBORDER	Requests a border around probability plots.
INTERTILE=*value*	Specifies the distance between tiles.
LFIT=*linetype* | (*linetype list*)	Specifies line styles for fit lines and confidence curves in a probability plot. The *linetype list* is a list of numbers from 1 to 46 representing different linetypes; they can be separated by blanks or commas or can be a list in the form n_1 to n_2 <by n_3>.
MISSING1	Requests that missing values of the first GROUP= variable be treated as a level of the variable.
MISSING2	Requests that missing values of the second GROUP= variable be treated as a level of the variable.
NCOLS=*n*	Specifies that *n* columns be plotted on a page.
NOCENPLOT	Suppresses the plotting of censored data points.
NOCONF	Suppresses the plotting of percentile confidence curves.
NOFIT	Suppresses the plotting of fit line and percentile confidence curves.
NOFRAME	Suppresses the frame around the plotting area.
NOINSET	Suppresses the inset.
NOPPLEGEND	Suppresses the legend for overlaid probability plots
NOPPOS	Suppresses plotting of symbols for failures in a probability plot.
NROWS=*n*	Specifies that *n* rows be plotted on a page.
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	Specifies display order for values of the first GROUP= variable.
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	Specifies the display order for values of the second GROUP= variable.
OVERLAY	Requests overlaid plots for group variables.
PCONFPLT	Plots confidence intervals on probabilities for readout data.
PPLEGEND = *legend-statement-name* | NONE	Identifies LEGEND*n* statement to specify a legend for overlaid probability plots.
PPOSSYMBOL=*symbol* | (*symbol-list*)	Specifies symbols to represent failures on a probability plot.
ROTATE	Requests probability plots with the probability scale on the horizontal axis.

Table 18.37 Probability Plot Layout Options (continued)

Option	Option Description
SHOWMULTIPLES	Requests that the count be displayed for multiple overlaying symbols.
TURNVLABELS	Vertically strings out characters in labels for the vertical axis.
VOFFSET=*value*	Specifies *value* as the length of the offset at the upper end of the vertical axis.
WFIT=*n*	Specifies the line width for the fit line and confidence curves.

Table 18.38 lists plot layout options that are available when you use ODS graphics.

Table 18.38 Probability Plot Layout Options for ODS Graphics

Option	Option Description
MISSING1	Requests that missing values of first GROUP= variable be treated as a level of the variable.
MISSING2	Requests that missing values of second GROUP= variable be treated as a level of the variable.
NCOLS=*n*	Specifies that *n* columns be plotted on a page.
NOCENPLOT	Suppresses plotting of censored data points.
NOCONF	Suppresses plotting of percentile confidence curves.
NOFIT	Suppresses plotting of the fit line and percentile confidence curves.
NOINSET	Suppresses the inset.
NOPPLEGEND	Suppresses the legend for overlaid probability plots.
NOPPOS	Suppresses plotting of symbols for failures in a probability plot.
NROWS=*n*	Specifies that *n* rows be plotted on a page.
ORDER1=DATA | FORMATTED | FREQ | INTERNAL	Specifies the display order for values of the first GROUP= variable.
ORDER2=DATA | FORMATTED | FREQ | INTERNAL	Specifies the display order for values of the second GROUP= variable.
OVERLAY	Requests overlaid plots for group variables.
PCONFPLT	Plots confidence intervals on probabilities for readout data.
ROTATE	Requests probability plots with the probability scale on the horizontal axis.

Table 18.39 lists reference line options that are available when you use traditional graphics.

Table 18.39 Reference Line Options for Traditional Graphics

Option	Option Description
HREF <(INTERSECT)>=*value-list*	Requests reference lines perpendicular to horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value.
HREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for HREF= lines.
HREFLABPOS=*n*	Specifies vertical position of labels for HREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table: *n* / **Label Placement** 1 Top 2 Staggered from top 3 Bottom 4 Staggered from bottom 5 Alternating from top 6 Alternating from bottom
LHREF=*linetype*	Specifies the line style for HREF= lines.
LVREF=*linetype*	Specifies the line style for VREF= lines.
VREF <(INTERSECT)>=*value-list*	Specifies reference lines perpendicular to vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value.
VREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for VREF= lines.
VREFLABPOS=*n*	Specifies horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table: *n* / **Label Placement** 1 Left 2 Right

Table 18.40 lists reference line options that are available when you use ODS graphics.

Table 18.40 Reference Line Options for ODS Graphics

Option	Option Description
LREF < (INTERSECT) >=*value-list*	Requests reference lines perpendicular to the lifetime axis. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the probability axis and intersects the fit line at the same point as the lifetime axis reference line. If a lifetime axis reference line label is specified, the intersecting probability axis reference line is labeled with the probability axis value.
LREFLABELS =(*'label1' ... 'labeln'*)	Specifies labels for LREF= lines.
PREF < (INTERSECT) >=*value-list*	Specifies reference lines perpendicular to the probability axis. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the lifetime axis and intersects the fit line at the same point as the probability axis reference line. If a probability axis reference line label is specified, the intersecting lifetime axis reference line is labeled with the lifetime axis value.
PREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for PREF= lines.

Table 18.41 lists options that control the appearance of the text when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.41 Text Enhancement Options

Option	Option Description
FONT=*font*	Specifies a software font for text.
HEIGHT=*value*	Specifies the height of text used outside framed areas.
INFONT=*font*	Specifies a software font for text inside framed areas.
INHEIGHT=*value*	Specifies the height of text inside framed areas.

Table 18.42 lists options that control the appearance of the axes when you use traditional graphics.

Table 18.42 Axis Options for Traditional Graphics

Option	Option Description

LAXIS=*value1 to value2< by value3 >*	Specifies tick mark values for the lifetime axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of LAXIS= lists are
	`laxis = -1 to 10`
	`laxis = 0 to 200 by 10`

LGRID=*number*	Specifies a line style for all grid lines. The *number* is between 1 and 46 and specifies a linestyle for grids.
LIFELOWER | LLOWER=*number*	Specifies the lower limit on the lifetime axis scale. The LLOWER option specifies *number* as the lower lifetime axis tick mark. The tick interval and the upper lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
LIFEUPPER | LUPPER=*number*	Specifies the upper limit on the lifetime axis scale. The LUPPER option specifies *number* as the upper lifetime axis tick mark. The tick interval and the lower lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
MPGRID	Adds a minor grid for the probability axis.
MINORLOGGRID	Adds a minor grid for log axes.
NOGRID	Suppresses grid lines.
NOLLABEL	Suppresses label for life, or analysis variable, axis.
NOLTICK	Suppresses tick marks and tick mark labels for lifetime or analysis variable axis.
NOPLABEL	Suppresses label for probability axis.
NOPTICK	Suppresses tick marks and tick mark labels for the probability axis.
NTICK=*number*	Specifies the number of tick intervals for the lifetime axis. This option has no effect if the LAXIS option is used.

Table 18.42 Axis Options (continued)

Option	Option Description
PCTLOWER | PLOWER=*number*	Specifies the lower limit on probability axis scale.
PCTUPPER | PUPPER=*number*	Specifies the upper limit on probability axis scale.
PAXISLABEL='*string*'	Specifies a label for the probability axis.
WAXIS=*n*	Specifies the line thickness for axes and frame.

Table 18.43 lists options that control the appearance of the axes when you use ODS Graphics.

Table 18.43 Axis Options for ODS Graphics

Option	Option Description

LAXIS=*value1 to value2< by value3 >*	Specifies tick mark values for the lifetime axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of LAXIS= lists are
	`laxis = -1 to 10`
	`laxis = 0 to 200 by 10`

LIFELOWER | LLOWER=*number*	Specifies the lower limit on the lifetime axis scale. The LLOWER option specifies *number* as the lower lifetime axis tick mark. The tick interval and the upper lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
LIFEUPPER | LUPPER=*number*	Specifies the upper limit on the lifetime axis scale. The LUPPER option specifies *number* as the upper lifetime axis tick mark. The tick interval and the lower lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
MINORLOGGRID	Adds a minor grid for log axes.
NOGRID	Suppresses grid lines.
PCTLOWER | PLOWER=*number*	Specifies the lower limit on probability axis scale.

Table 18.43 Axis Options (continued)

Option	Option Description
PCTUPPER | PUPPER=*number*	Specifies the upper limit on probability axis scale.
PAXISLABEL='*string*'	Specifies a label for the probability axis.

Table 18.44 lists options that control colors and patterns used in the graph when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.44 Color and Pattern Options

Option	Option Description
CAXIS=*color*	Color for axis
CCENSOR=*color*	Color for filling censor plot area
CENCOLOR=*color*	Color for censor symbol
CFIT=*color* | (*color list*)	color for fit lines and confidence curves in a probability plot
CFRAME=*color*	Color for frame
CFRAMESIDE=*color*	Color for filling frame for row labels
CFRAMETOP=*color*	Color for filling frame for column labels
CGRID=*color*	Color for grid lines
CHREF=*color*	Color for HREF= lines
CTEXT=*color*	Color for text
CVREF=*color*	Color for VREF= lines
PPOSCOLOR=*color* | (*color list*)	Colors of symbols that represent failures on a probability plot

Table 18.45 lists options that control the use of a graphics catalog to store graphs if you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.45 Graphics Catalog Options

Option	Option Description
DESCRIPTION='*string*'	Description for graphics catalog member
NAME='*string*'	Name for plot in graphics catalog

RELATIONPLOT Statement

< *label:* **>RELATIONPLOT** *variable* **<** ∗*censor-variable*(*values*) **>** **<** **=***group-variable* **>**
< / *options* **>** **;**

< *label:* **>RELATIONPLOT (***variable1 variable2***)** **<** **=***group-variable* **>** **<** / *options* **>** **;**

You use the RELATIONPLOT statement to create life-stress relation plots. A life-stress relation plot is a graphical tool for the analysis of data from accelerated life tests. The plot is a display of the relationship between life and *stress*, such as temperature or voltage. You can also use the RELATIONPLOT statement to display a probability plot alongside the relation plot. See Figure 18.9 for an example of a relation plot.

You can specify the keyword RPLOT as an alias for RELATIONPLOT. You can use any number of RELATIONPLOT statements after a PROC RELIABILITY statement. You can specify an optional *label* to distinguish between multiple RELATIONPLOT statements in the output.

See the section "Analysis of Accelerated Life Test Data" on page 1214 for an example that uses the RELATIONPLOT statement.

To create a life-stress relation plot, you must specify one *variable*. If your data are right censored, you must specify a *censor-variable* and, in parentheses, the *values* of the *censor-variable* that correspond to censored data values. You must specify one *group-variable* to represent the values of stress. The *group-variable* must be a numeric variable.

The RELATIONPLOT statement plots the uncensored values of your data given by *variable* versus the values of the *group-variable*. You can optionally display a boxplot of the values of the data. You can also plot percentiles of the distribution fitted to the data. The RELATIONPLOT statement produces the same tabular output as the PROBPLOT statement, and all the analysis options are the same as for the PROBPLOT statement.

The elements of the RELATIONPLOT statement are described as follows.

variable
: represents the data for which a plot is to be produced. The *variable* must be a numeric variable in the input data set.

censor-variable(values)
: indicates which observations in the input data set are right censored. You specify the values of *censor-variable* that represent censored observations by placing those values in parentheses after the variable name. If your data are not right censored, then you omit the specification of *censor-variable*; otherwise, *censor-variable* must be a numeric variable in the input data set.

(variable1 variable2)
: is another method of specifying the data for which a life-stress plot is to be produced. You can use this syntax in a situation where uncensored, interval-censored, left-censored, and right-censored values occur in the same set of data. Table 18.31 shows how you use this syntax to specify different types of censoring by using combinations of missing and nonmissing values. See the section "Lognormal Analysis with Arbitrary Censoring" on page 1224 for an example that uses this syntax to create a probability plot.

group-variable
is a group variable. The *group-variable* must be a numeric variable in the input data set.

options
control the features of the relation plot. All *options* are specified after the slash (/) in the RELATIONPLOT statement. The "Summary of Options" section, which follows, lists all options by function.

The only type of relation plot currently available for interval data is the type in which percentiles of the fitted distribution are plotted at each stress level.

Summary of Options

Table 18.46 Analysis Options

Option	Option Description
CONFIDENCE=*number*	Specifies the confidence coefficient for all confidence intervals. The *number* must be between 0 and 1. The default value is 0.95.
CONVERGE=*number*	Specifies the convergence criterion for maximum likelihood fit. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CONVH=*number*	Specifies the convergence criterion for the relative Hessian convergence criterion. See the section "Maximum Likelihood Estimation" on page 1354 for details.
CORRB	Requests the parameter correlation matrix.
COVB	Requests the parameter covariance matrix.
FITTYPE | FIT= *fit-specification*	Specifies the method of estimating distribution parameters. The available *fit-specifications* and their meanings are shown in the following table.

Table 18.46 Analysis Options (continued)

Option	Option Description

Fit Specification	Definition
LSYX	Least squares fit to the probability plot. The probability axis is the dependent variable.
LSXY	Least squares fit to the probability plot. The lifetime axis is the dependent variable.
MLE	Maximum likelihood (default).
MODEL	Use the fit from the preceding MODEL statement
NONE	No fit is computed.
REGRESSION<(*variable=number-list*)>	Use the fit from the preceding MODEL statement. Nonlinear relations and percentiles from models that use two independent variables can be plotted. *variable* is a second independent variable in the MODEL statement, and *number-list* is a list of values of the second independent variable for which to plot the percentiles as a function of the first independent variable.
WEIBAYES <(CONFIDENCE | CONF=*number*)>	Weibayes fit. *number* is the confidence coefficient for the Weibayes fit and is between 0 and 1. The default is 0.95.

ITPRINT	Requests the iteration history for the maximum likelihood fit.
LRCL	Requests likelihood ratio confidence intervals for distribution parameters.
LRCLPER	Requests likelihood ratio confidence intervals for distribution percentiles.
LOCATION=*number* <LINIT>	Specifies a fixed or initial value of the location parameter.
MAXIT=*number*	Specifies the maximum number of iterations allowed for a maximum likelihood fit.
NOPCTILES	Suppresses computation of percentiles.

Table 18.46 Analysis Options (continued)

Option	Option Description
PCTLIST=*number-list*	Specifies a list of percentages for which to compute percentile estimates. The *number-list* must be a list of numbers separated by blanks or commas. Each number in the list must be between 0 and 100. If this option is not specified, percentiles are computed for a standard list of percentages.
PPOS=*plotting-position*	Specifies the *plotting-position* type used to compute nonparametric estimates of the probability distribution function. See the section "Probability Plotting" on page 1343 for details. The plotting position types available are shown in the following table.

Plotting Position	Type
EXPRANK	Expected ranks
MEDRANK	Median ranks
MEDRANK1	Median ranks (exact formula)
KM	Kaplan-Meier
MKM	Modified Kaplan-Meier (default)
NA | NELSONAALEN	Nelson-Aalen

PPOUT	Requests a table of nonparametric cumulative probabilities in the printed output.
PSTABLE=*number*	Specifies a stable parameterization. The *number* must be between zero and one. See the section "Stable Parameters" on page 1359 for further information.
RELATION=*transformation-keyword*	Specifies the type of relationship between independent (stress) and dependent (lifetime) variables. The transformation specified is applied to the independent (stress) variable in the model. This determines the horizontal scale used in the relation plot. *transformation-keyword* can be any of the transformations listed in the following table. See Table 18.67 for definitions of the transformations.

Transformation Keyword	Type of Transformation
ARRHENIUS	Arrhenius (Nelson parameterization)
ARRHENIUS2	Arrhenius (activation energy)
POWER	Logarithmic
LINEAR	Linear
LOGISTIC	Logistic

Table 18.46 Analysis Options (continued)

Option	Option Description
READOUT	Specifies the data has the readout structure.
SCALE=*number* < SCINIT >	Specifies a fixed or initial value of scale parameter.
SHAPE=*number* < SHINIT >	Specifies a fixed or initial value of shape parameter.
SINGULAR=*number*	Specifies the singularity criterion for matrix inversion.
SURVTIME=*number-list*	Requests that survival function be computed for values in *number-list*. See the section "Reliability Function" on page 1364 for details.
THRESHOLD=*number*	Specifies a fixed threshold parameter. See Table 18.57 for the distributions with a threshold parameter.
variable=number-list	Enables creation of plots of percentiles from a regression model when two independent variables are used in a MODEL statement *effect-list*. The FIT=REGRESSION option must be used with this option. Percentile plots are created for each value of the independent *variable* in the *number-list*. *number-list* is a list of numeric values separated by blanks or commas, or in the form of a list n_1 to n_2 < by n_3 >.

Table 18.47 lists plot layout options that are available when you use traditional graphics.

Table 18.47 Plot Layout Options for Traditional Graphics

Option	Option Description
CENSYMBOL=*symbol* | (*symbol-list*)	Specifies symbols for censored values. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z). If you are creating overlaid plots for groups of data, you can specify different symbols for the groups with a list of symbols or letters, separated by blanks, enclosed in parentheses. If no CENSYMBOL option is specified, the symbol used for censored values is the same as for failures.
HOFFSET=*value*	Specifies an offset for horizontal axis.
INBORDER	Requests a border around plots.
LBOXES=*number*	Specifies a line style for boxplots.

Table 18.47 Plot Layout Options (continued)

Option	Option Description
LFIT=*linetype* | (*linetype-list*)	Specifies line styles for fit lines and confidence curves in a probability plot. The *linetype-list* is a list of numbers from 1 to 46 representing different linetypes; the numbers can be separated by blanks or commas or can be a list in the form n_1 to n_2 <by n_3>.
LPLOTFIT=*linetype* | (*linetype-list*)	Specifies line styles for percentile lines. *linetype-list* is a list of numbers that represent different linetypes; the numbers can be separated by blanks or commas or can be a list in the form n_1 to n_2 <by n_3>.
NOCENPLOT	Suppresses plotting of censored data points.
NOCONF	Suppresses plotting of percentile confidence curves.
NOFIT	Suppresses plotting of fit line and percentile confidence curves.
NOFRAME	Suppresses the frame around the plotting area.
NOPPLEGEND	Suppresses the legend for overlaid probability plots.
NOPPOS	Suppresses plotting of symbols for failures in a probability plot.
NORPLEGEND	Suppresses the legend for the relation plot.
PINTERVALS=*interval-type*	Specifies the type of parametric pointwise confidence interval displayed in a probability plot. The available types of intervals are listed in the following table. The default type is PROBABILITY, pointwise confidence intervals on cumulative failure probability.

Interval Type	Definition
LIKELIHOOD | LRCI	Likelihood ratio confidence intervals
PERCENTILES | PER	Pointwise parametric confidence intervals for the percentiles of the fitted CDF
PROBABILITY | CDF	Pointwise parametric confidence intervals for the cumulative failure probabilities. See the section "Reliability Function" on page 1364 for details.

Table 18.47 Plot Layout Options (continued)

Option	Option Description
PLOTDATA <DATA | MEDIANS | BOXES>	Requests that the data be plotted on the relationplot and specifies the representation of the data populations to be plotted.
PLOTFIT <*number-list*>	Specifies that percentiles of the fitted distribution be plotted on the relation plot. The optional *number-list* is a list of percentiles (between 0 and 100); if not specified, the 50th percentile (median) is plotted.
PPLEGEND = *legend-statement-name* | NONE	Identifies a LEGEND*n* statement to specify legend for overlaid probability plots.
PPLOT	Places a probability plot on the same page as the relation plot.
RCENSYMBOL=*symbol* | (*symbol-list*)	Specifies symbols that represent right-censored and left-censored observations in a relation plot. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, circle) or a letter (A–Z).
RPLEGEND = *legend-statement-name* | NONE	Identifies a LEGEND*n* statement to specify legend for the relation plot.
SHOWMULTIPLES	Displays the count for multiple overlaying symbols.
TURNVLABELS	Vertically strings out characters in labels for vertical axis.
VOFFSET=*value*	Specifies length of offset at upper end of vertical axis.
WFIT=*linetype*	Specifies line width for fit line and confidence curves.

Table 18.48 lists plot layout options that are available when you use ODS graphics.

Table 18.48 Plot Layout Options for ODS Graphics

Option	Option Description
NOCENPLOT	Suppresses plotting of censored data points.
NOCONF	Suppresses plotting of percentile confidence curves.
NOFIT	Suppresses plotting of fit line and percentile confidence curves.

Table 18.48 Plot Layout Options (continued)

Option	Option Description
NOPPOS	Suppresses plotting of symbols for failures in a probability plot.
PINTERVALS=*interval-type*	Specifies the type of parametric pointwise confidence interval displayed in a probability plot. The available types of intervals are listed in the following table. The default type is PROBABILITY, pointwise confidence intervals on cumulative failure probability.

Interval Type	Definition
LIKELIHOOD | LRCI	Likelihood ratio confidence intervals
PERCENTILES | PER	Pointwise parametric confidence intervals for the percentiles of the fitted CDF
PROBABILITY | CDF	Pointwise parametric confidence intervals for the cumulative failure probabilities. See the section "Reliability Function" on page 1364 for details.

PLOTDATA	Requests that the data be plotted on the relationplot.
PLOTFIT < *number-list* >	Specifies that percentiles of the fitted distribution be plotted on the relation plot. The optional *number-list* is a list of percentiles (between 0 and 100); if not specified, the 50th percentile (median) is plotted.
PPLOT	Places a probability plot on the same page as the relation plot.

Table 18.49 lists reference line options that are available when you use traditional graphics.

Table 18.49 Reference Line Options for Traditional Graphics

Option	Option Description
HREF < (INTERSECT) >=*value-list*	Requests reference lines perpendicular to horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value.
HREFLABELS=*('label1' ... 'labeln')*	Specifies labels for HREF= lines.
HREFLABPOS=*n*	Specifies the vertical position of labels for HREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table: *n* — **Label Placement** 1 — Top 2 — Staggered from top 3 — Bottom 4 — Staggered from bottom 5 — Alternating from top 6 — Alternating from bottom
LHREF=*linetype*	Specifies a line style for HREF= lines.
LSREF=*linetype*	Specifies a line style for SREF= lines.
LVREF=*linetype*	Specifies a line style for VREF= lines.
SREF=*value-list*	Specifies reference lines perpendicular to horizontal stress axis.
SREFLABELS=*('label1' ... 'labeln')*	Specifies labels for SREF= lines
SREFLABPOS=*n*	Specifies horizontal position of labels for SREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table: *n* — **Label Placement** 1 — Top 2 — Staggered from top 3 — Bottom 4 — Staggered from bottom

Table 18.49 Reference Line Options (continued)

Option	Option Description
VREF < (INTERSECT) >=*value-list*	Requests reference lines perpendicular to vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value.
VREFLABELS=*('label1' ... 'labeln')*	Specifies labels for VREF= lines.
VREFLABPOS=*n*	Specifies the horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table:

n	Label Placement
1	Left
2	Right

Table 18.50 lists reference line options that are available when you use ODS graphics.

Table 18.50 Reference Line Options for ODS Graphics

Option	Option Description
LREF < (INTERSECT) >=*value-list*	Requests reference lines perpendicular to lifetime axis. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the stress axis (and the probability axis if applicable), and it intersects the fit line at the same point as the lifetime axis reference line. If a lifetime axis reference line label is specified, the intersecting stress/probability axis reference line is labeled with the stress/probability axis value.
LREFLABELS=*('label1' ... 'labeln')*	Specifies labels for LREF= lines.
PREF < (INTERSECT) >=*value-list*	Requests reference lines perpendicular to probability axis, if a probability plot is specified with the PPLOT option. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the lifetime axis and intersects the fit line at the same point as the probability axis reference line. If a probability axis reference line label is specified, the intersecting lifetime axis reference line is labeled with the lifetime axis value.

Table 18.50 Reference Line Options (continued)

Option	Option Description
PREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for PREF= lines.
SREF <(INTERSECT)>=*value-list*	Specifies reference lines perpendicular to the stress axis. If (INTERSECT) is specified, a second reference line is drawn perpendicular to the lifetime axis and intersects the fit line at the same point as the stress axis reference line. If a stress axis reference line label is specified, the intersecting lifetime axis reference line is labeled with the lifetime axis value.
SREFLABELS=(*'label1' ... 'labeln'*)	Specifies labels for SREF= lines.

Table 18.51 lists options that control the appearance of the text when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.51 Text Enhancement Options

Option	Option Description
FONT=*font*	Specifies a software font for text.
HEIGHT=*value*	Specifies the height of text used outside framed areas.
INFONT=*font*	Specifies a software font for text inside framed areas.
INHEIGHT=*value*	Specifies the height of text inside framed areas.

Table 18.52 lists options that control the appearance of the axes when you use traditional graphics.

Table 18.52 Axis Options for Traditional Graphics

Option	Option Description

LAXIS=*value1* TO *value2*< BY *value3*>	Specifies tick mark values for the lifetime axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used. This method of specification of tick marks is not valid for logarithmic axes. Examples of LAXIS= lists are
	`laxis = -1 to 10`
	`laxis = 0 to 200 by 10`

LGRID=*number*	Specifies a line style for all grid lines. The *number* is between 1 and 46 and specifies a linestyle for grids.
LIFELOWER | LLOWER=*number*	Specifies the lower limit on the lifetime axis scale. The LLOWER option specifies *number* as the lower lifetime axis tick mark. The tick interval and the upper lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
LIFEUPPER | LUPPER=*number*	Specifies the upper limit on the lifetime axis scale. The LUPPER option specifies *number* as the upper lifetime axis tick mark. The tick interval and the lower lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
MPGRID	Adds a minor grid for the probability axis.
MINORLOGGRID	Adds a minor grid for log axes.
NOGRID	Suppresses grid lines.
NOLLABEL	Suppresses the label for the lifetime axis.
NOLTICK	Suppresses tick marks and tick mark labels for lifetime or analysis variable axis.
NOPLABEL	Suppresses the label for the probability axis.
NOPTICK	Suppresses tick marks and tick mark labels for probability axis.
NOSLABEL	Suppresses the label for the stress axis.
NOSTICK	Suppresses tick marks and tick mark labels for stress axis.

Table 18.52 Axis Options (continued)

Option	Option Description
NSTRESSTICK=*number*	Specifies the number of tick intervals for stress axis for relation plot.
NTICK=*number*	Specifies the number of tick intervals for the lifetime axis. This option has no effect if the LAXIS option is used.
PCTLOWER | PLOWER=*number*	Specifies the lower limit on the probability axis scale.
PCTUPPER | PUPPER=*number*	Specifies the upper limit on the probability axis scale.
STRESSLOWER | SLOWER=*number*	Specifies the lower limit on the stress axis scale.
STRESSUPPER | SUPPER=*number*	Specifies the upper limit on the stress axis scale.
PAXISLABEL='*string*'	Specifies a label for the probability axis.
WAXIS=*n*	Specifies the line thickness for axes and frame.

Table 18.53 lists options that control the appearance of the axes when you use ODS Graphics.

Table 18.53 Axis Options for ODS Graphics

Option	Option Description
LIFELOWER | LLOWER=*number*	Specifies the lower limit on the lifetime axis scale. The LLOWER option specifies *number* as the lower lifetime axis tick mark. The tick interval and the upper lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
LIFEUPPER | LUPPER=*number*	Specifies the upper limit on the lifetime axis scale. The LUPPER option specifies *number* as the upper lifetime axis tick mark. The tick interval and the lower lifetime axis limit are determined automatically. This option has no effect if the LAXIS option is used.
NOGRID	Suppresses grid lines.
PCTLOWER | PLOWER=*number*	Specifies lower the limit on the probability axis scale.
PCTUPPER | PUPPER=*number*	Specifies upper the limit on the probability axis scale.
STRESSLOWER | SLOWER=*number*	Specifies the lower limit on the stress axis scale.
STRESSUPPER | SUPPER=*number*	Specifies the upper limit on the stress axis scale.
PAXISLABEL='*string*'	Specifies a label for the probability axis.

Table 18.54 lists options that control the use of a graphics catalog to store graphs if you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.54 Graphics Catalog Options

Option	Option Description
DESCRIPTION='*string*'	Description for graphics catalog member
NAME='*string*'	Name for plot in graphics catalog

Table 18.55 lists options that control colors and patterns used in the graph when you use traditional graphics. These options are not available if ODS Graphics is enabled.

Table 18.55 Color and Pattern Options

Option	Option Description
CAXIS=*color*	Color for axis
CBOXES=*color*	Color for box frame for boxplots
CBOXFILL=*color*	Color for filling boxes for boxplots
CCENSOR=*color*	Color for filling censor plot area
CENCOLOR=*color*	Color for censor symbol
CFIT=*color* | (*color-list*)	color for fit lines and confidence curves in a probability plot
CFRAME=*color*	Color for frame
CGRID=*color*	Color for grid lines
CHREF=*color*	Color for HREF= lines
CPLOTFIT=*color* | (*color-list*)	colors for percentile lines
CSREF=*color*	Color for SREF= lines
CTEXT=*color*	Color for text
CVREF=*color*	Color for VREF= lines
RCENCOLOR=*color* | (*color-list*)	Colors for the symbols representing uncensored, right-censored, and left-censored observations in a relation plot

SLICE Statement

SLICE *model-effect* < / *options* > **;**

The SLICE statement provides a general mechanism for performing a partitioned analysis of the LS-means for an interaction. This analysis is also known as an analysis of simple effects.

The SLICE statement uses the same *options* as the LSMEANS statement, which are summarized in Table 19.21 (*SAS/STAT User's Guide*). For details about the syntax of the SLICE statement, see the section "SLICE Statement" (Chapter 19, *SAS/STAT User's Guide*).

STORE Statement

STORE < **OUT=** >*item-store-name* < / **LABEL=**'*label*' > **;**

The STORE statement requests that the procedure save the context and results of the statistical analysis. The resulting item store has a binary file format that cannot be modified. The contents of the item store can be processed with the PLM procedure. For details about the syntax of the STORE statement, see the section "STORE Statement" (Chapter 19, *SAS/STAT User's Guide*).

TEST Statement

TEST < *model-effects* > < / *options* > **;**

The TEST statement enables you to perform *F* tests for model effects that test Type I, Type II, or Type III hypotheses. See Chapter 15, "The Four Types of Estimable Functions" (*SAS/STAT User's Guide*), for details about the construction of Type I, II, and III estimable functions.

Table 18.56 summarizes the *options* available in the TEST statement.

Table 18.56 TEST Statement Options

Option	Description
CHISQ	Requests chi-square tests
DDF=	Specifies denominator degrees of freedom for fixed effects
E	Requests Type I, Type II, and Type III coefficients
E1	Requests Type I coefficients
E2	Requests Type II coefficients
E3	Requests Type III coefficients
HTYPE=	Indicates the type of hypothesis test to perform
INTERCEPT	Adds a row that corresponds to the overall intercept

For details about the syntax of the TEST statement, see the section "TEST Statement" (Chapter 19, *SAS/STAT User's Guide*) in Chapter 19, "Shared Concepts and Topics" (*SAS/STAT User's Guide*).

UNITID Statement

UNITID *variable* **;**

The UNITID statement names a *variable* in the input data set that is used to identify each individual unit in an MCFPLOT statement. The value of the UNITID variable for an observation corresponds to the name of the unit in the study for which a repair or end of history has occurred. See the section "Analysis of Recurrence Data on Repairs" on page 1245 for an example that uses the UNITID statement with the MCFPLOT statement.

Details: RELIABILITY Procedure

Abbreviations and Notation

The following abbreviations and notation are used in this section:

CDF	Cumulative distribution function: $F(x) = Pr\{X \leq x\}$
log	Base e logarithm
$\log_{10}$	Base 10 logarithm
Reliability or survivor function	$R(x) = Pr\{X > x\}$
x_p	$p \times 100\%$ percentile: $Pr\{X \leq x_p\} = p$

Types of Lifetime Data

This section describes various types of data that you can analyze with the RELIABILITY procedure.

Lifetime data for which the values of all sample units are observed are called *complete* data. This means that the failure times are observed for all units.

Many practical problems in life data analysis involve data for which some units are unfailed. The failure time for an unfailed unit is known only to be greater than the last running time. This type of data is said to be *right censored*, and the censoring time is used in the analysis of the data. Data for which censoring times are intermixed with failure times are sometimes called *multiply censored* or *progressively censored*.

Failure times may be known only to be less than some value. This type of data is called *left censored*.

Another common situation is where the failure times of units are not known exactly, but time intervals that contain the failure times are known. This type of data is called *interval censored*.

Interval-censored data for which all units share common interval endpoints are called *readout*, *inspection*, or *grouped* data.

Arbitrarily censored data can contain a combination of failures, right-, left-, and interval-censored data.

Probability Distributions

This section describes the probability distributions available in the RELIABILITY procedure for probability plotting and parameter estimation.

PROBPLOT and RELATIONPLOT Statements

Probability plots can be constructed for each of the probability distributions in Table 18.57.

For all distributions other than the three-parameter Weibull, estimates of two distribution parameters (*location* and *scale* or *scale* and *shape*) are computed by maximum likelihood or by least squares fitted to points on the probability plot. If one of the parameters is specified as fixed, the other is estimated. In addition, you can

specify a fixed *threshold*, or *shift*, parameter for distributions for which a threshold parameter is indicated in Table 18.57. If you do not specify a threshold parameter, the threshold is set to 0.

For the three-parameter Weibull distribution described in Table 18.57, the scale, shape, and threshold parameters are estimated by maximum likelihood.

You should not interpret the parameters μ and σ as representing the means and standard deviations for all of the distributions in Table 18.57. The normal is the only distribution in Table 18.57 for which this is the case.

Table 18.57 Distributions and Parameters for PROBPLOT and RELATIONPLOT Statements

		Parameters			
Distribution	**Density Function**	**Location**	**Scale**	**Shape**	**Threshold**
Normal	$\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	μ	σ		
Lognormal	$\frac{1}{\sqrt{2\pi}\sigma(x-\theta)}\exp\left(-\frac{(\log(x-\theta)-\mu)^2}{2\sigma^2}\right)$	μ	σ		θ
Lognormal (base 10)	$\frac{\log(10)}{\sqrt{2\pi}\sigma(x-\theta)}\exp\left(-\frac{(\log_{10}(x-\theta)-\mu)^2}{2\sigma^2}\right)$	μ	σ		θ
Extreme value	$\frac{1}{\sigma}\exp\left(\frac{x-\mu}{\sigma}\right)\exp\left(-\exp\left(\frac{x-\mu}{\sigma}\right)\right)$	μ	σ		
Weibull	$\frac{\beta}{\alpha^\beta}(x-\theta)^{\beta-1}\exp\left(-\left(\frac{x-\theta}{\alpha}\right)^\beta\right)$		α	β	θ
Exponential	$\frac{1}{\alpha}\exp\left(-\left(\frac{x-\theta}{\alpha}\right)\right)$		α		θ
Logistic	$\frac{\exp\left(\frac{x-\mu}{\sigma}\right)}{\sigma\left[1+\exp\left(\frac{x-\mu}{\sigma}\right)\right]^2}$	μ	σ		
Log-logistic	$\frac{\exp\left(\frac{\log(x-\theta)-\mu}{\sigma}\right)}{(x-\theta)\sigma\left[1+\exp\left(\frac{\log(x-\theta)-\mu}{\sigma}\right)\right]^2}$	μ	σ		θ
Three-parameter Weibull	$\frac{\beta}{\alpha^\beta}(x-\theta)^{\beta-1}\exp\left(-\left(\frac{x-\theta}{\alpha}\right)^\beta\right)$		α	β	θ
Gompertz	$\frac{\beta}{\alpha}\exp\left\{\frac{t}{\alpha}+\beta-\beta\exp\left(\frac{t}{\alpha}\right)\right\}$		α	β	
Three-parameter Gompertz	$\left\{\frac{\beta}{\alpha}\exp(\frac{t}{\alpha})+\frac{\theta}{\alpha}\right\}$ $\times\exp\left\{-\frac{\theta t}{\alpha}+\beta-\beta\exp(\frac{t}{\alpha})\right\}$		α	β	θ

The threshold parameter for the three-parameter Gompertz distribution is known as the Makeham mortality component and is labeled as such in the output. This component represents an additive risk that is constant over time. The exponential distribution shown in Table 18.57 is a special case of the Weibull distribution with $\beta = 1$. The remaining distributions in Table 18.57 are related to one another as shown in Table 18.58. The threshold parameter, θ, is assumed to be 0 in Table 18.58.

Table 18.58 Relationship among Life Distributions

Distribution of T	Parameters		Distribution of Y=logT	Parameters	
Lognormal	μ	σ	Normal	μ	σ
Weibull	α	β	Extreme value	$\mu = \log\alpha$	$\sigma = \frac{1}{\beta}$
Log-logistic	μ	σ	Logistic	μ	σ

MODEL Statement

All the distributions in Table 18.57 except the three-parameter Weibull and the Gompertz distributions are available for regression model estimation by using the MODEL statement. In addition, you can fit regression models with the generalized gamma distribution with the following probability density function $f(t)$:

$$f(t) = \frac{|\lambda|}{t\sigma\Gamma(\lambda^{-2})}(\lambda^{-2})^{(\lambda^{-2})}\exp\left[\lambda^{-2}\left(\lambda\left(\frac{\log(t)-\mu}{\sigma}\right) - \exp\left(\lambda\left(\frac{\log(t)-\mu}{\sigma}\right)\right)\right)\right]$$

If a lifetime T has the generalized gamma distribution, then the logarithm of the lifetime $X = \log(T)$ has the generalized log-gamma distribution, with the following probability density function $g(x)$. When the gamma distribution is specified, the logarithms of the lifetimes are used as responses, and the generalized log-gamma distribution is used to estimate the parameters by maximum likelihood.

$$g(x) = \frac{|\lambda|}{\sigma\Gamma(\lambda^{-2})}(\lambda^{-2})^{(\lambda^{-2})}\exp\left[\lambda^{-2}\left(\lambda\left(\frac{x-\mu}{\sigma}\right) - \exp\left(\lambda\left(\frac{x-\mu}{\sigma}\right)\right)\right)\right]$$

See Lawless (2003) and Meeker and Escobar (1998) for a description of the generalized gamma and generalized log-gamma distributions.

When $\lambda = 1$, the generalized log-gamma distribution reduces to the extreme value distribution with parameters μ and σ. In this case, the log lifetimes have the extreme value distribution, or, equivalently, the lifetimes have the Weibull distribution with parameters $\alpha = \exp(\mu)$ and $\beta = 1/\sigma$. When $\lambda = 0$, the generalized log-gamma reduces to the normal distribution with parameters μ and σ. In this case, the (unlogged) lifetimes have the lognormal distribution with parameters μ and σ. This chapter uses the notation μ for the *location*, σ for the *scale*, and λ for the *shape* parameters for the generalized log-gamma distribution.

ANALYZE Statement

You can use the ANALYZE statement to compute parameter estimates and other statistics for the distributions in Table 18.57. In addition, you can compute estimates for the binomial and Poisson distributions. The forms of these distributions are shown in Table 18.59.

Table 18.59 Binomial and Poisson Distributions

Distribution	Pr{Y=y}	Parameter	Parameter Name
Binomial	$\binom{n}{y} p^y (1-p)^{n-y}$	p	binomial probability
Poisson	$\frac{\mu^y}{y!} \exp(-\mu)$	μ	Poisson mean

Probability Plotting

Probability plots are useful tools for the display and analysis of lifetime data. See Abernethy (2006) for examples that use probability plots in the analysis of reliability data. Probability plots use a special scale so that a cumulative distribution function (CDF) plots as a straight line. Thus, if lifetime data are a sample from a distribution, the CDF estimated from the data plots approximately as a straight line on a probability plot for the distribution.

You can use the RELIABILITY procedure to construct probability plots for data that are complete, right censored, or interval censored (in readout form) for each of the probability distributions in Table 18.57.

A random variable Y belongs to a *location-scale* family of distributions if its CDF F is of the form

$$Pr\{Y \leq y\} = F(y) = G\left(\frac{y-\mu}{\sigma}\right)$$

where μ is the location parameter, and σ is the scale parameter. Here, G is a CDF that cannot depend on any unknown parameters, and G is the CDF of Y if $\mu = 0$ and $\sigma = 1$. For example, if Y is a normal random variable with mean μ and standard deviation σ,

$$G(u) = \Phi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du$$

and

$$F(y) = \Phi\left(\frac{y-\mu}{\sigma}\right)$$

Of the distributions in Table 18.57, the normal, extreme value, and logistic distributions are location-scale models. As shown in Table 18.58, if T has a lognormal, Weibull, or log-logistic distribution, then $\log(T)$ has a distribution that is a location-scale model. Probability plots are constructed for lognormal, Weibull, and log-logistic distributions by using $\log(T)$ instead of T in the plots.

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample with distribution function $F(y)$. A probability plot is a plot of the points $y_{(i)}$ against $m_i = G^{-1}(a_i)$, where $a_i = \hat{F}(y_i)$ is an estimate of the CDF $F(y_{(i)}) = G\left(\frac{y_{(i)}-\mu}{\sigma}\right)$. The points a_i are called *plotting positions*. The axis on which the points m_is are plotted is usually labeled with a probability scale (the scale of a_i).

If F is one of the location-scale distributions, then y is the lifetime; otherwise, the log of the lifetime is used to transform the distribution to a location-scale model.

If the data actually have the stated distribution, then $\hat{F} \approx F$,

$$m_i = G^{-1}(\hat{F}(y_i)) \approx G^{-1}\left(G\left(\frac{y_{(i)} - \mu}{\sigma}\right)\right) = \frac{y_{(i)} - \mu}{\sigma}$$

and points $(y_{(i)}, m_i)$ should fall approximately on a straight line.

There are several ways to compute plotting positions from failure data. These are discussed in the next two sections.

Complete and Right-Censored Data

The censoring times must be taken into account when you compute plotting positions for right-censored data. The RELIABILITY procedure provides several methods for computing plotting positions. These are specified with the PPOS= option in the ANALYZE, PROBPLOT, and RELATIONPLOT statements. All of the methods give similar results, as illustrated in the section "Expected Ranks, Kaplan-Meier, and Modified Kaplan-Meier Methods" on page 1344, the section "Nelson-Aalen" on page 1346, and the section "Median Ranks" on page 1346.

Expected Ranks, Kaplan-Meier, and Modified Kaplan-Meier Methods

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. Order the data in increasing order. Label all the data with reverse ranks r_i, with $r_1 = n, \ldots, r_n = 1$. For the failure corresponding to reverse rank r_i, compute the reliability, or survivor function estimate

$$R_i = \left[\frac{r_i}{r_i + 1}\right] R_{i-1}$$

with $R_0 = 1$. The expected rank plotting position is computed as $a_i = 1 - R_i$. The option PPOS=EXPRANK specifies the expected rank plotting position.

For the Kaplan-Meier method,

$$R_i = \left[\frac{r_i - 1}{r_i}\right] R_{i-1}$$

The Kaplan-Meier plotting position is then computed as $a'_i = 1 - R_i$. The option PPOS=KM specifies the Kaplan-Meier plotting position.

For the modified Kaplan-Meier method, use

$$R'_i = \frac{R_i + R_{i-1}}{2}$$

where R_i is computed from the Kaplan-Meier formula with $R_0 = 1$. The plotting position is then computed as $a''_i = 1 - R'_i$. The option PPOS=MKM specifies the modified Kaplan-Meier plotting position. If the PPOS option is not specified, the modified Kaplan-Meier plotting position is used as the default method.

For complete samples, $a_i = i/(n+1)$ for the expected rank method, $a'_i = i/n$ for the Kaplan-Meier method, and $a''_i = (i - .5)/n$ for the modified Kaplan-Meier method. If the largest observation is a failure for the Kaplan-Meier estimator, then $F_n = 1$ and the point is not plotted. These three methods are shown for the field winding data in Table 18.60 and Table 18.61.

Table 18.60 Expected Rank Plotting Position Calculations

Ordered Observation	Reverse Rank	$r_i/(r_i+1)$	$\times R_{i-1}$	$= R_i$	$a_i = 1 - R_i$
31.7	16	16/17	1.0000	0.9411	0.0588
39.2	15	15/16	0.9411	0.8824	0.1176
57.5	14	14/15	0.8824	0.8235	0.1765
65.0+	13				
65.8	12	12/13	0.8235	0.7602	0.2398
70.0	11	11/12	0.7602	0.6968	0.3032
75.0+	10				
75.0+	9				
87.5+	8				
88.3+	7				
94.2+	6				
101.7+	5				
105.8	4	4/5	0.6968	0.5575	0.4425
109.2+	3				
110.0	2	2/3	0.5575	0.3716	0.6284
130.0+	1				

+ Censored Times

Table 18.61 Kaplan-Meier and Modified Kaplan-Meier Plotting Position Calculations

Ordered Observation	Reverse Rank	$(r_i-1)/r_i$	$\times R_{i-1}$	$= R_i$	$a_i' = 1 - R_i$	a_i''
31.7	16	15/16	1.0000	0.9375	0.0625	0.0313
39.2	15	14/15	0.9375	0.8750	0.1250	0.0938
57.5	14	13/14	0.8750	0.8125	0.1875	0.1563
65.0+	13					
65.8	12	11/12	0.8125	0.7448	0.2552	0.2214
70.0	11	10/11	0.7448	0.6771	0.3229	0.2891
75.0+	10					
75.0+	9					
87.5+	8					
88.3+	7					
94.2+	6					
101.7+	5					
105.8	4	3/4	0.6771	0.5078	0.4922	0.4076
109.2+	3					
110.0	2	1/2	0.5078	0.2539	0.7461	0.6192
130.0+	1					

+ Censored Times

Nelson-Aalen

Estimate the cumulative hazard function by

$$H_i = \frac{1}{r_i} + H_{i-1}$$

with $H_0 = 0$. The reliability is $R_i = \exp(-H_i)$, and the plotting position, or CDF, is $a_i''' = 1 - R_i$. You can show that $R_{KM} < R_{NA}$ for all ages. The Nelson-Aalen method is shown for the field winding data in Table 18.62.

Table 18.62 Nelson-Aalen Plotting Position Calculations

Ordered Observation	Reverse Rank	$1/r_i$	$+H_{i-1}$	$= H_i$	$a_i''' = 1 - \exp(-H_i)$
31.7	16	1/16	0.0000	0.0625	0.0606
39.2	15	1/15	0.0625	0.1292	0.1212
57.5	14	1/14	0.1292	0.2006	0.1818
65.0+	13				
65.8	12	1/12	0.2006	0.2839	0.2472
70.0	11	1/11	0.2839	0.3748	0.3126
75.0+	10				
75.0+	9				
87.5+	8				
88.3+	7				
94.2+	6				
101.7+	5				
105.8	4	1/4	0.3748	0.6248	0.4647
109.2+	3				
110.0	2	1/2	0.6248	1.1248	0.6753
130.0+	1				

+ Censored Times

Median Ranks

See Abernethy (2006) for a discussion of the methods described in this section. Let $y_{(1)} \le y_{(2)} \le \ldots \le y_{(n)}$ be ordered observations of a random sample including failure times and censor times. A failure order number j_i is assigned to the ith failure: $j_i = j_{i-1} + \Delta$, where $j_0 = 0$. The increment Δ is initially 1 and is modified when a censoring time is encountered in the ordered sample. The new increment is computed as

$$\Delta = \frac{(n+1) - \text{ previous failure order number}}{1 + \text{ number of items beyond previous censored item}}$$

The plotting position is computed for the ith failure time as

$$a_i = \frac{j_i - .3}{n + .4}$$

For complete samples, the failure order number j_i is equal to i, the order of the failure in the sample. In this case, the preceding equation for a_i is an approximation to the median plotting position computed as the median of the ith-order statistic from the uniform distribution on (0, 1). In the censored case, j_i is not necessarily an integer, but the preceding equation still provides an approximation to the median plotting position. The PPOS=MEDRANK option specifies the median rank plotting position.

For complete data, an alternative method of computing the median rank plotting position for failure i is to compute the exact median of the distribution of the ith order statistic of a sample of size n from the uniform distribution on (0,1). If the data are right censored, the adjusted rank j_i, as defined in the preceding paragraph, is used in place of i in the computation of the median rank. The PPOS=MEDRANK1 option specifies this type of plotting position.

Nelson (1982, p. 148) provides the following example of multiply right-censored failure data for field windings in electrical generators. Table 18.63 shows the data, the intermediate calculations, and the plotting positions calculated by exact (a'_i) and approximate (a_i) median ranks.

Table 18.63 Median Rank Plotting Position Calculations

Ordered Observation	Increment Δ	Failure Order Number j_i	a_i	a'_i
31.7	1.0000	1.0000	0.04268	0.04240
39.2	1.0000	2.0000	0.1037	0.1027
57.5	1.0000	3.0000	0.1646	0.1637
65.0+	1.0769			
65.8	1.0769	4.0769	0.2303	0.2294
70.0	1.0769	5.1538	0.2960	0.2953
75.0+	1.1846			
75.0+	1.3162			
87.5+	1.4808			
88.3+	1.6923			
94.2+	1.9744			
101.7+	2.3692			
105.8	2.3692	7.5231	0.4404	0.4402
109.2+	3.1590			
110.0	3.1590	10.6821	0.6331	0.6335
130.0+	6.3179			

+ Censored Times

Interval-Censored Data

Readout Data

The RELIABILITY procedure can create probability plots for interval-censored data when all units share common interval endpoints. This type of data is called *readout* data in the RELIABILITY procedure. Estimates of the cumulative distribution function are computed at times corresponding to the interval endpoints. Right censoring can also be accommodated if the censor times correspond to interval endpoints. See the section "Weibull Analysis of Interval Data with Common Inspection Schedule" on page 1219 for an example of a Weibull plot and analysis for interval data.

Table 18.64 illustrates the computational scheme used to compute the CDF estimates. The data are failure data for microprocessors (Nelson 1990, p. 147). In Table 18.64, t_i are the interval upper endpoints, in hours, f_i is the number of units failing in interval i, and n_i is the number of unfailed units at the beginning of interval i.

Note that there is right censoring as well as interval censoring in these data. For example, two units fail in the interval (24, 48) hours, and there are 1414 unfailed units at the beginning of the interval, 24 hours. At the beginning of the next interval, (48, 168) hours, there are 573 unfailed units. The number of unfailed units that are removed from the test at 48 hours is $1414 - 2 - 573 = 839$ units. These are right-censored units.

The reliability at the end of interval i is computed recursively as

$$R_i = (1 - (f_i/n_i))R_{i-1}$$

with $R_0 = 1$. The plotting position is $a_i = 1 - R_i$.

Table 18.64 Interval-Censored Plotting Position Calculations

Interval i	Interval Endpoint t_i	f_i/n_i	$R_i^{'} = 1 - (f_i/ni)$	$R_i = R_i^{'}R_{i-1}$	$a_i = 1 - R_i$
1	6	6/1423	0.99578	0.99578	.00421
2	12	2/1417	0.99859	0.99438	.00562
3	24	0/1415	1.00000	0.99438	.00562
4	48	2/1414	0.99859	0.99297	.00703
5	168	1/573	0.99825	0.99124	.00876
6	500	1/422	0.99763	0.98889	.01111
7	1000	2/272	0.99265	0.98162	.01838
8	2000	1/123	0.99187	0.97364	.02636

Arbitrarily Censored Data

The RELIABILITY procedure can create probability plots for data that consists of combinations of exact, left-censored, right-censored, and interval-censored lifetimes. Unlike the method in the previous section, failure intervals need not share common endpoints, although if the intervals share common endpoints, the two methods give the same results. The RELIABILITY procedure uses an iterative algorithm developed by Turnbull (1976) to compute a nonparametric maximum likelihood estimate of the cumulative distribution function for the data. Since the technique is maximum likelihood, standard errors of the cumulative probability estimates are computed from the inverse of the associated Fisher information matrix. A technique developed by Gentleman and Geyer (1994) is used to check for convergence to the maximum likelihood estimate. Also see Meeker and Escobar (1998, chap. 3) for more information.

Although this method applies to more general situations, where the intervals may be overlapping, the example of the previous section will be used to illustrate the method. Table 18.65 contains the microprocessor data of the previous section, arranged in intervals. A missing (.) lower endpoint indicates left censoring, and a missing upper endpoint indicates right censoring. These can be thought of as semi-infinite intervals with lower (upper) endpoint of negative (positive) infinity for left (right) censoring.

Table 18.65 Interval-Censored Data

Lower Endpoint	Upper Endpoint	Number Failed
.	6	6
6	12	2
24	48	2
24	.	1
48	168	1
48	.	839
168	500	1
168	.	150
500	1000	2
500	.	149
1000	2000	1
1000	.	147
2000	.	122

The following SAS statements compute the Turnbull estimate and create a lognormal probability plot:

```
data micro;
   input t1 t2 f ;
   datalines;
. 6 6
6 12 2
12 24 0
24 48 2
24 .  1
48 168 1
48 .   839
168 500 1
168 .   150
500 1000 2
500 .    149
1000 2000 1
1000 . 147
2000 . 122
;

proc reliability data=micro;
   distribution lognormal;
   freq f;
   pplot ( t1 t2 ) / itprintem
                     printprobs
                     maxitem = ( 1000, 25 )
                     nofit
                     npintervals = simul
                     ppout;
run;
```

The nonparametric maximum likelihood estimate of the CDF can only increase on certain intervals, and must remain constant between the intervals. The Turnbull algorithm first computes the intervals on which the nonparametric maximum likelihood estimate of the CDF can increase. The algorithm then iteratively estimates the probability associated with each interval. The ITPRINTEM option along with the PRINTPROBS option instructs the procedure to print the intervals on which probability increases can occur and the iterative history of the estimates of the interval probabilities. The PPOUT option requests tabular output of the estimated CDF, standard errors, and confidence limits for each cumulative probability.

Figure 18.55 shows every 25th iteration and the last iteration for the Turnbull estimate of the CDF for the microprocessor data. The initial estimate assigns equal probabilities to each interval. You can specify different initial values with the PROBLIST= option. The algorithm converges in 130 iterations for this data. Convergence is determined if the change in the loglikelihood between two successive iterations less than Δ, where the default value is $\Delta = 10^{-8}$. You can specify a different value for delta with the TOLLIKE= option. This algorithm is an example of an expectation-maximization (EM) algorithm. EM algorithms are known to converge slowly, but the computations within each iteration for the Turnbull algorithm are moderate. Iterations will be terminated if the algorithm does not converge after a fixed number of iterations. The default maximum number of iterations is 1000. Some data may require more iterations for convergence. You can specify the maximum allowed number of iterations with the MAXITEM= option in the PROBPLOT, ANALYZE, or RPLOT statement.

Figure 18.55 Iteration History for Turnbull Estimate

The RELIABILITY Procedure

Iteration History for the Turnbull Estimate of the CDF

Iteration	Loglikelihood	(., 6)	(6, 12)	(24, 48)	(48, 168)	(168, 500)	(500, 1000)	(1000, 2000)	(2000, .)
0	-1133.4051	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
25	-104.16622	0.00421644	0.00140548	0.00140648	0.00173338	0.00237846	0.00846094	0.04565407	0.93474475
50	-101.15151	0.00421644	0.00140548	0.00140648	0.00173293	0.00234891	0.00727679	0.01174486	0.96986811
75	-101.06641	0.00421644	0.00140548	0.00140648	0.00173293	0.00234891	0.00727127	0.00835638	0.9732621
100	-101.06534	0.00421644	0.00140548	0.00140648	0.00173293	0.00234891	0.00727125	0.00801814	0.97360037
125	-101.06533	0.00421644	0.00140548	0.00140648	0.00173293	0.00234891	0.00727125	0.00798438	0.97363413
130	-101.06533	0.00421644	0.00140548	0.00140648	0.00173293	0.00234891	0.00727125	0.007983	0.97363551

If an interval probability is smaller than a tolerance (10^{-6} by default) after convergence, the probability is set to zero, the interval probabilities are renormalized so that they add to one, and iterations are restarted. Usually the algorithm converges in just a few more iterations. You can change the default value of the tolerance with the TOLPROB= option. You can specify the NOPOLISH option to avoid setting small probabilities to zero and restarting the algorithm.

If you specify the ITPRINTEM option, the table in Figure 18.56 summarizing the Turnbull estimate of the interval probabilities is printed. The columns labeled 'Reduced Gradient' and 'Lagrange Multiplier' are used in checking final convergence to the maximum likelihood estimate. The Lagrange multipliers must all be greater than or equal to zero, or the solution is not maximum likelihood. See Gentleman and Geyer (1994) for more details of the convergence checking.

Figure 18.56 Final Probability Estimates for Turnbull Algorithm

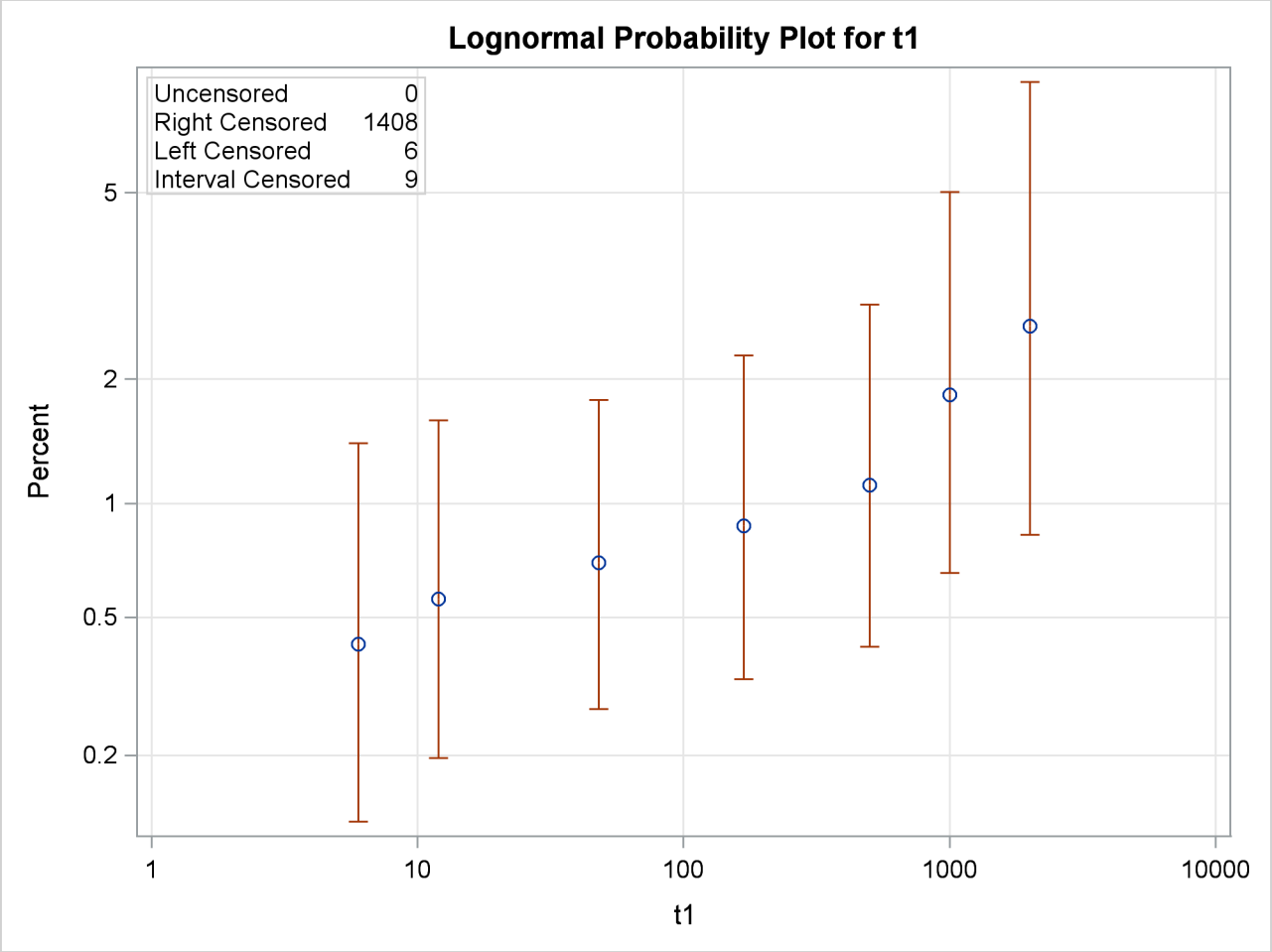
Lower Lifetime	Upper Lifetime	Probability	Reduced Gradient	Lagrange Multiplier
.	6	0.0042	0	0
6	12	0.0014	0	0
24	48	0.0014	0	0
48	168	0.0017	0	0
168	500	0.0023	0	0
500	1000	0.0073	-7.219342E-9	0
1000	2000	0.0080	-0.037063236	0
2000	.	0.9736	0.0003038877	0

Figure 18.57 shows the final estimate of the CDF, along with standard errors and confidence limits. Figure 18.58 shows the CDF and simultaneous confidence limits plotted on a lognormal probability plot.

Figure 18.57 Final CDF Estimates for Turnbull Algorithm

Cumulative Probability Estimates					
			Pointwise 95% Confidence Limits		
Lower Lifetime	Upper Lifetime	Cumulative Probability	Lower	Upper	Standard Error
6	6	0.0042	0.0019	0.0094	0.0017
12	24	0.0056	0.0028	0.0112	0.0020
48	48	0.0070	0.0038	0.0130	0.0022
168	168	0.0088	0.0047	0.0164	0.0028
500	500	0.0111	0.0058	0.0211	0.0037
1000	1000	0.0184	0.0094	0.0357	0.0063
2000	2000	0.0264	0.0124	0.0553	0.0101

Figure 18.58 Lognormal Probability Plot for the Microprocessor Data



Nonparametric Confidence Intervals for Cumulative Failure Probabilities

The method used in the RELIABILITY procedure for computation of approximate pointwise and simultaneous confidence intervals for cumulative failure probabilities relies on the Kaplan-Meier estimator of the cumulative distribution function of failure time and approximate standard deviation of the Kaplan-Meier estimator. For the case of arbitrarily censored data, the Turnbull algorithm, discussed previously, provides an extension of the Kaplan-Meier estimator.

For multiply censored data, the Kaplan-Meier estimator of the cumulative distribution function at failure time t_i is $\hat{F}(t_i) = 1 - \hat{S}(t_i)$, where

$$\hat{S}(t_i) = \prod_{j=1}^{i} (1 - \hat{p}_j),$$

$$\hat{p}_i = \frac{d_i}{n_i},$$

d_i is the number of failures in the interval (t_{i-1}, t_i) , and n_i is the number of unfailed units at the beginning of the interval. This definition of the Kaplan-Meier estimator is equivalent to the one previously given.

An estimator of the variance v_i of the Kaplan-Meier estimator $\hat{F}(t_i)$ is given by

$$\hat{v}_i = [\hat{S}(t_i)]^2 \sum_{j=1}^{i} \frac{\hat{p}_j}{n_j(1-\hat{p}_j)}$$

An estimator of the standard deviation of $\hat{F}(t_i)$ is $\text{se}_{\hat{F}} = \sqrt{\hat{v}_i}$.

For arbitrarily censored data, the Kaplan-Meier estimator is replaced by the nonparametric maximum likelihood estimator computed with the Turnbull algorithm, and the approximate variance of the estimator of $F(t_i)$ is computed from the inverse of the Fisher information matrix.

Pointwise Confidence Intervals

Approximate $(1-\alpha)100\%$ pointwise confidence intervals are computed as in Meeker and Escobar (1998, section 3.6) as

$$[F_L, F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right]$$

where

$$w = \exp\left[\frac{z_{1-\alpha/2}\text{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right]$$

where z_p is the pth quantile of the standard normal distribution.

Simultaneous Confidence Intervals

Approximate $(1-\alpha)100\%$ simultaneous confidence bands valid over the lifetime interval (t_a, t_b) are computed as the "Equal Precision" case of Nair (1984) and Meeker and Escobar (1998, section 3.8)

$$[F_L, F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right]$$

where

$$w = \exp\left[\frac{e_{a,b,1-\alpha/2}\text{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right]$$

where the factor $x = e_{a,b,1-\alpha/2}$ is the solution of

$$x \exp(-x^2/2) \log\left[\frac{(1-a)b}{(1-b)a}\right] / \sqrt{8\pi} = \alpha/2$$

The time interval (t_a, t_b) over which the bands are valid depends in a complicated way on the constants a and b defined in Nair (1984), $0 < a < b < 1$. a and b are chosen by default, so that the confidence bands are valid between the lowest and highest times corresponding to failures in the case of multiply censored data, or, to the lowest and highest intervals for which probabilities are computed for arbitrarily censored data. You can optionally specify a and b directly with the NPINTERVALS=SIMULTANEOUS(a,b) option in the PROBPLOT statement.

Parameter Estimation and Confidence Intervals

Maximum Likelihood Estimation

Maximum likelihood estimation of the parameters of a statistical model involves maximizing the likelihood or, equivalently, the log likelihood with respect to the parameters. The parameter values at which the maximum occurs are the maximum likelihood estimates of the model parameters. The likelihood is a function of the parameters and of the data.

Let $x_1, x_2, \ldots, x_n$ be the observations in a random sample, including the failures and censoring times (if the data are censored). Let $f(\boldsymbol{\theta}; x)$ be the probability density of failure time, $S(\boldsymbol{\theta}; x) = Pr\{X \geq x\}$ be the reliability function, and $F(\boldsymbol{\theta}; x) = Pr\{X \leq x\}$ be the cumulative distribution function, where $\boldsymbol{\theta}$ is the vector of parameters to be estimated, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)$. The probability density, reliability function, and CDF are determined by the specific distribution selected as a model for the data. The log likelihood is defined as

$$\begin{aligned} L(,) \quad = \quad & \sum_i \log(f(\boldsymbol{\theta}; x_i)) + \sum_i{}' \log(S(\boldsymbol{\theta}; x_i)) + \\ & \sum_i{}'' \log(F(\boldsymbol{\theta}; x_i)) + \sum_i{}''' [\log(F(\boldsymbol{\theta}; x_{ui}) - F(\boldsymbol{\theta}; x_{li}))] \end{aligned}$$

where

- $\sum$ is the sum over failed units
- $\sum'$ is the sum over right-censored units
- $\sum''$ is the sum over left-censored units
- $\sum'''$ is the sum over interval-censored units

and (x_{li}, x_{ui}) is the interval in which the ith unit is interval censored. Only the sums appropriate to the type of censoring in the data are included when the preceding equation is used.

The RELIABILITY procedure maximizes the log likelihood with respect to the parameters $\boldsymbol{\theta}$ by using a Newton-Raphson algorithm. The Newton-Raphson algorithm is a recursive method for computing the maximum of a function. On the rth iteration, the algorithm updates the parameter vector $\boldsymbol{\theta}_r$ with

$$\boldsymbol{\theta}_{r+1} = \boldsymbol{\theta}_r - \mathbf{H}^{-1}\mathbf{g}$$

where $\mathbf{H}$ is the Hessian (second derivative) matrix, and $\mathbf{g}$ is the gradient (first derivative) vector of the log-likelihood function, both evaluated at the current value of the parameter vector. That is,

$$\mathbf{g} = [g_j] = \left[\frac{\partial L}{\partial \theta_j}\right]\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r}$$

and

$$\mathbf{H} = [h_{ij}] = \left[\frac{\partial^2 L}{\partial \theta_i \partial \theta_j}\right]\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r}$$

Iteration continues until the parameter estimates converge. The convergence criterion is

$$|\theta_i^{r+1} - \theta_i^r| \le c \quad \text{if} \quad |\theta_i^{r+1}| < 0.01$$

$$\left|\frac{\theta_i^{r+1} - \theta_i^r}{\theta_i^{r+1}}\right| \le c \quad \text{if} \quad |\theta_i^{r+1}| \ge 0.01$$

for all $i = 1, 2, \ldots, p$ where c is the convergence criterion. The default value of c is 0.001, and it can be specified with the CONVERGE= option in the MODEL, PROBPLOT, RELATIONPLOT, and ANALYZE statements.

After convergence by the preceding criterion, the quantity

$$tc = \frac{\mathbf{g}\mathbf{H}^{-1}\mathbf{g}}{L}$$

is computed. If $tc > d$ then a warning is printed that the algorithm did not converge. tc is called the *relative Hessian* convergence criterion. The default value of d is 0.0001. You can specify other values for d with the CONVH= option. The relative Hessian criterion is useful in detecting the occasional case where no progress can be made in increasing the log likelihood, yet the gradient $\mathbf{g}$ is not zero.

A location-scale model has a CDF of the form

$$F(x) = G\left(\frac{x-\mu}{\sigma}\right)$$

where μ is the location parameter, σ is the scale parameter, and G is a standardized form ($\mu = 0, \sigma = 1$) of the cumulative distribution function. The parameter vector is $\boldsymbol{\theta}$=(μ σ). It is more convenient computationally to maximize log likelihoods that arise from location-scale models. If you specify a distribution from Table 18.57 that is not a location-scale model, it is transformed to a location-scale model by taking the natural

(base e) logarithm of the response. If you specify the lognormal base 10 distribution, the logarithm (base 10) of the response is used. The Weibull, lognormal, and log-logistic distributions in Table 18.57 are not location-scale models. Table 18.58 shows the corresponding location-scale models that result from taking the logarithm of the response.

Maximum likelihood is the default method of estimating the location and scale parameters in the MODEL, PROBPLOT, RELATIONPLOT, and ANALYZE statements. If the Weibull distribution is specified, the logarithms of the responses are used to obtain maximum likelihood estimates ($\hat{\mu}$ $\hat{\sigma}$) of the location and scale parameters of the extreme value distribution. The maximum likelihood estimates ($\hat{\alpha}$, $\hat{\beta}$) of the Weibull scale and shape parameters are computed as $\hat{\alpha} = \exp(\hat{\mu})$ and $\hat{\beta} = 1/\hat{\sigma}$.

Maximum likelihood estimates for the Gompertz distributions are obtained by expressing the log-likelihood in terms of $\log(\alpha)$, $\log(\beta)$, and (if applicable) $\log(\theta)$. After the log likelihood is maximized, parameter estimates and their standard errors are transformed from the logarithm metric to the standard metric by using the delta method.

Three-Parameter Weibull

The parameters of the three-parameter Weibull distribution are estimated by maximizing the log likelihood function. The threshold parameter θ must be less than the minimum failure time t_0, unless $\beta = 1$, in which case, θ can be equal to t_0. The RELIABILITY procedure sets a default upper bound of $t_0 - 0.001$ for the threshold in the iterative estimation computations and a default lower bound of 0.0. You can set different bounds by specifying an INEST data set as described in the section "INEST Data Set for the Three-Parameter Weibull" on page 1356.

If the shape parameter β is less than one, then the density function in Table 18.57 has a singularity at $t = \theta$, and the log likelihood is unbounded above as the threshold parameter approaches the minimum failure time t_0. For any fixed $\theta < t_0$, maximum likelihood estimates of the scale and shape parameters α and β exist. If $\hat{\beta} < 1$ in the iterative estimation procedure, the estimate of the threshold θ is set to the upper bound and maximum likelihood estimates of α and β are computed.

INEST Data Set for the Three-Parameter Weibull

You can specify a SAS data set to set lower bounds, upper bounds, equality constraints, or initial values for estimating the parameters of a three-parameter Weibull distribution by using the INEST= option in the ANALYZE or PROBPLOT statement. The data set must contain a variable named _TYPE_ that specifies the action that you want to take in the iterative estimation process, and some combination of variables named _SCALE_, _SHAPE_, and _THRESHOLD_ that represent the distribution parameters. If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group.

The possible values of _TYPE_ and corresponding actions are summarized in Table 18.66.

Table 18.66 _TYPE_ Variable Values

Value of _TYPE_	Action
LB	Lower bound
UB	Upper bound
EQ	Equality
PARMS	Initial value

For example, you can use the INEST data set In created by using the following SAS statements to specify bounds for data that contain the BY variable Group with three BY groups: A, B, and D. The data set In specifies a lower bound for the threshold parameter of –100 for groups A, B, and D, and an upper bound of 3 for the threshold parameter for group D. Since the variables _Scale_ and _Shape_ are set to missing, no action is taken for them, and these variables could be omitted from the data set.

```
data In;
   input Group$1 _Type_$ 2-11 _Scale_ _Shape_ _Threshold_;
   datalines;
A    lb    . . -100
B    lb    . . -100
D    lb    . . -100
D    ub    . .  3
;
```

Regression Models

You can specify a regression model by using the MODEL statement. For example, if you want to relate the lifetimes of electronic parts in a test to Arrhenius-transformed operating temperature, then an appropriate model might be

$$\mu_i = \beta_0 + x_i \beta_1$$

where $x_i = 1000/(T_i + 273.15)$, and T_i is the centigrade temperature at which the ith unit is tested. Here, $\mathbf{x}_i' = [\ 1\ x_i\ ]$.

There are two types of explanatory variables: *continuous* variables and *classification* variables. Continuous variables represent physical quantities, such as temperature or voltage, and they must be numeric. Continuous explanatory variables are sometimes called *covariates*.

Classification variables identify classification levels and are declared in the CLASS statement. These are also referred to as *categorical, dummy, qualitative, discrete,* or *nominal* variables. Classification variables can be either character or numeric. The values of classification variables are called *levels*. For example, the classification variable Batch could have levels 'batch1' and 'batch2' to identify items from two production batches. An indicator (0-1) variable is generated for each level of a classification variable and is used as an explanatory variable. See Nelson (1990, p. 277) for an example that uses an indicator variable in the analysis of accelerated life test data. In a model, an explanatory variable that is not declared in a CLASS statement is assumed to be continuous.

By default, all regression models automatically contain an intercept term; that is, the model is of the form

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \ldots$$

where β_0 does not have an explanatory variable multiplier. The intercept term can be excluded from the model by specifying INTERCEPT=0 as a MODEL statement option.

For numerical stability, continuous explanatory variables are centered and scaled internally to the procedure. This transforms the parameters $\boldsymbol{\beta}$ in the original model to a new set of parameters. The parameter estimates $\boldsymbol{\beta}$ and covariances are transformed back to the original scale before reporting, so that the parameters should be

interpreted in terms of the originally specified model. Covariates that are indicator variables—that is, those specified in a CLASS statement—are not centered and scaled.

Initial values of the regression parameters used in the Newton-Raphson method are computed by ordinary least squares. The parameters $\boldsymbol{\beta}$ and the scale parameter σ are jointly estimated by maximum likelihood, taking a logarithmic transformation of the responses, if necessary, to get a location-scale model.

The generalized gamma distribution is fit using log lifetime as the response variable. The regression parameters $\boldsymbol{\beta}$, the scale parameter σ, and the shape parameter λ are jointly estimated.

The Weibull distribution shape parameter estimate is computed as $\hat{\beta} = 1/\hat{\sigma}$, where σ is the scale parameter from the corresponding extreme value distribution. The Weibull scale parameter $\hat{\alpha}_i = \exp(\mathbf{x}'\hat{\boldsymbol{\beta}})$ is not computed by the procedure. Instead, the regression parameters $\boldsymbol{\beta}$ and the shape β are reported.

In a model with one to three continuous explanatory variables x, you can use the RELATION= option in the MODEL statement to specify a transformation that is applied to the variables before model fitting. Table 18.67 shows the available transformations.

Table 18.67 Variable Transformations

Relation	Transformed variable
ARRHENIUS (Nelson parameterization)	$1000/(x + 273.15)$
ARRHENIUS2 (activation energy parameterization)	$11605/(x + 273.15)$
POWER	$\log(x), \quad x > 0$
LINEAR	x
LOGISTIC	$\log\left(\frac{x}{1-x}\right), \quad 0 < x < 1$

Nonconstant Scale Parameter

In some situations, it is desirable for the scale parameter to change with the values of explanatory variables. For example, Meeker and Escobar (1998, section 17.5) present an analysis of accelerated life test data where the spread of the data is greater at lower levels of the stress. You can use the LOGSCALE statement to specify the scale parameter as a function of explanatory variables. You must also have a MODEL statement to specify the location parameter. Explanatory variables can be continuous variables, indicator variables specified in the CLASS statement, or any interaction combination. The variables can be the same as specified in the MODEL statement, or they can be different variables. Any transformation specified with the RELATION= MODEL statement option will be applied to the same variable appearing in the LOGSCALE statement. See the section "Regression Model with Nonconstant Scale" on page 1235 for an example of fitting a model with nonconstant scale parameter.

The form of the model for the scale parameter is

$$\log(\sigma_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$$

where β_0 is the intercept term. The intercept term can be excluded from the model by specifying INTERCEPT=0 as a LOGSCALE statement option.

The parameters $\beta_0, \beta_1, \ldots, \beta_p$ are estimated by maximum likelihood jointly with all the other parameters in the model.

Stable Parameters

The location and scale parameters (μ, σ) are estimated by maximizing the likelihood function by numerical methods, as described previously. An alternative parameterization that is likely to have better numerical properties for heavy censoring is (η, σ), where $\eta = \mu + z_p\sigma$ and z_p is the pth quantile of the standardized distribution. See Meeker and Escobar (1998, p. 90) and Doganaksoy and Schmee (1993) for more details on alternate parameterizations.

By default, RELIABILITY estimates a value of z_p from the data that will improve the numerical properties of the estimation. You can also specify values of p from which the value of z_p will be computed with the PSTABLE= option in the ANALYZE, PROBPLOT, RELATIONPLOT, or MODEL statement. Note that a value of $p = 0.632$ for the Weibull and extreme value and $p = 0.5$ for all other distributions will give $z_p = 0$ and the parameterization will then be the usual location-scale parameterization.

All estimates and related statistics are reported in terms of the location and scale parameters (μ, σ). If you specify the ITPRINT option in the ANALYZE, PROBPLOT, or RELATIONPLOT statement, a table showing the values of p, ν, σ, and the last evaluation of the gradient and Hessian for these parameters is produced.

Covariance Matrix

An estimate of the covariance matrix of the maximum likelihood estimators (MLEs) of the parameters $\boldsymbol{\theta}$ is given by the inverse of the negative of the matrix of second derivatives of the log likelihood, evaluated at the final parameter estimates:

$$\boldsymbol{\Sigma} = [\sigma_{ij}] = -\mathbf{H}^{-1} = -\left[\frac{\partial^2 L}{\partial\theta_i\,\partial\theta_j}\right]^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

The negative of the matrix of second derivatives is called the observed Fisher information matrix. The diagonal term σ_{ii} is an estimate of the variance of $\hat{\theta}_i$. Estimates of standard errors of the MLEs are provided by

$$\mathrm{SE}_{\theta_i} = \sqrt{\sigma_{ii}}$$

An estimator of the correlation matrix is

$$\mathbf{R} = \left[\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}\right]$$

The covariance matrix for the Weibull distribution parameter estimators is computed by a first-order approximation from the covariance matrix of the estimators of the corresponding extreme value parameters (μ, σ) as

$$\begin{aligned}
\mathrm{Var}(\hat{\alpha}) &= [\exp(\hat{\mu})]^2\mathrm{Var}(\hat{\mu}) \\
\mathrm{Var}(\hat{\beta}) &= \frac{\mathrm{Var}(\hat{\sigma})}{\hat{\sigma}^4} \\
\mathrm{Cov}(\hat{\alpha}, \hat{\beta}) &= -\frac{\exp(\hat{\mu})}{\hat{\sigma}^2}\mathrm{Cov}(\hat{\mu}, \hat{\sigma})
\end{aligned}$$

For the regression model, the variance of the Weibull shape parameter estimator $\hat{\beta}$ is computed from the variance of the estimator of the extreme value scale parameter σ as shown previously. The covariance of the regression parameter estimator $\hat{\beta}_i$ and the Weibull shape parameter estimator $\hat{\beta}$ is computed in terms of the covariance between $\hat{\beta}_i$ and $\hat{\sigma}$ as

$$\mathrm{Cov}(\hat{\beta}_i, \hat{\beta}) = -\frac{\mathrm{Cov}(\hat{\beta}_i, \hat{\sigma})}{\hat{\sigma}^2}$$

Confidence Intervals for Distribution Parameters

Table 18.68 shows the method of computation of approximate two-sided $\gamma \times 100\%$ confidence limits for distribution parameters. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. In Table 18.68, K_γ represents the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution, and $\hat{\mu}$ and $\hat{\sigma}$ are the MLEs of the location and scale parameters for the normal, extreme value, and logistic distributions. For the lognormal, Weibull, and log-logistic distributions, $\hat{\mu}$ and $\hat{\sigma}$ represent the MLEs of the corresponding location and scale parameters of the location-scale distribution that results when the logarithm of the lifetime is used as the response. For the Weibull distribution, μ and σ are the location and scale parameters of the extreme value distribution for the logarithm of the lifetime. $\mathrm{SE}_{\hat{\theta}}$ denotes the standard error of the MLE of θ, computed as the square root of the appropriate diagonal element of the inverse of the Fisher information matrix.

For the Gompertz distributions, estimation of all parameters takes place in the logarithm metric. For example, a confidence interval for the logarithm of scale is computed as $\log(\hat{\alpha}) \pm K_\gamma \mathrm{SE}_{\log(\hat{\alpha})}$. The confidence interval in the standard metric is then obtained by taking e to the power equal to each endpoint.

Table 18.68 Confidence Limit Computation

	Parameters		
Distribution	**Location μ or Threshold θ**	**Scale**	**Shape**
Normal	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Lognormal	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Lognormal (base 10)	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Extreme value	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Weibull		$\alpha_L = \exp[\hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})]$ $\alpha_U = \exp[\hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})]$	$\beta_L = \exp[-K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]/\hat{\sigma}$ $\beta_U = \exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]/\hat{\sigma}$
Exponential		$\alpha_L = \exp[\hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})]$ $\alpha_U = \exp[\hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})]$	
Logistic	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Log-logistic	$\mu_L = \hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})$ $\mu_U = \hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})$	$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	
Generalized Gamma		$\sigma_L = \hat{\sigma}/\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$ $\sigma_U = \hat{\sigma}\exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$	$\mu_L = \hat{\lambda} - K_\gamma(\mathrm{SE}_{\hat{\lambda}})$ $\mu_U = \hat{\lambda} + K_\gamma(\mathrm{SE}_{\hat{\lambda}})$
Three-parameter Weibull	$\theta_L = \hat{\theta} - K_\gamma(\mathrm{SE}_{\hat{\theta}})$ $\theta_U = \hat{\theta} + K_\gamma(\mathrm{SE}_{\hat{\theta}})$	$\alpha_L = \exp[\hat{\mu} - K_\gamma(\mathrm{SE}_{\hat{\mu}})]$ $\alpha_U = \exp[\hat{\mu} + K_\gamma(\mathrm{SE}_{\hat{\mu}})]$	$\beta_L = \exp[-K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]/\hat{\sigma}$ $\beta_U = \exp[K_\gamma(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]/\hat{\sigma}$

Regression Parameters Approximate $\gamma \times 100\%$ confidence limits for the regression parameter β_i are given by

$$\beta_{iL} = \hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$$

$$\beta_{iU} = \hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$$

Percentiles

The maximum likelihood estimate of the $p \times 100\%$ percentile x_p for the extreme value, normal, and logistic distributions is given by

$$\hat{x}_p = \hat{\mu} + z_p\hat{\sigma}$$

where $z_p = G^{-1}(p)$, G is the standardized CDF shown in Table 18.69, and $(\hat{\mu}, \hat{\sigma})$ are the maximum likelihood estimates of the location and scale parameters of the distribution. The maximum likelihood estimate of the percentile t_p for the Weibull, lognormal, and log-logistic distributions is given by

$$\hat{t}_p = \exp[\hat{\mu} + z_p\hat{\sigma}]$$

where $z_p = G^{-1}(p)$, and G is the standardized CDF of the location-scale model corresponding to the logarithm of the response. For the lognormal (base 10) distribution,

$$\hat{t}_p = 10^{[\hat{\mu}+z_p\hat{\sigma}]}$$

The maximum likelihood estimate of the percentile t_p for the three-parameter Weibull distribution is computed by

$$\hat{t}_p = \hat{\theta} + \exp[\hat{\mu} + z_p\hat{\sigma}]$$

where $z_p = G^{-1}(p)$, and G is the standardized CDF of extreme value distribution.

The maximum likelihood estimate of the percentile t_p for the standard Gompertz distribution is computed by

$$\hat{t}_p = \hat{\alpha}\log\{1 - \hat{\beta}^{-1}\log(1-p)\}$$

Because the quantile function depends on Lambert's W function and has no closed form, the percentile for the three-parameter Gompertz distribution is obtained by using a bisection algorithm to solve the following equation:

$$p = F(\hat{t}_p) = 1 - \exp\left\{\hat{\beta} - \frac{\hat{\theta}\hat{t}_p}{\hat{\alpha}} - \hat{\beta}\exp\left(\frac{\hat{t}_p}{\hat{\alpha}}\right)\right\}$$

Table 18.69 Standardized Cumulative Distribution Functions

Distribution	Location-Scale Distribution	Location-Scale CDF
Weibull	Extreme value	$1 - \exp[-\exp(z)]$
Lognormal	Normal	$\int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du$
Log-logistic	Logistic	$\frac{\exp(z)}{1+\exp(z)}$

Confidence Intervals The variance of the MLE of the $p \times 100\%$ percentile for the normal, extreme value, or logistic distribution is

$$\mathrm{Var}(\hat{x}_p) = \mathrm{Var}(\hat{\mu}) + z_p^2 \mathrm{Var}(\hat{\sigma}) + 2z_p \mathrm{Cov}(\hat{\mu}, \hat{\sigma})$$

Two-sided approximate $100\gamma\%$ confidence limits for x_p are

$$\begin{aligned} x_{pL} &= \hat{x}_p - K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)} \\ x_{pU} &= \hat{x}_p + K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)} \end{aligned}$$

where K_γ represents the $100(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution.

The limits for the lognormal, Weibull, or log-logistic distributions are

$$\begin{aligned} t_{pL} &= \exp\left(\hat{x}_p - K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)}\right) \\ t_{pU} &= \exp\left(\hat{x}_p + K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)}\right) \end{aligned}$$

where x_p refers to the percentile of the corresponding location-scale distribution (normal, extreme value, or logistic) for the logarithm of the lifetime. For the lognormal (base 10) distribution,

$$\begin{aligned} t_{pL} &= 10^{\left(\hat{x}_p - K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)}\right)} \\ t_{pU} &= 10^{\left(\hat{x}_p + K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)}\right)} \end{aligned}$$

Approximate limits for the three-parameter Weibull distribution are computed as

$$\begin{aligned} t_{pL} &= \hat{\theta} + \exp\left(\hat{x}_p - K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)}\right) \\ t_{pU} &= \hat{\theta} + \exp\left(\hat{x}_p + K_\gamma \sqrt{\mathrm{Var}(\hat{x}_p)}\right) \end{aligned}$$

where x_p refers to the percentile of the standard extreme value distribution.

For the Gompertz distributions, confidence limits are computed as

$$\begin{aligned} t_{pL} &= \hat{t}_p - K_\gamma \sqrt{\mathrm{Var}(\hat{t}_p)} \\ t_{pU} &= \hat{t}_p + K_\gamma \sqrt{\mathrm{Var}(\hat{t}_p)} \end{aligned}$$

where $\mathrm{Var}(\hat{t}_p)$ is calculated by the delta method. Because the quantile function has no closed form, the derivatives that are required for the three-parameter Gompertz distribution are obtained by the numerical method of finite differencing.

Reliability Function

For the extreme value, normal, and logistic distributions shown in Table 18.69, the maximum likelihood estimate of the reliability function $R(x) = \Pr\{X > x\}$ is given by

$$\hat{R}(x) = 1 - F\left(\frac{x - \hat{\mu}}{\hat{\sigma}}\right)$$

For the Gompertz distributions, the MLE of the reliability function is

$$\hat{R}(x) = \exp\left\{\hat{\beta} - \frac{\hat{\theta}x}{\hat{\alpha}} - \hat{\beta}\exp\left(\frac{x}{\hat{\alpha}}\right)\right\}$$

where $\hat{\theta} = 0$ for the standard, two-parameter Gompertz distribution.

The MLE of the CDF is $\hat{F}(x) = 1 - \hat{R}(x)$.

Confidence Intervals Let $\hat{u} = \frac{x-\hat{\mu}}{\hat{\sigma}}$. The approximate variance of u is

$$\mathrm{Var}(\hat{u}) \approx \frac{\mathrm{Var}(\hat{\mu}) + \hat{u}^2\mathrm{Var}(\hat{\sigma}) + 2\hat{u}\mathrm{Cov}(\hat{\mu}, \hat{\sigma})}{\hat{\sigma}^2}$$

Two-sided approximate $\gamma \times 100\%$ confidence intervals for $R(x)$ are computed as

$$R_L(x) = \hat{R}(u_2)$$

$$R_U(x) = \hat{R}(u_1)$$

where

$$u_1 = \hat{u} - K_\gamma\sqrt{\mathrm{Var}(\hat{u})}$$

$$u_2 = \hat{u} + K_\gamma\sqrt{\mathrm{Var}(\hat{u})}$$

and K_γ represents the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution. The corresponding limits for the CDF are

$$F_L(x) = 1 - R_U(x)$$

$$F_U(x) = 1 - R_L(x)$$

For the Gompertz distributions, confidence intervals for $R(x)$ are computed as

$$R_L(x) = \exp\{-\exp(u_2)\}$$

$$R_U(x) = \exp\{-\exp(u_1)\}$$

where $\hat{u} = \log[-\log\{\hat{R}_G(x)\}]$, $R_G(x)$ is the Gompertz reliability function, and

$$u_1 = \hat{u} - K_\gamma\sqrt{\mathrm{Var}(\hat{u})}$$

$$u_2 = \hat{u} + K_\gamma\sqrt{\mathrm{Var}(\hat{u})}$$

The variance term $\mathrm{Var}(\hat{u})$ is obtained by the delta method from the covariance matrix of the original parameter estimates.

As an alternative, you can request that two-sided $\gamma \times 100\%$ likelihood ratio confidence limits for the reliability function and the CDF be computed by specifying the LRCLSURV option in the ANALYZE statement or the LRCLSURV option in the PROBPLOT statement.

Limits for the Weibull, lognormal, and log-logistic reliability function $R(t)$ are the same as those for the corresponding extreme value, normal, or logistic reliability $R(y)$, where $y = \log(t)$. Limits for the three-parameter Weibull use $y = \log(t - \hat{\theta})$ and the extreme value CDF.

You can create a table containing estimates of the reliability function, the CDF, and confidence limits computed as described in this section with the SURVTIME= option in the ANALYZE statement or with the SURVTIME= option in the PROBPLOT statement. You can plot confidence limits for the CDF on probability plots created with the PROBPLOT statement with the PINTERVALS=CDF option in the PROBPLOT statement. PINTERVALS=CDF is the default option for parametric confidence limits on probability plots.

Estimation with the Binomial and Poisson Distributions

In addition to estimating the parameters of the distributions in Table 18.57, you can estimate parameters, compute confidence limits, compute predicted values and prediction limits, and compute chi-square tests for differences in groups for the binomial and Poisson distributions by using the ANALYZE statement. Specify either BINOMIAL or POISSON in the DISTRIBUTION statement to use one of these distributions. The ANALYZE statement options available for the binomial and Poisson distributions are given in Table 18.5. See the section "Analysis of Binomial Data" on page 1260 for an example of an analysis of binomial data.

Binomial Distribution

If r is the number of successes and n is the number of trials in a binomial experiment, then the maximum likelihood estimator of the probability p in the binomial distribution in Table 18.59 is computed as

$$\hat{p} = r/n$$

Two-sided $\gamma \times 100\%$ confidence limits for p are computed as in Johnson, Kotz, and Kemp (1992, p. 130):

$$p_L = \frac{\nu_1 F[(1-\gamma)/2; \nu_1, \nu_2]}{\nu_2 + \nu_1 F[(1-\gamma)/2; \nu_1, \nu_2]}$$

with $\nu_1 = 2r$ and $\nu_2 = 2(n - r + 1)$ and

$$p_U = \frac{\nu_1 F[(1+\gamma)/2; \nu_1, \nu_2]}{\nu_2 + \nu_1 F[(1+\gamma)/2; \nu_1, \nu_2]}$$

with $\nu_1 = 2(r + 1)$ and $\nu_2 = 2(n - r)$, where $F[\gamma; \nu_1, \nu_2]$ is the $\gamma \times 100\%$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator.

You can compute a sample size required to estimate p within a specified tolerance w with probability γ. Nelson (1982, p. 206) gives the following formula for the approximate sample size:

$$n \approx \hat{p}(1-\hat{p}) \left(\frac{K_\gamma}{w} \right)^2$$

where K_γ is the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution. The formula is based on the normal approximation for the distribution of $\hat{p}$. Nelson recommends using this formula if $np > 10$

and $np(1-p) > 10$. The value of γ used for computing confidence limits is used in the sample size computation. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. You specify a tolerance of *number* with the TOLERANCE(*number*) option.

The predicted number of successes X in a future sample of size m, based on the previous estimate of p, is computed as

$$\hat{X} = m(r/n) = m\hat{p}$$

Two-sided approximate $\gamma \times 100)\%$ prediction limits are computed as in Nelson (1982, p. 208). The prediction limits are the solutions X_L and X_U of

$$X_U/m = [(r+1)/n]F[(1+\gamma)/2; 2(r+1), 2X_U]$$

$$m/(X_L+1) = (n/r)F[(1+\gamma)/2; 2(X_L+1), 2r]$$

where $F[\gamma; \nu_1, \nu_2]$ is the $\gamma \times 100\%$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator. You request predicted values and prediction limits for a future sample of size *number* with the PREDICT(*number*) option.

You can test groups of binomial data for equality of their binomial probability by using the ANALYZE statement. You specify the K groups to be compared with a group variable having K levels.

Nelson (1982, p. 450) discusses a chi-square test statistic for comparing K binomial proportions for equality. Suppose there are r_i successes in n_i trials for $i = 1, 2, \ldots, K$. The grouped estimate of the binomial probability is

$$\hat{p} = \frac{r_1 + r_2 + \cdots + r_K}{n_1 + n_2 + \cdots + n_K}$$

The chi-square test statistic for testing the hypothesis $p_1 = p_2 = \ldots = p_K$ against $p_i \neq p_j$ for some i and j is

$$Q = \sum_{i=1}^{K} \frac{(r_i - n_i\hat{p})^2}{n_i\hat{p}(1-\hat{p})}$$

The statistic Q has an asymptotic chi-square distribution with $K-1$ degrees of freedom. The RELIABILITY procedure computes the contribution of each group to Q, the value of Q, and the p-value for Q based on the limiting chi-square distribution with $K-1$ degrees of freedom. If you specify the PREDICT option, predicted values and prediction limits are computed for each group, as well as for the pooled group. The p-value is defined as $p_0 = 1 - \chi^2_{K-1}[Q]$, where $\chi^2_{K-1}[x]$ is the chi-square CDF with $K-1$ degrees of freedom, and Q is the observed value. A test of the hypothesis of equal binomial probabilities among the groups with significance level α is

- $p_0 > \alpha$: do not reject the equality hypothesis
- $p_0 \leq \alpha$: reject the equality hypothesis

Poisson Distribution

You can use the ANALYZE statement to model data by using the Poisson distribution. The data consist of a count Y of occurrences in a "length" of observation T. Observation T is typically an *exposure time*, but it can have other units, such as distance. The ANALYZE statement enables you to compute the rate of occurrences, confidence limits, and prediction limits.

An estimate of the rate λ is computed as

$$\hat{\lambda} = Y/T$$

Two-sided $\gamma \times 100\%$ confidence limits for λ are computed as in Nelson (1982, p. 201):

$$\lambda_L = 0.5\chi^2[(1-\gamma)/2; 2Y]/T$$

$$\lambda_U = 0.5\chi^2[(1+\gamma)/2; 2(Y+1)]/T$$

where $\chi^2[\delta; \nu]$ is the $\delta \times 100\%$ percentile of the chi-square distribution with ν degrees of freedom.

You can compute a length T required to estimate λ within a specified tolerance w with probability γ. Nelson (1982, p. 202) provides the following approximate formula:

$$\hat{T} \approx \hat{\lambda}\left(\frac{K_\gamma}{w}\right)^2$$

where K_γ is the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution. The formula is based on the normal approximation for $\hat{\lambda}$ and is more accurate for larger values of λT. Nelson recommends using the formula when $\lambda T > 10$. The value of γ used for computing confidence limits is also used in the length computation. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. You specify a tolerance of *number* with the TOLERANCE(*number*) option.

The predicted future number of occurrences in a length S is

$$\hat{X} = (Y/T)S = \hat{\lambda}S$$

Two-sided approximate $\gamma \times 100\%$ prediction limits are computed as in Nelson (1982, p. 203). The prediction limits are the solutions X_L and X_U of

$$X_U/S = [(Y+1)/T]F[(1+\gamma)/2; 2(Y+1), 2X_U]$$

$$S/(X_L+1) = (T/Y)F[(1+\gamma)/2; 2(X_L+1), 2Y]$$

where $F[\gamma; \nu_1, \nu_2]$ is the $\gamma \times 100\%$ percentile of the F distribution with ν_1 degrees of freedom in the numerator and ν_2 degrees of freedom in the denominator. You request predicted values and prediction limits for a future exposure *number* with the PREDICT(*number*) option.

You can compute a chi-square test statistic for comparing K Poisson rates for equality. You specify the K groups to be compared with a group variable having K levels.

See Nelson (1982, p. 444) for more information. Suppose that there are Y_i Poisson counts in lengths T_i for $i = 1, 2, \ldots, K$ and that the Y_i are independent. The grouped estimate of the Poisson rate is

$$\hat{\lambda} = \frac{Y_1 + Y_2 + \cdots + Y_K}{T_1 + T_2 + \cdots + T_K}$$

The chi-square test statistic for testing the hypothesis $\lambda_1 = \lambda_2 = \ldots = \lambda_K$ against $\lambda_i \neq \lambda_j$ for some i and j is

$$Q = \sum_{i=1}^{K} \frac{(Y_i - \hat{\lambda} T_i)^2}{\hat{\lambda} T_i}$$

The statistic Q has an asymptotic chi-square distribution with $K - 1$ degrees of freedom. The RELIABILITY procedure computes the contribution of each group to Q, the value of Q, and the p-value for Q based on the limiting chi-square distribution with $K - 1$ degrees of freedom. If you specify the PREDICT option, predicted values and prediction limits are computed for each group, as well as for the pooled group. The p-value is defined as $p_0 = 1 - \chi^2_{K-1}[Q]$, where $\chi^2_{K-1}[x]$ is the chi-square CDF with $K - 1$ degrees of freedom and Q is the observed value. A test of the hypothesis of equal Poisson rates among the groups with significance level α is

- $p_0 > \alpha$: accept the equality hypothesis
- $p_0 \leq \alpha$: reject the equality hypothesis

Least Squares Fit to the Probability Plot

Fitting to the probability plot by least squares is an alternative to maximum likelihood estimation of the parameters of a life distribution. Only the failure times are used. A least squares fit is computed using points $(x_{(i)}, m_i)$, where $m_i = F^{-1}(a_i)$ and a_i are the plotting positions as defined in the section "Probability Plotting" on page 1343. The x_i are either the lifetimes for the normal, extreme value, or logistic distributions or the log lifetimes for the lognormal, Weibull, or log-logistic distributions. The ANALYZE, PROBPLOT, or RELATIONPLOT statement option FITTYPE=LSXY specifies the $x_{(i)}$ as the dependent variable ('y-coordinate') and the m_i as the independent variable ('x-coordinate'). You can optionally reverse the quantities used as dependent and independent variables by specifying the FITTYPE=LSYX option.

Weibayes Estimation

Weibayes estimation is a method of performing a Weibull analysis when there are few or no failures. The FITTYPE=WEIBAYES option requests this method. The method of Nelson (1985) is used to compute a one-sided confidence interval for the Weibull scale parameter when the Weibull shape parameter is specified. See Abernethy (2006) for more discussion and examples. The Weibull shape parameter β is assumed to be known and is specified to the procedure with the SHAPE=*number* option. Let $T_1, T_2, \ldots, T_n$ be the failure and censoring times, and let $r \geq 0$ be the number of failures in the data. If there are no failures ($r = 0$), a lower $\gamma \times 100\%$ confidence limit for the Weibull scale parameter α is computed as

$$\alpha_L = \{\sum_{i=1}^{n} T_i^{\beta} / [-\log(1 - \gamma)]\}^{1/\beta}$$

The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option.

If $r \geq 1$, the MLE of α is given by

$$\hat{\alpha} = [\sum_{i=1}^{n} T_i^{\beta}/r]^{1/\beta}$$

and a lower $\gamma \times 100\%$ confidence limit for the Weibull scale parameter α is computed as

$$\alpha_L = \hat{\alpha}[2r/\chi^2(\gamma, 2r+2)]^{1/\beta}$$

where $\chi^2(\gamma, 2r+2)$ is the γ percentile of a chi-square distribution with $2r+2$ degrees of freedom. The procedure uses the specified value of β and the computed value of α_L to compute distribution percentiles and the reliability function.

Estimation With Multiple Failure Modes

In many applications, units can experience multiple causes of failure, or *failure modes*. For example, in the section "Weibull Probability Plot for Two Combined Failure Modes" on page 1241, insulation specimens can experience either early failures due to manufacturing defects or degradation failures due to aging. The FMODE statement is used to analyze this type of data. See the section "FMODE Statement" on page 1283 for the syntax of the FMODE statement. This section describes the analysis of data when units experience multiple failure modes.

The assumptions used in the analysis are

- a cause, or mode, can be identified for each failure
- failure modes follow a series-system model; i.e., a unit fails when a failure due to one of the modes occurs
- each failure mode has the specified lifetime distribution with different parameters
- failure modes act statistically independently

Suppose there are m failure modes, with lifetime distribution functions $F_1(t), F_2(t), \ldots, F_m(t)$.

If you wish to estimate the lifetime distribution of a failure mode, say mode i, acting alone, specify the KEEP keyword in the FMODE statement. The failures from all other modes are treated as right-censored observations, and the lifetime distribution is estimated by one of the methods described in other sections, such as maximum likelihood. This lifetime distribution is interpreted as the distribution if the specified failure mode is acting alone, with all other modes eliminated. You can also specify more than one mode to KEEP, but the assumption is that all the specified modes have the same distribution.

If you specify the ELIMINATE keyword, failures due to the specified modes are treated as right censored. The resulting distribution estimate is the failure distribution if the specified modes are eliminated.

If you specify the COMBINE keyword, the failure distribution when all the modes specified in the FMODE statement modes act is estimated. The failure distribution $F_i(t), i = 1, 2, \ldots, m$, from each individual

mode is first estimated by treating all failures from other modes as right censored. The estimated failure distributions are then combined to get an estimate of the lifetime distribution when all modes act,

$$\hat{F}(t) = 1 - \prod_{i=1}^{m}[1 - \hat{F}_i(t)]$$

Pointwise approximate asymptotic normal confidence limits for $F(t)$ can be obtained by the delta method. See Meeker and Escobar (1998, appendix B.2). The delta method variance of $\hat{F}(t)$ is, assuming independence of failure modes,

$$\mathrm{Var}(\hat{F}(t)) = \sum_{i=1}^{m}[S_0(u_1)S_0(u_2)\ldots f_0(u_i)\ldots S_0(u_m)]^2\mathrm{Var}(u_i)$$

where $u_i = \frac{y-\hat{\mu}_i}{\hat{\sigma}_i}$, y is t for the extreme value, normal, and logistic distributions or $\log(t)$ for the Weibull, lognormal or log-logistic distributions, $\hat{\mu}_i$ and $\hat{\sigma}_i$ are location and scale parameter estimates for mode i, and S_0 and f_0 are the standard ($\mu = 0, \sigma = 1$) survival function and density function for the specified distribution.

Two-sided approximate $(1-\alpha)100\%$ pointwise confidence intervals are computed as in Meeker and Escobar (1998, section 3.6) as

$$[F_L, F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w}\right]$$

where

$$w = \exp\left[\frac{z_{1-\alpha/2}\mathrm{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))}\right]$$

where $\mathrm{se}_{\hat{F}} = \sqrt{\mathrm{Var}(\hat{F}(t))}$ and z_p is the pth quantile of the standard normal distribution.

Regression Model Statistics Computed for Each Observation for Lifetime Data

This section describes statistics that are computed for each observation when you fit a model for lifetime data. For regression models that are fit using the MODEL statement, you can specify a variety of statistics to be computed for each observation in the input data set. This section describes the method of computation for each statistic. See Table 18.32 and Table 18.33 for the syntax to request these statistics.

Predicted Values

The linear predictor is

$$\hat{\mu}_i = \mathbf{x}'\hat{\boldsymbol{\beta}}$$

where $\mathbf{x}_i$ is the vector of explanatory variables for the ith observation.

Percentiles

An estimator of the $p \times 100\%$ percentile x_p for the ith observation for the extreme value, normal, and logistic distributions is

$$\hat{x}_{i,p} = \mathbf{x}'\hat{\boldsymbol{\beta}} + z_p\hat{\sigma}$$

where $z_p = G^{-1}(p)$, G is the standardized CDF, and σ is the distribution scale parameter.

An estimator of the $p \times 100\%$ percentile t_p for the ith observation for the Weibull, lognormal, and log-logistic distributions is

$$\hat{t}_{i,p} = \exp[\mathbf{x}'\hat{\boldsymbol{\beta}} + z_p\hat{\sigma}]$$

where G is the standardized CDF of the extreme value, normal, or logistic distribution that corresponds to the logarithm of the lifetime, and σ is the distribution scale parameter.

The percentile of the lognormal (base 10) distribution is

$$\hat{t}_{i,p} = 10^{[\mathbf{x}'\hat{\boldsymbol{\beta}} + z_p\hat{\sigma}]}$$

where G is the CDF of the standard normal distribution.

An estimator of the $p \times 100\%$ percentile t_p for the ith observation for the generalized gamma distribution is

$$\hat{t}_{i,p} = \exp[\mathbf{x}'\hat{\boldsymbol{\beta}} + w_{\lambda,p}\hat{\sigma}]$$

where

$$w_{\lambda,p} = \frac{1}{\lambda}\log\left(\frac{\lambda^2}{2}\chi^2_{(2/\lambda^2),p}\right)$$

and $\chi^2_{k,p}$ is the $p \times 100\%$ percentile of the chi-square distribution with k degrees of freedom.

Standard Errors of Percentile Estimator

For the extreme value, normal, and logistic distributions, the standard error of the estimator of the $p \times 100\%$ percentile is computed as

$$\sigma_{i,p} = \sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_i \\ z_p \end{bmatrix}$$

and $\boldsymbol{\Sigma}$ is the covariance matrix of $(\hat{\boldsymbol{\beta}}, \hat{\sigma})$.

For the Weibull, lognormal, and log-logistic distributions, the standard error is computed as

$$\sigma_{i,p} = \exp(x_{i,p})\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

where $x_{i,p}$ is the percentile computed from the extreme value, normal, or logistic distribution that corresponds to the logarithm of the lifetime. The standard error for the lognormal (base 10) distribution is computed as

$$\sigma_{i,p} = 10^{x_{i,p}}\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

The standard error for the generalized gamma distribution percentile is computed as

$$\sigma_{i,p} = \exp[\mathbf{x}'\hat{\boldsymbol{\beta}} + w_{\lambda,p}\hat{\sigma}]\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_i \\ w_{\lambda,p} \\ \hat{\sigma}\frac{\partial w_{\lambda,p}}{\partial \lambda} \end{bmatrix}$$

$\boldsymbol{\Sigma}$ is the covariance matrix of $(\hat{\boldsymbol{\beta}}, \hat{\sigma}, \hat{\lambda})$, $\boldsymbol{\beta}$ is the vector of regression parameters, σ is the scale parameter, and λ is the shape parameter.

Confidence Limits for Percentiles

Two-sided approximate $100\gamma\%$ confidence limits for $x_{i,p}$ for the extreme value, normal, and logistic distributions are computed as

$$\begin{aligned} x_L &= \hat{x}_{i,p} - K_\gamma \sigma_{i,p} \\ x_U &= \hat{x}_{i,p} + K_\gamma \sigma_{i,p} \end{aligned}$$

where K_γ represents the $100(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution.

Limits for the Weibull, lognormal, and log-logistic percentiles are computed as

$$\begin{aligned} t_L &= \exp(x_L) \\ t_U &= \exp(x_U) \end{aligned}$$

where x_L and x_U are computed from the corresponding distributions for the logarithms of the lifetimes. For the lognormal (base 10) distribution,

$$\begin{aligned} t_L &= 10^{x_L} \\ t_U &= 10^{x_U} \end{aligned}$$

Limits for the generalized gamma distribution percentiles are computed as

$$t_L = \exp\left[\mathbf{x}_i'\boldsymbol{\beta} + w_{\lambda,p}\hat{\sigma} - K_\gamma\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}\right]$$
$$t_U = \exp\left[\mathbf{x}_i'\boldsymbol{\beta} + w_{\lambda,p}\hat{\sigma} + K_\gamma\sqrt{\mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}}\right]$$

Reliability Function

For the extreme value, normal, and logistic distributions, an estimate of the reliability function evaluated at the response y_i is computed as

$$R(y_i) = 1 - G\left(\frac{y_i - \mathbf{x}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)$$

where $G(x)$ is the standardized CDF of the distribution from Table 18.69.

Estimates of the reliability function evaluated at the response t_i for the Weibull, lognormal, log-logistic, and generalized gamma distributions are computed as

$$R(t_i) = 1 - G\left(\frac{\log(t_i) - \mathbf{x}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)$$

where $G(x)$ is the standardized CDF of the corresponding extreme value, normal, logistic, or generalized log-gamma distributions.

Residuals

The RELIABILITY procedure computes several different kinds of residuals. In the following equations, y_i represents the ith response value if the extreme value, normal, or logistic distributions are specified. If t_i is the ith response and if the Weibull, lognormal, log-logistic, or generalized gamma distributions are specified, then y_i represents the logarithm of the response $y_i = \log(t_i)$. If the lognormal (base 10) distribution is specified, then $y_i = \log_{10}(t_i)$.

Raw Residuals

The raw residual is computed as

$$r_{Ri} = y_i - \mathbf{x}'\hat{\boldsymbol{\beta}}$$

Standardized Residuals

The standardized residual is computed as

$$r_{Si} = \frac{y_i - \mathbf{x}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}$$

Adjusted Residuals

If an observation is right censored, then the standardized residual for that observation is also right censored. Adjusted residuals adjust censored standardized residuals upward by adding a percentile of the residual lifetime distribution, given that the standardized residual exceeds the censoring value. The default percentile is the median (50th percentile), but you can, optionally, specify a $\gamma \times 100\%$ percentile by using the RESIDALPHA=γ option in the MODEL statement. The $\gamma \times 100$ percentile residual life is computed as in Joe and Proschan (1984). The adjusted residual is computed as

$$r_{Ai} = \begin{cases} G^{-1}[1 - (1-\gamma)S(u_i)] & \text{for right-censored observations} \\ u_i & \text{for uncensored observations} \end{cases}$$

where G is the standard CDF,

$$S(u) = 1 - G(u)$$

is the reliability function, and

$$u_i = \frac{y_i - \mathbf{x}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}$$

If the generalized gamma distribution is specified, the standardized CDF and reliability functions include the estimated shape parameter $\hat{\lambda}$.

Modified Cox-Snell Residuals

Let

$$\delta_i = \begin{cases} 1 & \text{for uncensored observations} \\ 0 & \text{for right-censored observations} \end{cases}$$

The Cox-Snell residual is defined as

$$r_{Ci} = -\log(R(y_i))$$

where

$$R(y) = 1 - G\left(\frac{y - \mathbf{x}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)$$

is the reliability function. The modified Cox-Snell residual is computed as in Collett (1994, p. 152):

$$r'_{Ci} = r_{Ci} + (1 - \delta_i)\alpha$$

where α is an adjustment factor. If the fitted model is correct, the Cox-Snell residual has approximately a standard exponential distribution for uncensored observations. If an observation is censored, the residual evaluated at the censoring time is not as large as the residual evaluated at the (unknown) failure time. The adjustment factor α adjusts the censored residuals upward to account for the censoring. The default is $\alpha = 1.0$, the mean of the standard exponential distribution. You can, optionally, specify any adjustment factor by using the MODEL statement option RESIDADJ=α. Another commonly used value is the median of the standard exponential distribution, $\alpha = 0.693$.

Deviance Residuals

Deviance residuals are a zero-mean, symmetrized version of modified Cox-Snell residuals. Deviance residuals are computed as in Collett (1994, p. 153):

$$r_{Di} = \text{sgn}(\delta_i - r_{Ci})\{-2[\delta_i - r_{Ci} + \delta_i \log(r_{Ci})]\}^{1/2}$$

where

$$\text{sgn}(u) = \begin{cases} -1 & \text{if } u < 0 \\ 1 & \text{if } u \geq 0 \end{cases}$$

Regression Model Statistics Computed for Each Observation for Recurrent Events Data

This section describes statistics that are computed for each observation when you fit a model for recurrent events data. For regression models that are fit using the MODEL statement, you can specify a variety of statistics to be computed for each observation in the input data set. This section describes the method of computation for each statistic. See Table 18.32 and Table 18.34 for the syntax to request these statistics.

Let t_i be the event time in the ith observation in the input data set. The following statistics use the definitions of the mean function $\mathrm{M}(t;\eta,\beta)$ and intensity function $\lambda(t;\eta,\beta)$ in Table 18.72, where η and β are replaced by their maximum likelihood estimates. The shape parameter β is assumed to be constant for all observations. For regression models, the scale parameter η in Table 18.72 for the ith observation is

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \ldots$$

where $x_{i1}, x_{i2}, \ldots$ are regression coefficients and $\beta_0, \beta_1, \ldots$ are the maximum likelihood estimates of the regression parameters.

Predicted Values of Scale Parameter

The scale parameter that is predicted by the model for the ith observation is

$$\hat{\eta}_i = \mathbf{x}_i' \hat{\boldsymbol{\beta}}$$

where $\mathbf{x}_i$ is the vector of explanatory variables for the ith observation and $\boldsymbol{\beta}$ is the vector of maximum likelihood estimates of the regression parameters.

Mean Function

The predicted mean function is computed as $\mathrm{M}(t_i, \hat{\eta}_i, \hat{\beta})$.

Confidence Limits for the Mean Function

Confidence limits for the estimated $\mathrm{M}(t_i)$ are computed as described in the section "NHPP Model Parameter Confidence Limit Computation" on page 1384, using t_i, $\hat{\eta}_i$, and $\hat{\beta}$.

Standard Error of the Mean Function

The standard error of the estimated $\mathrm{M}(t_i)$ is computed as described in the section "NHPP Model Parameter Confidence Limit Computation" on page 1384, using t_i, $\hat{\eta}_i$, and $\hat{\beta}$.

Intensity Function

The predicted intensity function is computed as $\lambda(t_i, \hat{\eta}_i, \hat{\beta})$.

Confidence Limits for the Intensity Function

Confidence limits for the estimated $\lambda(t_i)$ are computed as described in the section "NHPP Model Parameter Confidence Limit Computation" on page 1384, using t_i, $\hat{\eta}_i$, and $\hat{\beta}$.

Standard Error of the Intensity Function

The standard error of the estimated $\lambda(t_i)$ is computed as described in the section "NHPP Model Parameter Confidence Limit Computation" on page 1384, using t_i, $\hat{\eta}_i$, and $\hat{\beta}$.

Recurrence Data from Repairable Systems

Failures in a system that can be repaired are sometimes modeled as *recurrence data*, or *recurrent events data*. When a repairable system fails, it is repaired and placed back in service. As a repairable system ages, it accumulates a history of repairs and costs of repairs. The mean cumulative function (MCF) $\mathrm{M}(t)$ is defined as the population mean of the cumulative number (or cost) of repairs up until time t. You can use the RELIABILITY procedure to compute and plot nonparametric estimates and plots of the MCF for the number of repairs or the cost of repairs. The Nelson (1995) confidence limits for the MCF are also computed and plotted. You can compute and plot estimates of the difference of two MCFs and confidence intervals. This is useful for comparing the repair performance of two systems.

See Nelson (2002, 1995, 1988), Doganaksoy and Nelson (1998), and Nelson and Doganaksoy (1989) for discussions and examples of nonparametric analysis of recurrence data.

You can also fit a parametric model for recurrent event data and display the resulting model on a plot, along with nonparametric estimates of the MCF.

See Rigdon and Basu (2000), Tobias and Trindade (1995), and Meeker and Escobar (1998) for discussions of parametric models for recurrent events data.

Nonparametric Analysis

Recurrent Events Data with Exact Ages

See the section "Analysis of Recurrence Data on Repairs" on page 1245 and the section "Comparison of Two Samples of Repair Data" on page 1250 for examples of the analysis of recurrence data with exact ages.

Formulas for the MCF estimator $\hat{\mathrm{M}}(t)$ and the variance of the estimator $\mathrm{Var}(\hat{\mathrm{M}}(t))$ are given in Nelson (1995). Table 18.70 shows a set of artificial repair data from Nelson (1988). For each system, the data consist of the system and cost for each repair. If you want to compute the MCF for the number of repairs, rather than cost of repairs, then you should set the cost equal to 1 for each repair. A plus sign (+) in place of a cost indicates that the age is a censoring time. The repair history of each system ends with a censoring time.

Table 18.70 System Repair Histories for Artificial Data

Unit	(Age in Months, Cost in $100)			
6	(5,$3)	(12,$1)	(12,+)	
5	(16,+)			
4	(2,$1)	(8,$1)	(16,$2)	(20,+)
3	(18,$3)	(29,+)		
2	(8,$2)	(14,$1)	(26,$1)	(33,+)
1	(19,$2)	(39,$2)	(42,+)	

Table 18.71 illustrates the calculation of the MCF estimate from the data in Table 18.70. The RELIABILITY procedure uses the following rules for computing the MCF estimates.

1. Order all events (repairs and censoring) by age from smallest to largest.
   - If the event ages of the same or different systems are equal, the corresponding data are sorted from the largest repair cost to the smallest. Censoring events always sort as smaller than repair events with equal ages.
   - When event ages and values of more than one system coincide, the corresponding data are sorted from the largest system identifier to the smallest. The system IDs can be numeric or character, but they are always sorted in ASCII order.
2. Compute the number of systems I in service at the current age as the number in service at the last repair time minus the number of censored units in the intervening times.
3. For each repair, compute the mean cost as the cost of the current repair divided by the number in service I.
4. Compute the MCF for each repair as the previous MCF plus the mean cost for the current repair.

Table 18.71 Calculation of MCF for Artificial Data

Event	(Age,Cost)	Number *I* in Service	Mean Cost	MCF
1	(2,$1)	6	$1/6=0.17	0.17
2	(5,$3)	6	$3/6=0.50	0.67
3	(8,$2)	6	$2/6=0.33	1.00
4	(8,$1)	6	$1/6=0.17	1.17
5	(12,$1)	6	$1/6=0.17	1.33
6	(12,+)	5		
7	(14,$1)	5	$1/5=0.20	1.53
8	(16,$2)	5	$2/5=0.40	1.93
9	(16,+)	4		
10	(18,$3)	4	$3/4=0.75	2.68
11	(19,$2)	4	$2/4=0.50	3.18
12	(20,+)	3		
13	(26,$1)	3	$1/3=0.33	3.52
14	(29,+)	2		
15	(33,+)	1		
16	(39,$2)	1	$2/1=2.00	5.52
17	(42,+)	0		

If you specify the VARIANCE=NELSON option, the variance of the estimator of the MCF $\mathrm{Var}(\hat{\mathrm{M}}(t))$ is computed as in Nelson (1995). If the VARIANCE=LAWLESS or VARMETHOD2 option is specified, the method of Lawless and Nadeau (1995) is used to compute the variance of the estimator of the MCF. This method is recommended if the number of systems or events is large or if a FREQ statement is used to specify

a frequency variable. If you do not specify a variance computation method, the method of Lawless and Nadeau (1995) is used.

Default approximate two-sided $\gamma \times 100\%$ pointwise confidence limits for $\mathrm{M}(t)$ are computed as

$$\mathrm{M}_L(t) = \hat{\mathrm{M}}(t) - K_\gamma \sqrt{\mathrm{Var}(\hat{\mathrm{M}}(t))}$$

$$\mathrm{M}_U(t) = \hat{\mathrm{M}}(t) + K_\gamma \sqrt{\mathrm{Var}(\hat{\mathrm{M}}(t))}$$

where K_γ represents the $100(1+\gamma)/2$ percentile of the standard normal distribution.

If you specify the LOGINTERVALS option in the MCFPLOT statement, alternative confidence intervals based on the asymptotic normality of $\log(\hat{\mathrm{M}}(t))$, rather than of $\hat{\mathrm{M}}(t)$, are computed. Let

$$w = \exp\left[\frac{K_\gamma \sqrt{\mathrm{Var}(\hat{\mathrm{M}}(t))}}{\hat{\mathrm{M}}(t)}\right]$$

Then the limits are computed as

$$\mathrm{M}_L(t) = \frac{\hat{\mathrm{M}}(t)}{w}$$

$$\mathrm{M}_U(t) = \hat{\mathrm{M}}(t) \times w$$

These alternative limits are always positive, and can provide better coverage than the default limits when the MCF is known to be positive, such as for counts or for positive costs. They are not appropriate for MCF differences, and are not computed in this case.

The following SAS statements create the tabular output shown in Figure 18.59 and the plot shown in Figure 18.60:

```
data Art;
   input Sysid $ Time Cost;
   datalines;
sys1 19 2
sys1 39 2
sys1 42 -1
sys2 8  2
sys2 14 1
sys2 26 1
sys2 33 -1
sys3 18 3
sys3 29 -1
```

```
sys4 16 2
sys4 2  1
sys4 20  -1
sys4 8  1
sys5 16 -1
sys6 5 3
sys6 12 1
sys6 12 -1
;

proc reliability data=Art;
   unitid Sysid;
   mcfplot Time*Cost(-1) ;
run;
```

The first table in Figure 18.59 displays the input data set, the number of observations used in the analysis, the number of systems (units), and the number of repair events. The second table displays the system age, MCF estimate, standard error, approximate confidence limits, and system ID for each event.

Figure 18.59 PROC RELIABILITY Output for the Artificial Data

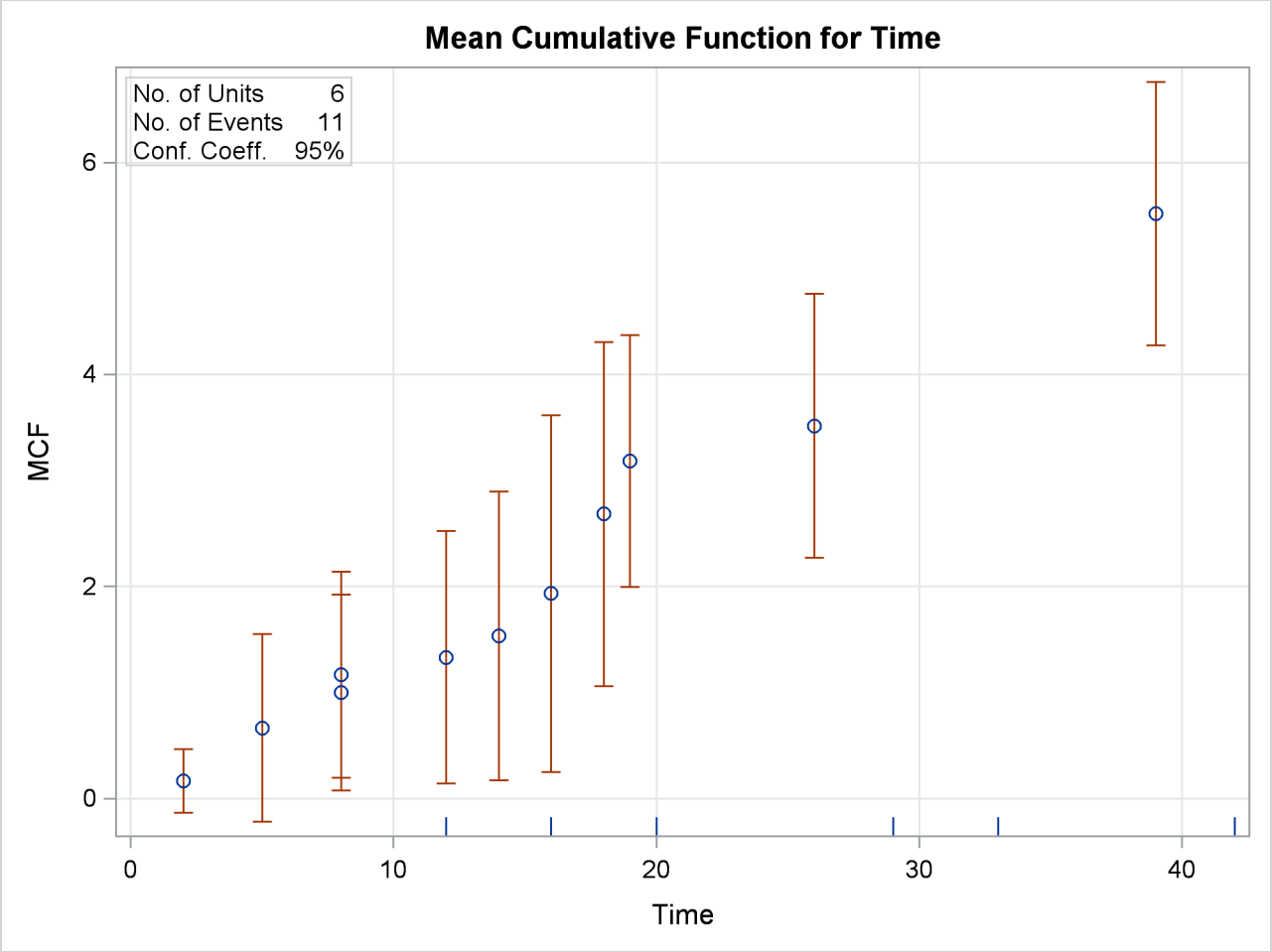
The RELIABILITY Procedure

Recurrence Data Summary	
Input Data Set	WORK.ART
Observations Used	17
Number of Units	6
Number of Events	11

Recurrence Data Analysis

			95% Confidence Limits		
Age	**Sample MCF**	**Standard Error**	**Lower**	**Upper**	**Unit ID**
2.00	0.167	0.152	-0.132	0.465	sys4
5.00	0.667	0.451	-0.218	1.551	sys6
8.00	1.000	0.471	0.076	1.924	sys2
8.00	1.167	0.495	0.196	2.138	sys4
12.00	1.333	0.609	0.141	2.526	sys6
14.00	1.533	0.695	0.172	2.895	sys2
16.00	1.933	0.859	0.249	3.618	sys4
18.00	2.683	0.828	1.061	4.306	sys3
19.00	3.183	0.607	1.993	4.373	sys1
26.00	3.517	0.634	2.274	4.759	sys2
39.00	5.517	0.634	4.274	6.759	sys1

Figure 18.60 MCF Plot for the Artificial Data



Recurrent Events Data with Ages Grouped into Intervals

Recurrence data are sometimes grouped into time intervals for convenience, or to reduce the number of data records to be stored and analyzed. Interval recurrence data consist of the number of recurrences and the number of censored units in each time interval.

You can use PROC RELIABILITY to compute and plot MCFs and MCF differences for interval data. Formulas for the MCF estimator $\hat{\mathrm{M}}(t)$ and the variance of the estimator $\mathrm{Var}(\hat{\mathrm{M}}(t))$ for interval data, as well as examples and interpretations, are given in Nelson (2002, chapter 5). These calculations apply only to the number of recurrences, and not to cost.

Let N_0 be the total number of units, R_i the number of recurrences in interval i, $i = 1, \ldots, n$, and C_i the number of units censored into interval i. Then $\mathrm{N}_0 = \sum_{i=1}^{n} \mathrm{C}_i$ and the number entering interval i is $\mathrm{N}_i = \mathrm{N}_{i-1} - \mathrm{C}_{i-1}$ with $\mathrm{C}_0 = 0$. The MCF estimate for interval i is $\mathrm{M}_0 = 0$,

$$\mathrm{M}_i = \mathrm{M}_{i-1} + \frac{\mathrm{R}_i}{\mathrm{N}_i - 0.5\mathrm{C}_i}$$

The denominator $\mathrm{N}_i - 0.5\mathrm{C}_i$ approximates the number at risk in interval i, and treats the censored units as if they were censored halfway through the interval. Since no censored units are likely to have ages lasting

through the entire last interval, the MCF estimate for the last interval is likely to be biased. A footnote is printed in the tabular output as a reminder of this bias for the last interval.

See the section "Analysis of Interval Age Recurrence Data" on page 1257 for an example of interval recurrence data analysis.

Comparison of Two Groups of Recurrent Events Data

If you specify a group variable in an MCFPLOT statement, and the group variable has only two levels representing two groups of data, then there are two ways to compare the MCFs of the two groups for equality.

If you specify the MCFDIFF option in the MCFPLOT statement, estimates of the difference between two MCFs $\mathrm{MDIFF}(t) = \mathrm{M}_1(t) - \mathrm{M}_2(t)$ and the variance of the estimator are computed and plotted as in Doganaksoy and Nelson (1998). Confidence limits for the MCF difference function are computed in the same way as for the MCF, by using the variance of the MCF difference function estimator. If the confidence limits do not enclose zero at any time point, then the statistical hypothesis that the two MCFs are equal at all times is rejected.

Cook and Lawless (2007, section 3.7.5) describe statistical tests based on weighted sums of sample differences in the MCFs of the two groups. These tests, similar to log-rank tests for survival data, are computed and displayed in the "Tests for Equality of Mean Functions" table. Two cases are computed, corresponding to different weight functions in the test statistic. The "constant" case is powerful in cases where the two MCFs are approximately proportional. The "linear" case is more powerful for cases where the MCFs are not proportional, but do not cross.

These methods are not available for grouped data, as described in "Recurrent Events Data with Ages Grouped into Intervals" on page 1380.

Parametric Models for Recurrent Events Data

The parametric models used for recurrent events data in PROC RELIABILITY are called *Poisson process models*. Some important features of these models are summarized below. See, for example, Rigdon and Basu (2000) and Meeker and Escobar (1998) for a full mathematical description of these models. See Cook and Lawless (2007) for a general discussion of maximum likelihood estimation in Poisson processes. Abernethy (2006) and US Army (2000) provide examples of the application of Poisson process models to system reliability.

Let $\mathrm{N}(t)$ be the number of events up to time t, and let $\mathrm{N}(a, b)$ be the number of events in the interval $(a, b]$. Then, for a Poisson process,

- $\mathrm{N}(0) = 0$.
- $\mathrm{N}(a, b)$ and $\mathrm{N}(c, d)$ are statistically independent if $a < b \leq c < d$.
- $\mathrm{N}(a, b)$ is a Poisson random variable with mean $\mathrm{M}(a, b) = \mathrm{M}(b) - \mathrm{M}(a)$ where $\mathrm{M}(t)$ is the mean number of failures up to time t. $\mathrm{M}(0) = 0$.

Poisson processes are characterized by their cumulative mean function $\mathrm{M}(t)$, or equivalently by their *intensity*, *rate*, or *rate of occurrence of failure* (ROCOF) function $\lambda(t) = \frac{d}{dt}\mathrm{M}(t)$, so that

$$\mathrm{M}(a, b) = \mathrm{M}(b) - \mathrm{M}(a) = \int_a^b \lambda(t)dt$$

Poisson processes are parameterized through their mean and rate functions. The RELIABILITY procedure provides the Poisson process models shown in Table 18.72.

Table 18.72 Models and Parameters for Recurrent Events Data

Model	Intensity Function	Mean Function
Crow-AMSAA	$\beta\eta t^{\beta-1}$	ηt^{β}
Homogeneous	$\exp(\eta)$	$\exp(\eta)t$
Log-linear	$\exp(\eta+\beta t)$	$\frac{\exp(\eta)}{\beta}[\exp(\beta t)-1]$
Power	$\frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}$	$\left(\frac{t}{\eta}\right)^{\beta}$
Proportional intensity	$\exp(\eta)\beta t^{\beta-1}$	$\exp(\eta)t^{\beta}$

For a homogeneous Poisson process, the intensity function is a constant; that is, the rate of failures does not change with time. For the other models, the rate function can change with time, so that a rate of failures that increases or decreases with time can be modeled. These models are called *non-homogeneous* Poisson processes.

In the RELIABILITY procedure, you specify a Poisson model with a DISTRIBUTION statement and a MODEL statement. You must also specify additional statements, depending on whether failure times are observed exactly, or observed to occur in time intervals. These statements are explained in the sections "Recurrent Events Data with Exact Event Ages" on page 1386, "Recurrent Events Data with Interval Event Ages" on page 1387, and "MODEL Statement" on page 1302. The DISTRIBUTION statement specifications for the models described in Table 18.72 are summarized in Table 18.73.

Table 18.73 DISTRIBUTION Statement Specification for Recurrent Events Data Models

Model	DISTRIBUTION Statement Value
Crow-AMSAA	NHPP(CA)
Homogeneous	HPP
Log-linear	NHPP(LOG)
Power	NHPP(POW)
Proportional intensity	NHPP(PROP)

For each of the models, you can specify a regression model for the parameter η in Table 18.72 for the ith observation as

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \ldots$$

where $x_{i1}, x_{i2}, \ldots$ are regression coefficients specified as described in the section "MODEL Statement" on page 1302. The parameter β_0 is labeled Intercept in the printed output, and the parameter β in Table 18.72 is labeled Shape. If no regression coefficients are specified, Intercept represents the parameter η in Table 18.72.

Model parameters are estimated by maximizing the log-likelihood function, which is equivalent to maximizing the likelihood function. The sections "Recurrent Events Data with Exact Event Ages" on page 1386 and "Recurrent Events Data with Interval Event Ages" on page 1387 contain descriptions of the form of the log likelihoods for the different models. An estimate of the covariance matrix of the maximum likelihood estimators (MLEs) of the parameters $\boldsymbol{\theta}$ is given by the inverse of the negative of the matrix of second derivatives of the log likelihood, evaluated at the final parameter estimates:

$$\boldsymbol{\Sigma} = [\sigma_{ij}] = -\mathbf{H}^{-1} = -\left[\frac{\partial^2 LL}{\partial\theta_i \partial\theta_j}\right]^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

The negative of the matrix of second derivatives is called the observed Fisher information matrix. The diagonal term σ_{ii} is an estimate of the variance of $\hat{\theta}_i$. Estimates of standard errors of the MLEs are provided by

$$\mathrm{SE}_{\theta_i} = \sqrt{\sigma_{ii}}$$

An estimator of the correlation matrix is

$$\mathbf{R} = \left[\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}\right]$$

Wald-type confidence intervals are computed for the model parameters as described in Table 18.74. Wald intervals use asymptotic normality of maximum likelihood estimates to compute approximate confidence intervals. If a parameter must be greater than zero, then an approximation based on the asymptotic normality of the logarithm of the parameter estimate is often more accurate, and the lower endpoint is strictly positive. The intercept term β_0 in an intercept-only power NHPP model with no other regression parameters represents η in Table 18.72, and is a model parameter that must be strictly positive. Also, the shape parameter for the power and proportional intensity models, represented by β in Table 18.72, must be strictly positive. In these cases, formula 7.11 of Meeker and Escobar (1998, p. 163) is used in Table 18.74 to compute confidence limits.

Table 18.74 shows the method of computation of approximate two-sided $\gamma \times 100\%$ confidence limits for model parameters. The default value of confidence is $\gamma = 0.95$. Other values of confidence are specified using the CONFIDENCE= option. In Table 18.74, K_γ represents the $(1+\gamma)/2 \times 100\%$ percentile of the standard normal distribution, and $w(\hat{\theta}) = \exp[K_\gamma(\mathrm{SE}_{\hat{\theta}})/\hat{\theta}]$.

Table 18.74 NHPP Model Parameter Confidence Limit Computation

Model		Parameters: Intercept	Parameters: Regression Parameters	Parameters: Shape
Crow-AMSAA				
Intercept-only model				
	Lower	$\hat{\beta}_0/w(\hat{\beta}_0)$		$\hat{\beta}/w(\hat{\beta})$
	Upper	$\hat{\beta}_0 w(\hat{\beta}_0)$		$\hat{\beta} w(\hat{\beta})$
Regression model				
	Lower	$\hat{\beta}_0 - K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta}/w(\hat{\beta})$
	Upper	$\hat{\beta}_0 + K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta} w(\hat{\beta})$
Homogeneous				
	Lower	$\hat{\beta}_0 - K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	
	Upper	$\hat{\beta}_0 + K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	
Log-linear				
	Lower	$\hat{\beta}_0 - K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta} - K_\gamma(\mathrm{SE}_{\hat{\beta}})$
	Upper	$\hat{\beta}_0 + K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta} + K_\gamma(\mathrm{SE}_{\hat{\beta}})$
Power				
Intercept-only model				
	Lower	$\hat{\beta}_0/w(\hat{\beta}_0)$		$\hat{\beta}/w(\hat{\beta})$
	Upper	$\hat{\beta}_0 w(\hat{\beta}_0)$		$\hat{\beta} w(\hat{\beta})$
Regression model				
	Lower	$\hat{\beta}_0 - K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta}/w(\hat{\beta})$
	Upper	$\hat{\beta}_0 + K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta} w(\hat{\beta})$
Proportional intensity				
	Lower	$\hat{\beta}_0 - K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i - K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta}/w(\hat{\beta})$
	Upper	$\hat{\beta}_0 + K_\gamma(\mathrm{SE}_{\hat{\beta}_0})$	$\hat{\beta}_i + K_\gamma(\mathrm{SE}_{\hat{\beta}_i})$	$\hat{\beta} w(\hat{\beta})$

You can request that profile likelihood confidence intervals for model parameters be computed instead of Wald intervals with the LRCL option in the MODEL statement.

Confidence limits for the mean and intensity functions for plots that are created with the MCFPLOT statement and for the table that is created with the OBSTATS option in the MODEL statement are computed by using the delta method. See Meeker and Escobar (1998, Appendix B) for a full explanation of this method. If $\mathbf{\Sigma}$ represents the covariance matrix of the estimates of the parameters η and β in Table 18.72, then the variance of the mean or intensity function estimate is given by

$$\mathrm{V} = [\frac{\partial g}{\partial \eta} \frac{\partial g}{\partial \beta}] \mathbf{\Sigma} [\frac{\partial g}{\partial \eta} \frac{\partial g}{\partial \beta}]'$$

where $g = g(t; \eta, \beta)$ represents either the mean function or the intensity function in Table 18.72. Since both of these functions must be positive, formula 7.11 of Meeker and Escobar (1998, p. 163) is used to compute

confidence limits, using the standard error $\sigma = \sqrt{\mathrm{V}}$. For regression models, the full covariance matrix of all the regression parameter estimates and the parameter β is used to compute $\mathbf{\Sigma}$.

Tests of Trend

For the nonhomogeneous models in Table 18.72, you can request a test for a homogeneous Poisson process by specifying the HPPTEST option in the MODEL statement. In this case the test is a likelihood ratio test for $\beta = 1$ for the power, Crow-AMSAA, and proportional intensity models, and $\beta = 0$ for the log-linear model.

You can request other tests of trend by using the TREND= option in the MODEL statement. These tests are not available for the kind of grouped data that are described in the section "Recurrent Events Data with Interval Event Ages" on page 1387. See Lindqvist and Doksum (2003), Kvaloy and Lindqvist (1998), and Meeker and Escobar (1998) for a discussion of these kinds of tests.

Let there be m independent systems observed, and let $t_{i1}, t_{i2}, \ldots t_{in_i}$ be the times of observed events for system i. Let the last time of observation of system i be T_i, with $t_{in_i} \leq T_i$. The following test statistics can be computed. These are extended versions of trend tests for a single system, and they allow valid tests for HPP versus NHPP even if the intensities vary from system to system.

- Military Handbook (MH)

$$\mathrm{MH} = 2\sum_{i=1}^{m}\sum_{j=1}^{n_i}\log\left(\frac{T_i}{t_{ij}}\right)$$

  The asymptotic distribution of MH is the chi-square with $2\sum_{i=1}^{m} n_i$ degrees of freedom. This test statistic is powerful for testing HPP versus NHPP in the power law model.

- Laplace (LA)

$$\mathrm{LA} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i}(t_{ij} - \frac{1}{2}T_i)}{\sqrt{\frac{1}{12}\sum_{i=1}^{m} n_i T_i^2}}$$

  The asymptotic distribution of LA is the standard normal. This test statistic is powerful for testing HPP versus NHPP in a log-linear model.

- Lewis-Robinson (LR)

  Let LA_i be the Laplace test statistic for system i,

$$\mathrm{LA}_i = \frac{\sum_{j=1}^{n_i}(t_{ij} - \frac{1}{2}T_i)}{\sqrt{\frac{1}{12}n_i}}$$

  The extended Lewis-Robinson test statistic is defined as

$$\mathrm{LR} = \sum_{i=1}^{m}\frac{\bar{X}_i}{\sigma_i}\mathrm{LA}_i$$

  where $\bar{X}_i$ and σ_i are the estimated mean and standard deviation of the event interarrival times for system i. The asymptotic distribution of LR is the standard normal. This test statistic is powerful for testing HPP versus NHPP in a log-linear model.

INEST Data Set for Recurrent Events Models

You can specify a SAS data set to set lower bounds, upper bounds, equality constraints, or initial values for estimating the intercept and shape parameters of the models in Table 18.72 by using the INEST= option in the MODEL statement. The data set must contain a variable named _TYPE_ that specifies the action that you want to take in the iterative estimation process, and some combination of variables named _INTERCEPT_ and _SHAPE_ that represent the distribution parameters. If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group.

The possible values of _TYPE_ and corresponding actions are summarized in Table 18.75.

Table 18.75 _TYPE_ Variable Values

Value of _TYPE_	Action
LB	Lower bound
UB	Upper bound
EQ	Equality
PARMS	Initial value

For example, you can use the INEST data set In created by the following SAS statements to specify an equality constraint for the shape parameter. The data set In specifies that the shape parameter be constrained to be 1.5. Since the variable _Intercept_ is set to missing, no action is taken for it, and this variable could be omitted from the data set.

```
data In ;
   input _Type_$ 1-5  _Intercept_  _Shape_;
   datalines;
eq    . 1.5
;
```

Recurrent Events Data with Exact Event Ages

Let there be m independent systems observed, and let $t_{i1}, t_{i2}, \ldots t_{in_i}$ be the times of observed events for system i. Let the last time of observation of system i be T_i, with $t_{in_i} \leq T_i$.

If there are no regression parameters in the model, or there are regression parameters and they are constant for each system, then the log-likelihood function is

$$LL = \sum_{i=1}^{m} \{ \sum_{j=1}^{n_i} [\log(\lambda(t_{ij})] - \mathrm{M}(T_i) \}$$

If there are regression parameters that can change over time for individual systems, the RELIABILITY procedure uses the convention that a covariate value specified at a given event time takes effect immediately after the event time; that is, the value of a covariate used at an event time is the value specified at the previous event time. The value used at the first event time is the value specified at that event time. You can establish a different value for the first event time by specifying a zero cost event previous to the first actual event. The zero cost event is not used in the analysis, but it is used to establish a covariate value for the next event time. The covariate value used at the end time T_i is the value established at the last event time.

With these conventions, the log likelihood is

$$LL = \sum_{i=1}^{m}\{\sum_{j=1}^{n_i}[\log(\lambda(t_{ij}) - (\mathrm{M}(t_{ij}) - \mathrm{M}(t_{i,j-1})] - (\mathrm{M}(T_i) - \mathrm{M}(t_{in_i})\}$$

with $t_{i0} = 0$ for each $i = 1, 2, \ldots, m$. Note that this log likelihood reduces to the previous log likelihood if covariate values do not change over time for each system.

In order to specify a parametric model for recurrence data with exact event times, you specify the event times, end of observation times, and regression model, if any, with a MODEL statement, as described in the section "MODEL Statement" on page 1302. In addition, you specify a variable that uniquely identifies each system with a UNITID statement. See the section "Parametric Model for Recurrent Events Data" on page 1266 for an example of fitting a parametric recurrent events model to data with exact recurrence times.

Recurrent Events Data with Interval Event Ages

If n independent and statistically identical systems are observed in the time interval $(t_a, t_b]$, then the number r of events that occur in the interval is a Poisson random variable with mean $n\mathrm{M}(t_a, t_b) = n[\mathrm{M}(t_a) - \mathrm{M}(t_b)]$, where $\mathrm{M}(t)$ is the cumulative mean function for an individual system.

Let $(t_0, t_1], (t_1, t_2], \ldots, (t_{m-1}, t_m]$ be nonoverlapping time intervals for which r_i events are observed among the n_i systems observed in time interval $(t_{i-1}, t_i]$. The parameters in the mean function $\mathrm{M}(t)$ are estimated by maximizing the log likelihood

$$LL = \sum_{i=1}^{m}[r_i \log(n_i) + r_i \log(\mathrm{M}(t_{i-1}, t_i)) - n_i \mathrm{M}(t_{i-1}, t_i) - \log(r_i!)]$$

The time intervals do not have to be of the same length, and they do not have to be adjacent, although the preceding formula shows them as adjacent.

If you have data from groups of systems to which you are fitting a regression model (for example, to model the effects of different manufacturing lines or different vendors), the time intervals in the different groups do not have to coincide. The only requirement is that the data in the different groups be independent; for example, you cannot have data from the same systems in two different groups.

In order to specify a parametric model for interval recurrence data, you specify the time intervals and regression model, if any, with a MODEL statement, as described in the section "MODEL Statement" on page 1302. In addition, you specify a variable that contains the number n_i of systems under observation in time interval i with an NENTER statement, and the number of events r_i observed with a FREQ statement. See the section "Parametric Model for Interval Recurrent Events Data" on page 1268 for an example of fitting a parametric recurrent events model to data with interval recurrence times.

Duane Plots

A Duane plot is defined as a graph of the quantity $\mathrm{H}(t) = \mathrm{M}(t)/t$ versus t, where $\mathrm{M}(t)$ is the MCF. The graph axes are usually both on the log scale, so that if $\mathrm{M}(t)$ is the power law type in Table 18.72, a linear graph is produced. Duane plots are traditionally used as a visual assessment of the goodness of fit of a power law model. You should exercise caution in using Duane plots, because even if the underlying model is a power law process, a nonlinear Duane plot can result. See Rigdon and Basu (2000, section 4.1.1) for a discussion of Duane plots. You can create a Duane plot by specifying the DUANE option in the MCFPLOT statement. A scatter plot of nonparametric estimates of $\hat{\mathrm{H}}(t_i) = \hat{\mathrm{M}}(t_i)/t_i$ versus t_i is created on a log-log scale, where

$\hat{\mathrm{M}}(t_i)$ are the nonparametric estimates of the MCF that are described in the section "Nonparametric Analysis" on page 1376. If you specify a parametric model with the FIT=MODEL option in the MCFPLOT statement, the corresponding parametric estimate of $\mathrm{H}(t)$ is plotted on the same graph as the scatter plot.

ODS Table Names

The following tables contain the ODS table names created by the RELIABILITY Procedure, organized by the statements that produce them.

Table 18.76 Tables Produced with the ANALYZE Statement

Table Name	Description
ConvergenceStatus	Convergence status
CorrMat	Parameter correlation matrix
CovMat	Parameter covariance matrix
DatSum	Summary of fit
GradHess	Last evaluation of parameters, gradient, and Hessian
IterEM	Iteration history for Turnbull algorithm
IterLRParm	Iteration history for likelihood ratio confidence intervals for parameters
IterLRPer	Iteration history for likelihood ratio confidence intervals for percentiles
IterParms	Iteration history for parameter estimates
Lagrange	Lagrange multiplier statistics
NObs	Observations summary
PBEst	Poisson/binomial estimates by group
PBPred	Poisson/binomial predicted values
PBPredTol	Poisson/binomial predicted values by group
PBSum	Poisson/binomial analysis summary
PBTol	Poisson/binomial tolerance estimates
PctEst	Percentile estimates
ParmEst	Parameter estimates
ParmOther	Fitted distribution mean, median, mode
PGradHess	Last evaluation of parameters, gradient, and Hessian in terms of stable parameters
ProbabilityEstimates	Nonparametric cumulative distribution function estimates
RelInfo	Model information
SurvEst	Survival function estimates
TurnbullGrad	Interval probabilities, reduced gradient, Lagrange multipliers for Turnbull algorithm
WCorrMat	Parameter correlation matrix for Weibull distribution
WCovMat	Parameter covariance matrix for Weibull distribution

Table 18.77 Tables Produced with the MCFPLOT Statement

Table Name	Description
McfDEst	MCF difference estimates

Table 18.77 Tables Produced with the MCFPLOT Statement (continued)

Table Name	Description
McfDSum	MCF difference data summary
McfEst	MCF estimates
McfLogRank	Tests of difference between two MCFs
McfSum	MCF data summary

Table 18.78 Tables Produced with the MODEL Statement

Table Name	Description
MConvergenceStatus	Convergence status
ClassLevels	Class level information
ModCorMat	Parameter correlation matrix
ModCovMat	Parameter covariance matrix
ModFitSum	Summary of fit
ModInfo	Model information
ModIterLRparm	Iteration history for likelihood ratio confidence intervals for parameters
ModIterParms	Iteration history for parameter estimates
ModLagr	Lagrange multiplier statistics
ModLastGradHess	Last evaluation of the gradient and Hessian
ModNObs	Observations summary
ModObstats	Observation statistics
ModParmInfo	Parameter information
ModPrmEst	Parameter estimates
RecurGoodFit	Test for homogeneous Poisson process

Table 18.79 Tables Produced with PROBPLOT and RELATIONPLOT Statements

Table Name	Description
ConvergenceStatus	Convergence status
CorrMat	Parameter correlation matrix
CovMat	Parameter covariance matrix
DatSum	Summary of fit
GradHess	Last evaluation of parameters, gradient, and Hessian
IterEM	Iteration history for Turnbull algorithm
IterLRParm	Iteration history for likelihood ratio confidence intervals for parameters
IterLRPer	Iteration history for likelihood ratio confidence intervals for percentiles
IterParms	Iteration history for parameter estimates
Lagrange	Lagrange multiplier statistics
NObs	Observations summary
PctEst	Percentile estimates

Table 18.79 Tables Produced with PROBPLOT and RELATIONPLOT Statements (continued)

Table Name	Description
ParmEst	Parameter estimates
ParmOther	Fitted distribution mean, median, mode
PGradHess	Last evaluation of parameters, gradient, and Hessian in terms of stable parameters
ProbabilityEstimates	Nonparametric cumulative distribution function estimates
RelInfo	Model information
SurvEst	Survival function estimates
TurnbullGrad	Interval probabilities, reduced gradient, Lagrange multipliers for Turnbull algorithm
WCorrMat	Parameter correlation matrix for Weibull distribution
WCovMat	Parameter covariance matrix for Weibull distribution

ODS Graphics

SAS/QC procedures use ODS Graphics functionality to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

Before you create graphs, ODS Graphics must be enabled. For example:

```
ods graphics on;

proc reliability;
   probplot y;
run;

ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

See Chapter 4, "SAS/QC Graphics," for alternative methods of creating graphics with PROC RELIABILITY. See the section "Analysis of Right-Censored Data from a Single Population" on page 1206 for an example that uses ODS Graphics in PROC RELIABILITY to create a probability plot and the section "ODS Graph Names" on page 1390 for ODS Graphics table names.

ODS Graph Names

If ODS Graphics is enabled (for example, with the ODS GRAPHICS ON statement), PROC RELIABILITY creates graphs by using ODS Graphics. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC RELIABILITY generates are listed in Table 18.80, along with the required statements and options.

Table 18.80 Graphs Produced by PROC RELIABILITY

ODS Graph Name	Description	Statement	Option
IntensityPlot	Plot of intensity function for NHPP model	MCFPLOT	Fit=MODEL
IntensityPlots	Plot of intensity functions for NHPP model for multiple groups	MCFPLOT	Fit=MODEL
MCFDiffPlot	Plot of mean cumulative function differences	MCFPLOT	MCFDIFF
MCFPlot	Plot of mean cumulative function plot for single population	MCFPLOT	Default
MCFPlotPanel	Plot of mean cumulative function plots for multiple groups	MCFPLOT	Group variable
ProbabilityPlot	Probability plot for single population	PROBPLOT	Default
ProbabilityPlotFM	Probability plot with failure modes	PROBPLOT	FMODE
ProbabilityPlotPanel	Probability plots for multiple groups	PROBPLOT	Group variable
PercentilePlot	Plot of model percentiles	RELATIONPLOT	Default
RecurrentEventsPlot	Plot of recurrent event times	MCFPLOT	EVENTPLOT
RecurrentEventsPlotPanel	Plots of recurrent events for multiple groups	MCFPLOT	EVENTPLOT
RelationPlot	Plot of model percentiles with probability plot	RELATIONPLOT	PPLOT

References

Abernethy, R. B. (2006). *The New Weibull Handbook*. 5th ed. North Palm Beach, FL: Robert B. Abernethy.

Ascher, H., and Feingold, H. (1984). *Repairable Systems Reliability*. New York: Marcel Dekker.

Byar, D. P. (1980). *The Veterans Administration Study of Chemoprophylaxis for Recurrent Stage 1 Bladder Tumors: Comparisons of Placebo, Pyridoxine, and Topical Thiotepa.* Edited by M. Pavone-Macaluso, P. H. Smith, and F. Edsmyr. New York: Plenum.

Collett, D. (1994). *Modelling Survival Data in Medical Research*. London: Chapman & Hall.

Cook, R. J., and Lawless, J. F. (2007). *The Statistical Analysis of Recurrent Events*. New York: Springer.

Crowder, M. J., Kimber, A. C., Smith, R. L., and Sweeting, T. J. (1991). *Statistical Analysis of Reliability Data*. New York: Chapman & Hall.

Doganaksoy, N., Hahn, G. J., and Meeker, W. Q. (2002). "Reliability Analysis by Failure Mode." *Quality Progress* 35:47–52.

Doganaksoy, N., and Nelson, W. (1998). "A Method to Compare Two Samples of Recurrence Data." *Lifetime Data Analysis* 4:51–63.

Doganaksoy, N., and Schmee, J. (1993). "Orthogonal Parameters with Censored Data." *Communications in Statistics—Theory and Methods* 22:669–685.

Gentleman, R., and Geyer, C. J. (1994). "Maximum Likelihood for Interval Censored Data: Consistency and Computation." *Biometrika* 81:618–623.

Joe, H., and Proschan, F. (1984). "Percentile Residual Life Functions." *Operations Research* 32:668–678.

Kvaloy, J. T., and Lindqvist, B. H. (1998). "TTT-Based Tests for Trend in Repairable Systems Data." *Reliability Engineering and Systems Safety* 60:13–28.

Lawless, J. F. (2003). *Statistical Model and Methods for Lifetime Data*. 2nd ed. New York: John Wiley & Sons.

Lawless, J. F., and Nadeau, C. (1995). "Some Simple Robust Methods for the Analysis of Recurrent Events." *Technometrics* 37:158–168.

Lindqvist, B. H., and Doksum, K. A., eds. (2003). *Mathematical and Statistical Methods in Reliability.* Vol. 7 of Series on Quality, Reliability, and Engineering Statistics. Singapore: World Scientific.

Meeker, W. Q., and Escobar, L. A. (1998). *Statistical Methods for Reliability Data*. New York: John Wiley & Sons.

Nair, V. N. (1984). "Confidence Bands for Survival Functions with Censored Data: A Comparative Study." *Technometrics* 26:265–275.

Nelson, W. (1982). *Applied Life Data Analysis*. New York: John Wiley & Sons.

Nelson, W. (1985). "Weibull Analysis of Reliability Data with Few or No Failures." *Journal of Quality Technology* 17:140–146.

Nelson, W. (1988). "Graphical Analysis of System Repair Data." *Journal of Quality Technology* 20:24–35.

Nelson, W. (1990). *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*. New York: John Wiley & Sons.

Nelson, W. (1995). "Confidence Limits for Recurrence Data—Applied to Cost or Number of Product Repairs." *Technometrics* 37:147–157.

Nelson, W. (2002). *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*. Philadelphia: SIAM.

Nelson, W., and Doganaksoy, N. (1989). *A Computer Program for an Estimate and Confidence Limits for the Mean Cumulative Function for Cost or Number of Repairs of Repairable Products*. Technical Report 89CRD239, GE Research & Development Center, Schenectady, NY.

Rigdon, S. E., and Basu, A. P. (2000). *Statistical Methods for the Reliability of Repairable Systems*. New York: John Wiley & Sons.

Tobias, P. A., and Trindade, D. C. (1995). *Applied Reliability*. 2nd ed. New York: Van Nostrand Reinhold.

Turnbull, B. W. (1976). "The Empirical Distribution Function with Arbitrarily Grouped, Censored, and Truncated Data." *Journal of the Royal Statistical Society, Series B* 38:290–295.

US Army (2000). *AMSAA Reliability Growth Guide*. Technical Report TR-652, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD.

Chapter 19
The SHEWHART Procedure

Contents

Introduction: SHEWHART Procedure **1403**
Uses of Shewhart Charts 1404
Characteristics of Shewhart Charts 1405
Classification of Shewhart Charts 1406
Learning to Use the SHEWHART Procedure 1408
PROC SHEWHART and General Statements **1409**
Overview: SHEWHART Procedure 1409
Syntax: SHEWHART Procedure 1410
BY Statement 1410
ID Statement 1411
Graphical Enhancement Statements 1411
PROC SHEWHART Statement 1411
Summary of Options 1412
Dictionary of Options 1412
Input and Output Data Sets: SHEWHART Procedure 1416
BOXCHART Statement: SHEWHART Procedure **1417**
Overview: BOXCHART Statement 1417
Getting Started: BOXCHART Statement 1418
Creating Box Charts from Raw Data 1418
Creating Box Charts from Subgroup Summary Data 1423
Saving Summary Statistics 1426
Saving Control Limits 1428
Reading Preestablished Control Limits 1431
Syntax: BOXCHART Statement 1432
Summary of Options 1434
Details: BOXCHART Statement 1445
Constructing Box Charts 1445
Output Data Sets 1448
Input Data Sets 1453
Methods for Estimating the Standard Deviation 1459
Percentile Definitions 1460
Examples: BOXCHART Statement 1461
Example 19.1: Using Box Charts to Compare Subgroups 1461
Example 19.2: Creating Various Styles of Box-and-Whisker Plots 1464
Example 19.3: Creating Notched Box-and-Whisker Plots 1469
Example 19.4: Creating Box-and-Whisker Plots with Varying Widths 1470

Example 19.5: Creating Box-and-Whisker Plots with Different Line Styles and Colors 1472
Example 19.6: Computing the Control Limits for Subgroup Maximums 1474
Example 19.7: Constructing Multi-Vari Charts 1477
CCHART Statement: SHEWHART Procedure **1482**
Overview: CCHART Statement 1482
Getting Started: CCHART Statement 1483
Creating c Charts from Defect Count Data 1483
Saving Control Limits 1484
Reading Preestablished Control Limits 1487
Creating c Charts from Nonconformities per Unit 1488
Saving Nonconformities per Unit 1491
Syntax: CCHART Statement 1492
Summary of Options 1494
Details: CCHART Statement 1503
Constructing Charts for Numbers of Nonconformities (c Charts) 1503
Output Data Sets 1506
Input Data Sets 1508
Examples: CCHART Statement 1511
Example 19.8: Applying Tests for Special Causes 1511
Example 19.9: Specifying a Known Expected Number of Nonconformities 1514
Example 19.10: Creating c Charts for Varying Numbers of Units 1516
IRCHART Statement: SHEWHART Procedure **1518**
Overview: IRCHART Statement 1518
Getting Started: IRCHART Statement 1519
Creating Individual Measurements and Moving Range Charts 1519
Saving Individual Measurements and Moving Ranges 1521
Reading Individual Measurements and Moving Ranges 1522
Saving Control Limits 1523
Reading Preestablished Control Limits 1526
Specifying the Computation of the Moving Range 1528
Syntax: IRCHART Statement 1529
Summary of Options 1530
Details: IRCHART Statement 1542
Constructing Charts for Individual Measurements and Moving Ranges 1542
Output Data Sets 1544
Input Data Sets 1547
Methods for Estimating the Standard Deviation 1550
Interpreting Charts for Individual Measurements and Moving Ranges 1550
Examples: IRCHART Statement 1551
Example 19.11: Applying Tests for Special Causes 1551
Example 19.12: Specifying Standard Values for the Process Mean and Standard Deviation 1554
Example 19.13: Displaying Distributional Plots in the Margin 1556
MCHART Statement: SHEWHART Procedure **1559**
Overview: MCHART Statement 1559

Getting Started: MCHART Statement 1560
Creating Charts for Medians from Raw Data 1560
Creating Charts for Medians from Subgroup Summary Data 1563
Saving Summary Statistics 1566
Saving Control Limits 1569
Reading Preestablished Control Limits 1571
Syntax: MCHART Statement 1573
Summary of Options 1574
Details: MCHART Statement 1585
Constructing Median Charts 1585
Output Data Sets 1587
Input Data Sets 1591
Methods for Estimating the Standard Deviation 1594
Examples: MCHART Statement 1595
Example 19.14: Controlling Value of Central Line 1595
Example 19.15: Estimating the Process Standard Deviation 1600
MRCHART Statement: SHEWHART Procedure **1603**
Overview: MRCHART Statement 1603
Getting Started: MRCHART Statement 1604
Creating Charts for Medians and Ranges from Raw Data 1604
Creating Charts for Medians and Ranges from Summary Data 1606
Saving Summary Statistics 1609
Saving Control Limits 1610
Reading Preestablished Control Limits 1613
Syntax: MRCHART Statement 1615
Summary of Options 1617
Details: MRCHART Statement 1628
Constructing Charts for Medians and Ranges 1628
Output Data Sets 1631
Input Data Sets 1634
Methods for Estimating the Standard Deviation 1637
Examples: MRCHART Statement 1638
Example 19.16: Working with Unequal Subgroup Sample Sizes 1638
Example 19.17: Specifying Axis Labels 1643
NPCHART Statement: SHEWHART Procedure **1646**
Overview: NPCHART Statement 1646
Getting Started: NPCHART Statement 1647
Creating np Charts from Count Data 1647
Creating np Charts from Summary Data 1649
Saving Proportions of Nonconforming Items 1651
Saving Control Limits 1652
Reading Preestablished Control Limits 1655
Syntax: NPCHART Statement 1656
Summary of Options 1658

Details: NPCHART Statement 1667
Constructing Charts for Number Nonconforming (np Charts) 1667
Output Data Sets 1669
Input Data Sets 1672
Examples: NPCHART Statement 1676
Example 19.18: Applying Tests for Special Causes 1676
Example 19.19: Specifying Standard Average Proportion 1678
Example 19.20: Working with Unequal Subgroup Sample Sizes 1680
Example 19.21: Specifying Control Limit Information 1684
PCHART Statement: SHEWHART Procedure **1686**
Overview: PCHART Statement 1686
Getting Started: PCHART Statement 1687
Creating p Charts from Count Data 1687
Creating p Charts from Summary Data 1690
Saving Proportions of Nonconforming Items 1693
Saving Control Limits 1693
Reading Preestablished Control Limits 1696
Syntax: PCHART Statement 1697
Summary of Options 1699
Details: PCHART Statement 1708
Constructing Charts for Proportion Nonconforming (p Charts) 1708
Output Data Sets 1711
Input Data Sets 1713
Examples: PCHART Statement 1716
Example 19.22: Applying Tests for Special Causes 1716
Example 19.23: Specifying Standard Average Proportion 1719
Example 19.24: Working with Unequal Subgroup Sample Sizes 1721
Example 19.25: Creating a Chart with Revised Control Limits 1724
Example 19.26: OC Curve for Chart 1727
RCHART Statement: SHEWHART Procedure **1729**
Overview: RCHART Statement 1729
Getting Started: RCHART Statement 1730
Creating Range Charts from Raw Data 1731
Creating Range Charts from Summary Data 1733
Saving Summary Statistics 1736
Saving Control Limits 1737
Reading Preestablished Control Limits 1740
Syntax: RCHART Statement 1742
Summary of Options 1743
Details: RCHART Statement 1752
Constructing Range Charts 1753
Output Data Sets 1754
Input Data Sets 1757
Methods for Estimating the Standard Deviation 1760

Examples: RCHART Statement 1761
Example 19.27: Computing Probability Limits 1761
Example 19.28: Specifying Control Limit Information 1763
SCHART Statement: SHEWHART Procedure **1767**
Overview: SCHART Statement 1767
Getting Started: SCHART Statement 1768
Creating Standard Deviation Charts from Raw Data 1768
Creating Standard Deviation Charts from Subgroup Summary Data 1771
Saving Summary Statistics 1773
Saving Control Limits 1774
Reading Preestablished Control Limits 1776
Syntax: SCHART Statement 1778
Summary of Options 1779
Details: SCHART Statement 1789
Constructing Charts for Standard Deviations 1789
Output Data Sets 1790
Input Data Sets 1794
Methods for Estimating the Standard Deviation 1797
Examples: SCHART Statement 1798
Example 19.29: Specifying a Known Standard Deviation 1798
Example 19.30: Computing Average Run Lengths for s Charts 1800
UCHART Statement: SHEWHART Procedure **1801**
Overview: UCHART Statement 1801
Getting Started: UCHART Statement 1802
Creating u Charts from Defect Count Data 1802
Saving Control Limits 1805
Reading Preestablished Control Limits 1807
Creating u Charts from Nonconformities per Unit 1808
Saving Nonconformities per Unit 1811
Syntax: UCHART Statement 1812
Summary of Options 1814
Details: UCHART Statement 1823
Constructing Charts for Nonconformities per Unit (u Charts) 1823
Output Data Sets 1825
Input Data Sets 1828
Examples: UCHART Statement 1831
Example 19.31: Applying Tests for Special Causes 1831
Example 19.32: Specifying a Known Expected Number of Nonconformities 1833
Example 19.33: Creating u Charts for Varying Numbers of Units 1835
XCHART Statement: SHEWHART Procedure **1838**
Overview: XCHART Statement 1838
Getting Started: XCHART Statement 1839
Creating Charts for Means from Raw Data 1839
Creating Charts for Means from Subgroup Summary Data 1842

Saving Summary Statistics . . . 1845
Saving Control Limits . . . 1846
Reading Preestablished Control Limits . . . 1849
Syntax: XCHART Statement . . . 1851
Summary of Options . . . 1852
Details: XCHART Statement . . . 1863
Constructing Charts for Means . . . 1863
Output Data Sets . . . 1864
Input Data Sets . . . 1868
Methods for Estimating the Standard Deviation . . . 1871
Examples: XCHART Statement . . . 1873
Example 19.34: Applying Tests for Special Causes . . . 1873
Example 19.35: Estimating the Process Standard Deviation . . . 1875
Example 19.36: Plotting OC Curves for Mean Charts . . . 1879
Example 19.37: Computing Process Capability Indices . . . 1880
XRCHART Statement: SHEWHART Procedure . . . **1881**
Overview: XRCHART Statement . . . 1881
Getting Started: XRCHART Statement . . . 1882
Creating Charts for Means and Ranges from Raw Data . . . 1882
Creating Charts for Means and Ranges from Summary Data . . . 1885
Saving Summary Statistics . . . 1888
Saving Control Limits . . . 1889
Reading Preestablished Control Limits . . . 1892
Syntax: XRCHART Statement . . . 1894
Summary of Options . . . 1896
Details: XRCHART Statement . . . 1907
Constructing Charts for Means and Ranges . . . 1907
Output Data Sets . . . 1909
Input Data Sets . . . 1912
Methods for Estimating the Standard Deviation . . . 1915
Examples: XRCHART Statement . . . 1916
Example 19.38: Applying Tests for Special Causes . . . 1916
Example 19.39: Specifying Standard Values for the Process Mean and Standard Deviation 1919
Example 19.40: Working with Unequal Subgroup Sample Sizes . . . 1921
XSCHART Statement: SHEWHART Procedure . . . **1925**
Overview: XSCHART Statement . . . 1925
Getting Started: XSCHART Statement . . . 1926
Creating Charts for Means and Standard Deviations from Raw Data . . . 1926
Creating Charts for Means and Standard Deviations from Summary Data . . . 1929
Saving Summary Statistics . . . 1932
Saving Control Limits . . . 1933
Reading Preestablished Control Limits . . . 1935
Syntax: XSCHART Statement . . . 1936
Summary of Options . . . 1938

Details: XSCHART Statement 1949
Constructing Charts for Means and Standard Deviations 1949
Output Data Sets 1951
Input Data Sets 1954
Methods for Estimating the Standard Deviation 1957
Examples: XSCHART Statement 1959
Example 19.41: Specifying Probability Limits 1959
Example 19.42: Computing Subgroup Summary Statistics 1961
Example 19.43: Analyzing Nonnormal Process Data 1962
Chart Statement Details: SHEWHART Procedure **1966**
ODS Tables 1966
ODS Graphics 1968
ODS Graphics Template 1968
Subgroup Variables 1970
Numeric Subgroup Variables 1971
Character Subgroup Variables 1971
Capability Indices 1971
The Index C_p 1972
The Index CPL 1972
The Index CPU 1972
The Index C_{pk} 1972
The Index C_{pm} 1973
Axis Labels 1973
Missing Values 1975
INSET and INSET2 Statements: SHEWHART Procedure **1975**
Overview: INSET and INSET2 Statements 1975
Getting Started: INSET and INSET2 Statements 1976
Displaying Summary Statistics on a Control Chart 1976
Formatting Values and Customizing Labels 1978
Adding a Header and Positioning the Inset 1980
Syntax: INSET and INSET2 Statements 1981
Summary of INSET Keywords 1983
Summary of Options 1985
Dictionary of Options 1986
Details: INSET and INSET2 Statements 1988
Positioning the Inset Using Compass Points 1988
Positioning the Inset in the Margins 1989
Positioning the Inset Using Coordinates 1990
Dictionary of Options: SHEWHART Procedure **1993**
General Options 1994
Options for ODS Graphics 2051
Options for Traditional Graphics 2056
Options for Legacy Line Printer Charts 2070
Graphical Enhancements: SHEWHART Procedure **2071**

Overview: Graphical Enhancements . . . 2071
Displaying Stratified Process Data . . . 2071
Displaying Stratification in Levels of a Classification Variable . . . 2073
Displaying Stratification in Blocks of Observations . . . 2074
Displaying Stratification in Phases . . . 2079
Displaying Multiple Sets of Control Limits . . . 2081
Displaying Auxiliary Data with Stars . . . 2090
Creating a Basic Star Chart . . . 2092
Adding Reference Circles to Stars . . . 2093
Specifying the Style of Stars . . . 2095
Specifying the Method of Standardization . . . 2098
Displaying Trends in Process Data . . . 2100
Step 1: Preliminary Mean and Standard Deviation Charts . . . 2102
Step 2: Modeling the Trend . . . 2103
Step 3: Displaying the Trend Chart . . . 2104
Clipping Extreme Points . . . 2105
Labeling Axes . . . 2109
Default Labels . . . 2109
Labeling the Horizontal Axis . . . 2110
Labeling the Vertical Axis . . . 2111
Selecting Subgroups for Computation and Display . . . 2113
Using WHERE Statements . . . 2114
Using Switch Variables . . . 2117
Tests for Special Causes: SHEWHART Procedure . . . 2119
Standard Tests for Special Causes . . . 2119
Requesting Standard Tests . . . 2122
Interpreting Standard Tests for Special Causes . . . 2124
Modifying Standard Tests for Special Causes . . . 2124
Applying Tests with Varying Subgroup Sample Sizes . . . 2125
Labeling Signaled Points with a Variable . . . 2128
Applying Tests with Multiple Phases . . . 2129
Applying Tests with Multiple Sets of Control Limits . . . 2131
Enhancing the Display of Signaled Tests . . . 2134
Nonstandard Tests for Special Causes . . . 2134
Applying Tests to Range and Standard Deviation Charts . . . 2134
Applying Tests Based on Generalized Patterns . . . 2136
Customizing Tests with DATA Step Programs . . . 2141
Specialized Control Charts: SHEWHART Procedure . . . 2143
Overview: Specialized Control Charts . . . 2143
Autocorrelation in Process Data . . . 2144
Diagnosing and Modeling Autocorrelation . . . 2145
Strategies for Handling Autocorrelation . . . 2148
Multiple Components of Variation . . . 2152
Preliminary Examination of Variation . . . 2153

Determining the Components of Variation 2156
Short Run Process Control 2160
Analyzing the Difference from Nominal 2161
Testing for Constant Variances 2168
Standardizing Differences from Nominal 2170
Nonnormal Process Data 2171
Creating a Preliminary Individual Measurements Chart 2173
Calculating Probability Limits 2174
Multivariate Control Charts 2177
Calculating the Chart Statistic 2177
Examining the Principal Component Contributions 2181
Interactive Control Charts: SHEWHART Procedure **2183**
Overview: Interactive Control Charts 2183
Details: Interactive Control Charts 2183
Saving Graphics Coordinates in a Control Chart 2183
Associating URLs with Subgroups in HTML 2186
Links and Tests for Special Causes 2187
References **2188**

Introduction: SHEWHART Procedure

The Shewhart control chart is a graphical and analytical tool for deciding whether a process is in a state of statistical control. You can use the SHEWHART procedure to display many different types of control charts, including all commonly used charts for variables and attributes. In addition, you can use the SHEWHART procedure to

- create charts from either raw data (actual measurements) or summarized data
- analyze multiple process variables
- specify control limits in terms of a multiple of the standard error of the plotted summary statistic or as probability limits
- adjust control limits to compensate for unequal subgroup sizes
- estimate control limits from the data, compute control limits from specified values for population parameters (known standards), or read limits from an input data set
- create historical control charts that display distinct sets of control limits for multiple time phases
- perform tests for special causes based on runs patterns (Western Electric rules)
- estimate the process standard deviation using various methods (variable charts only)
- accept numeric-valued or character-valued subgroup variables

- display subgroups with date and time formats
- save chart statistics and control limits in output data sets
- tabulate chart statistics and control limits
- produce charts as traditional graphics, ODS Graphics output, or legacy line printer charts. Line printer charts can use special formatting characters that improve the appearance of the chart. Traditional graphics can be annotated, saved, and replayed.

Uses of Shewhart Charts

The Shewhart chart is named after Walter A. Shewhart (1891-1967), a physicist at the Bell Telephone Laboratories, who introduced the method in 1924 and elaborated upon it in his book *Economic Control of Quality of Manufactured Product*, (1931). The concepts underlying the control chart are that the natural variability in any manufacturing process can be quantified with a set of control limits and that the variation exceeding these limits signals a change in the process.

In industry, the Shewhart chart is the most commonly applied statistical quality control method for studying the variation in output from a manufacturing process. Shewhart charts are typically used to distinguish variation due to *special causes* from variation due to *common causes*. Special causes, also referred to as *assignable causes*, are local, sporadic problems such as the failure of a particular machine or a mistakenly recorded measurement. Common causes are problems inherent in the manufacturing system as a whole. Examples of common causes are inadequate product design, inherited defective material, and excessive humidity.

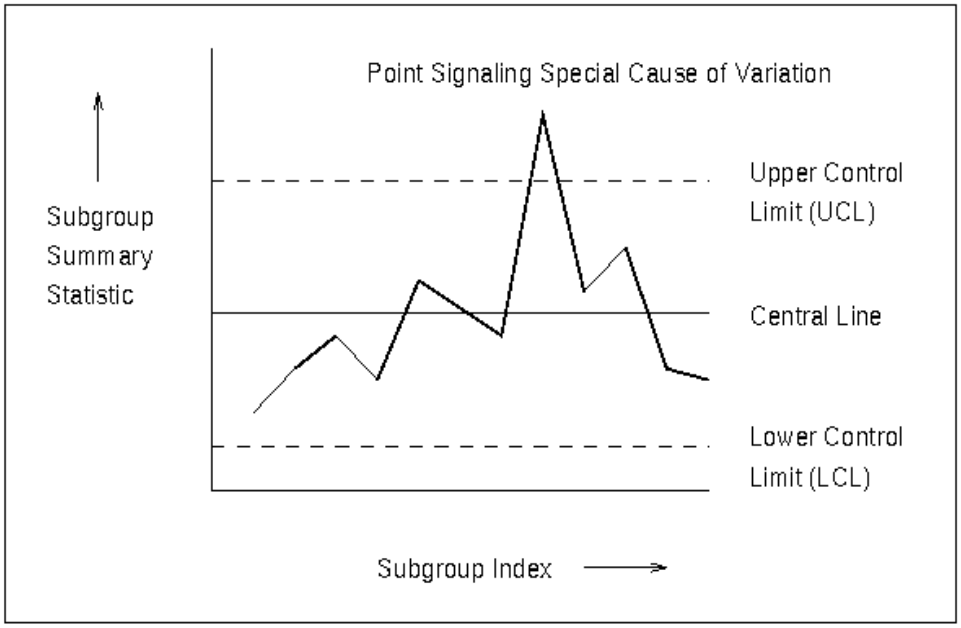
When the special causes have been identified and eliminated, the process is said to be in *statistical control*. Once statistical control has been established, Shewhart charts can be used to monitor the process for the occurrence of future special causes and to measure and reduce the effects of common causes.

Deming (1982) emphasized that the improvement of a process can begin only after statistical control has been established. Deming also noted that control chart techniques are applicable to quality improvement in service industries as well as manufacturing industries.

Characteristics of Shewhart Charts

Figure 19.1 illustrates a typical Shewhart chart.

Figure 19.1 A Shewhart Control Chart



All Shewhart charts have the following characteristics:

- Each point represents a *summary statistic* computed from a sample of measurements of a quality characteristic. For example, the summary statistic might be the average value of a critical dimension of five items selected at random, or it might be the proportion of nonconforming items in a sample of 100 items.
- The *vertical axis* of a Shewhart chart is scaled in the same units as the summary statistic.
- The samples from which the summary statistics are computed are referred to as *rational subgroups* or *subgroup samples*. The organization of the data into subgroups is critical to the interpretation of a Shewhart chart. Shewhart (1931) advocated selecting rational subgroups so that variation within subgroups is minimized and variation among subgroups is maximized; this makes the chart more sensitive to shifts in the process level. Various approaches to subgrouping are discussed by Grant and Leavenworth (1988), Montgomery (1996), and Kume (1985).
- The *horizontal axis* of a Shewhart chart identifies the subgroup samples. Frequently, the samples are indexed according to the order in which they were taken or the time at which they were taken. Subgroup samples can also be assigned labels that indicate some other type of classification (for example, lot number).
- The *central line* on a Shewhart chart indicates the average (expected value) of the summary statistic when the process is in statistical control.

- The *upper and lower control limits*, labeled UCL and LCL, respectively, indicate the range of variation to be expected in the summary statistic when the process is in statistical control. The control limits are commonly computed as 3σ limits[1] representing three standard errors[2] of variation in the summary statistic above and below the central line. However, the limits can also be determined using a multiple of the standard error other than three, or from a specified probability (α) that a single summary statistic will exceed the limits when the process is in statistical control. Limits determined by the latter method are referred to as *probability limits*.

  The control limits are also determined by the subgroup sample size because the standard error of the summary statistic is a function of sample size. If the sample size is constant across subgroups, the control limits are typically horizontal lines, as in Figure 19.1. However, if the sample size varies from subgroup to subgroup, the limits are usually adjusted to compensate for the effect of sample size, resulting in step-like boundaries.

  Control limits can be estimated from the data being analyzed, or they can be standard, previously determined values. Estimated limits are often used when statistical control is being established, and standard limits are often used when statistical control is being maintained.

- *A point outside the control limits* signals the presence of a special cause of variation. Additionally, *tests for special causes* (also referred to as *Western Electric rules* and *runs tests*) can signal an out-of-control condition if a statistically unusual pattern of points is observed in the control chart. For example, one pattern used to diagnose the existence of a trend is seven consecutive steadily increasing points.

  When the process is in statistical control, a point can fall outside the control limits purely by chance, resulting in a false out-of-control signal. However, when the Shewhart chart correctly signals the presence of a special cause, additional action is needed to determine the nature of the problem and eliminate it.

Classification of Shewhart Charts

Shewhart charts are broadly classified according to the type of data analyzed.

- Shewhart charts for *variables* are used when the quality characteristic of a process is measured on a continuous scale.

- Shewhart charts for *attributes* are used when the quality characteristic of a process is measured by counting the number of nonconformities (defects) in an item or the number of nonconforming (defective) items in a sample.

[1] In this context, the symbol σ always stands for the standard error of the subgroup summary statistic that is plotted on the chart. Elsewhere in this section, σ is also used to denote the standard deviation of a process, also referred to as the population standard deviation. This dual usage is standard practice.

[2] The term *standard deviation* is also used by some authors to refer to this quantity; see, for example, Montgomery (1996). This section uses the term *standard error* for the dispersion of the distribution of a statistic and the term *standard deviation* for the dispersion of a distribution of individual measurements.

Shewhart charts for variables are further classified according to the subgroup summary statistic plotted on the chart.

- $\bar{X}$ and R charts display subgroup means (averages) and ranges. Typically the two charts are presented on the same page, with the $\bar{X}$ chart aligned above the R chart to facilitate the simultaneous analysis of the central tendency and variability of the process.
- $\bar{X}$ and s charts display subgroup means (averages) and standard deviations. Typically the two charts are presented on the same page, with the $\bar{X}$ chart aligned above the s chart.
- Median and range charts display subgroup medians and ranges. Typically the two charts are presented on the same page, with the median chart aligned above the R chart.
- Charts for individual measurements and moving ranges display individual measurements and moving ranges of two of more successive measurements. In this case the subgroup sample consists of a single observation.

Likewise, Shewhart charts for attributes are classified according to the subgroup summary statistic plotted on the chart:

- A p chart displays the proportion of nonconforming (defective) items in a subgroup sample.
- An np chart displays the number of nonconforming (defective) items in a subgroup sample.
- A u chart displays the number of nonconformities (defects) per unit in a subgroup sample consisting of an arbitrary number of units.
- A c chart displays the number of nonconformities (defects) in a unit (here, a subgroup sample typically consists of one unit).

You can create all of the preceding types of Shewhart charts with the SHEWHART procedure. In addition, you can create a wide variety of nonstandard Shewhart charts, including

- a trend chart displaying a time trend plot and an $\bar{X}$ chart (or median chart) that has been created removing the time trend from the data. The trend chart and $\bar{X}$ chart are presented on the same page, with the $\bar{X}$ aligned above the trend chart, to facilitate the detection of special causes after accounting for the time trend effect. Trend charts are applicable when a time trend (for instance, due to tool wear) is observed in a preliminary $\bar{X}$ chart of the original data.
- a box chart displaying a box plot (box-and-whisker plot) for each subgroup and control limits for the subgroup means. This chart facilitates detailed analysis of the subgroup distributions and is applicable with large subgroup sample sizes (ten or more).

Learning to Use the SHEWHART Procedure

Although the SHEWHART procedure provides a large number of options, you can use the procedure to create a basic Shewhart chart with as few as two SAS statements:

- the PROC SHEWHART statement, which starts the procedure and specifies the input SAS data set
- a chart statement, which specifies the type of Shewhart chart you want to create and the variables in the input data set that you want to analyze

For example, you can use the following statements to create $\bar{X}$ and R charts with 3σ limits for measurements read from a SAS data set named Drums:

```
proc shewhart data=Drums;
   xrchart Flangewidth * Hour;
run;
```

The keyword XRCHART in the chart statement specifies that $\bar{X}$ and R charts are to be created. The following SAS variables are specified in the XRCHART statement:

- A SAS variable (Flangewidth), whose values are the process measurements, is specified before the asterisk. This variable is referred to as the *process*.
- A SAS variable (Hour), whose values classify the measurements into subgroups, is specified after the asterisk. This variable is referred to as a *subgroup-variable*.

The same form of specification is used with other chart statements to create different types of Shewhart charts. The following table lists the 13 chart statements that are available with the SHEWHART procedure:

Table 19.1 Chart Statements in the SHEWHART Procedure

Statement	Chart(s) Displayed	"Getting Started" Section
BOXCHART	Box chart with optional trend chart	"Getting Started: BOXCHART Statement" on page 1418
CCHART	*c* chart	"Getting Started: CCHART Statement" on page 1483
IRCHART	Individual and moving range charts	"Getting Started: IRCHART Statement" on page 1519
MCHART	Median chart with optional trend chart	"Getting Started: MCHART Statement" on page 1560
MRCHART	Median and *R* charts	"Getting Started: MRCHART Statement" on page 1604
NPCHART	*np* chart	"Getting Started: NPCHART Statement" on page 1647
PCHART	*p* chart	"Getting Started: PCHART Statement" on page 1687
RCHART	*R* chart	"Getting Started: RCHART Statement" on page 1730
SCHART	*s* chart	"Getting Started: SCHART Statement" on page 1768
UCHART	*u* chart	"Getting Started: UCHART Statement" on page 1802
XCHART	$\bar{X}$ chart with optional trend chart	"Getting Started: XCHART Statement" on page 1839
XRCHART	$\bar{X}$ and *R* charts	"Getting Started: XRCHART Statement" on page 1882
XSCHART	$\bar{X}$ and *s* charts	"Getting Started: XSCHART Statement" on page 1926

If you are using the SHEWHART procedure for the first time, you should do the following:

- Read "PROC SHEWHART and General Statements" on page 1409.
- Read the "Getting Started" subsection in the section for the chart statement you need to create your chart. Table 19.1 provides links to these sections.

Once you have learned to use a particular chart statement, you will find it straightforward to use the remaining chart statements because their syntax is nearly the same. A separate, self-contained section is provided for each chart statement.

PROC SHEWHART and General Statements

Overview: SHEWHART Procedure

The PROC SHEWHART statement starts the SHEWHART procedure and it optionally identifies various data sets.

To create a Shewhart chart, you specify a chart statement (after the PROC SHEWHART statement) that specifies the type of Shewhart chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request $\bar{X}$ and R charts:

```
proc shewhart data=Values;
   xrchart Weight*Lot;
run;
```

Here, the DATA= option specifies an input data set (Values) with the *process* measurement variable (Weight) and the *subgroup-variable* (Lot).

You can use options in the PROC SHEWHART statement to

- specify input data sets containing variables to be analyzed, control limit information, or annotation information
- specify a graphics catalog for saving traditional graphics
- specify whether charts are to be produced as traditional graphics or line printer charts
- define characters used for features on line printer charts

See Chapter 4, "SAS/QC Graphics," for a detailed discussion of the alternatives available for producing charts with SAS/QC procedures.

NOTE: If you are learning to use the SHEWHART procedure, you should read both this section and the "Getting Started" subsection in the section for the chart statement that corresponds to the chart you want to create.

Syntax: SHEWHART Procedure

The following are the primary statements that control the SHEWHART procedure:

PROC SHEWHART < *options* > **;**
BOXCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
CCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
IRCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
MCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
MRCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
NPCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
PCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
RCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
SCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
UCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
XCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
XRCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
XSCHART (*processes*) ∗ *subgroup-variable* < (*block-variables*) >
< **=***symbol-variable* | **=***'character'* > < */ options* > **;**
INSET *keyword-list* < */ options* > **;**
INSET2 *keyword-list* < */ options* > **;**

The PROC SHEWHART statement invokes the procedure and specifies the input data set. The chart statements create different types of control charts. You can specify one or more of each of the chart statements. For details, read the section on the chart statement that corresponds to the type of control chart you want to produce.

In addition, you can optionally specify one of each of the following statements:

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SHEWHART to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the SHEWHART procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The ID statement specifies variables used to identify observations. The ID variables must be variables in the DATA= or HISTORY= input data sets.

The ID variables are used in the following ways:

- If you create an OUTHISTORY= or OUTTABLE= data set, the ID variables are included. If the input data set is a DATA= data set, only the values of the ID variables from the first observation in each subgroup are passed to the output data set.
- If you specify the TABLEID or TABLEALL option in a chart statement, the table produced is augmented by a column for each of the ID variables. Only the values of the ID variables from the first observation in each subgroup are tabulated. See the entry for the TABLEID option in "Dictionary of Options: SHEWHART Procedure" on page 1993.
- If you specify the BOXSTYLE=SCHEMATICID option or the BOXSTYLE= SCHEMATICIDFAR option in the BOXCHART statement, the value of the first variable listed in the ID statement is used to label each extreme observation. See Output 19.2.3 and Output 19.2.4.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance graphical and printed output. If you are creating traditional graphics, you can also use AXIS, LEGEND, and SYMBOL statements to enhance your charts. For details, refer to *SAS/GRAPH: Reference* and see the section for the control chart statement that you are using.

PROC SHEWHART Statement

The syntax for the PROC SHEWHART statement is as follows:

PROC SHEWHART < *options* > **;**

The PROC SHEWHART statement starts the SHEWHART procedure, and it optionally identifies various data sets and requests graphics output. The following section lists all *options*. See "Dictionary of Options" on page 1412 below for detailed information.

Summary of Options

The following table lists the PROC SHEWHART *options* by function:

Table 19.2 PROC SHEWHART Statement Options

Option	Description
Input Data Sets Options	
ANNOTATE=	specifies input data set containing annotation information for primary chart
ANNOTATE2=	specifies input data set containing annotation information for secondary chart
BOX=	specifies input data set containing summary statistics, control limits, and box chart outlier values
DATA=	specifies input data set containing raw data
HISTORY=	specifies input data set containing summary statistics
LIMITS=	specifies input data set containing control limits
TABLE=	specifies input data set containing summary statistics and control limits
TESTHTML=	specifies input data set defining links to be associated with subgroups with positive tests for special causes
TESTURLS=	specifies input data set containing URLs associated with subgroups with positive tests for special causes
Plotting and Graphics Options	
FORMCHAR(*index*)=	defines characters used for features on charts
GOUT=	specifies catalog for saving traditional graphics output
LINEPRINTER	requests line printer charts be produced

Dictionary of Options

The following entries provide detailed descriptions of options in the PROC SHEWHART statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set containing Annotate variables as described in *SAS/GRAPH: Reference*. You can use this data set to add features to traditional graphics. Use this data set only when creating traditional graphics. This option is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option. Features provided in this data set are displayed on every chart produced in the current run of PROC SHEWHART.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an input data set that contains annotate variables. You can use this data set to add features to the secondary chart in statements that produce two charts (the IRCHART, MRCHART, XRCHART, and XSCHART statements and, when you specify the TRENDVAR= option, the BOXCHART, MCHART, and XCHART statements). The restrictions and features are the same as those for the ANNOTATE= option.

BOX=*SAS-data-set*

names an input data set that contains subgroup summary statistics, control limits, and outlier values in "strung out" form, with more than one observation per subgroup. Each observation corresponds to one feature of one subgroup's box-and-whisker plot. Typically, this data set is created as an OUTBOX= data set in a previous run of PROC SHEWHART with a BOXCHART statement. The BOX= data set is the only kind of summary data set you can use to produce schematic box-and-whisker plots. The BOXCHART statement is the only chart statement you can use with a BOX= input data set.

You cannot use a BOX= data set together with a DATA=, HISTORY=, or TABLE= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set.

DATA=*SAS-data-set*

names an input data set that contains raw data as observations. Note that the DATA= data set might need sorting. If the values of the *subgroup-variable* are numeric, you must sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted.

The DATA= data set can contain more than one observation for each value of the *subgroup-variable*. This happens, for example, when you produce a control chart for means and ranges with the XRCHART statement.

You cannot use a DATA= data set together with a BOX=, HISTORY=, or TABLE= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "DATA= Data Set" subsection in the section for the chart statement you are using.

FORMCHAR(*index*)='*string*'

defines characters used for features on legacy line printer charts, where *index* is a list of numbers ranging from 1 to 17, and *string* is a character or hexadecimal string. The *index* identifies which features are controlled with the *string* characters, as listed in Table 19.3. If you specify the FORMCHAR= option and omit the *index*, the *string* controls all 17 features.

Table 19.3 FORMCHAR= Features

Value of *index*	Description of Character	Chart Feature
1	Vertical bar	Frame
2	Horizontal bar	Frame, central line
3	Box character (upper left)	Frame
4	Box character (upper middle)	Serifs, tick (horizontal axis)
5	Box character (upper right)	Frame
6	Box character (middle left)	Not used
7	Box character (middle middle)	Serifs
8	Box character (middle right)	Tick (vertical axis)
9	Box character (lower left)	Frame
10	Box character (lower middle)	Serifs
11	Box character (lower right)	Frame
12	Vertical bar	Control limits
13	Horizontal bar	Control limits
14	Box character (upper right)	Control limits

Table 19.3 *continued*

Value of *index*	Description of Character	Chart Feature
15	Box character (lower left)	Control limits
16	Box character (lower right)	Control limits
17	Box character (upper left)	Control limits

Not all printers can produce all the characters in the preceding list. By default, the form character list specified with the SAS system FORMCHAR= option is used; otherwise, the default is FORMCHAR='|—-|+|—|====='. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that the FORMCHAR= option in the PROC SHEWHART statement enables you to override temporarily the values of the SAS system option of the same name. The values of the SAS system option are not altered by using the FORMCHAR= option in the PROC SHEWHART statement.

GOUT=*graphics-catalog*

specifies the graphics catalog for traditional graphics output from PROC SHEWHART. This is useful if you want to save the output. The GOUT= option is used only when creating traditional graphics. This option is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option.

HISTORY=*SAS-data-set*

HIST=*SAS-data-set*

names an input data set that contains subgroup summary statistics. For example, you can read sample sizes, means, and ranges for the subgroups to create $\bar{X}$ and R charts. Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC SHEWHART, but it can also be created using a SAS summarization procedure such as the MEANS procedure.

Note that the HISTORY= data sets might need sorting. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted. The HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set together with a BOX=, DATA=, or TABLE= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "HISTORY= Data Set" subsection in the section for the chart statement you are using.

LIMITS=*SAS-data-set*
names an input data set that contains preestablished control limits or the parameters from which control limits can be computed. Each observation in a LIMITS= data set provides control limit information for a *process*. Typically, this data set is created as an OUTLIMITS= data set in a previous run of PROC SHEWHART.

If you omit the LIMITS= option, then control limits are computed from the data in the DATA= or HISTORY= input data sets or read from the BOX= or TABLE= input data sets. For details about the variables needed in a LIMITS= data set, see the "LIMITS= Data Set" subsection in the section for the chart statement you are using.

LINEPRINTER
requests that legacy line printer charts be produced. By default, PROC SHEWHART produces ODS Graphics output if ODS Graphics is enabled and traditional graphics output if ODS Graphics is disabled and SAS/GRAPH is licensed.

TABLE=*SAS-data-set*
names an input data set that contains subgroup summary statistics and control limits. Each observation in a TABLE= data set provides information for a particular subgroup and *process*. Typically, this data set is created as an OUTTABLE= data set in a previous run of PROC SHEWHART.

You cannot use a TABLE= data set together with a BOX=, DATA=, or HISTORY= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" subsection in the section for the chart statement that you are using.

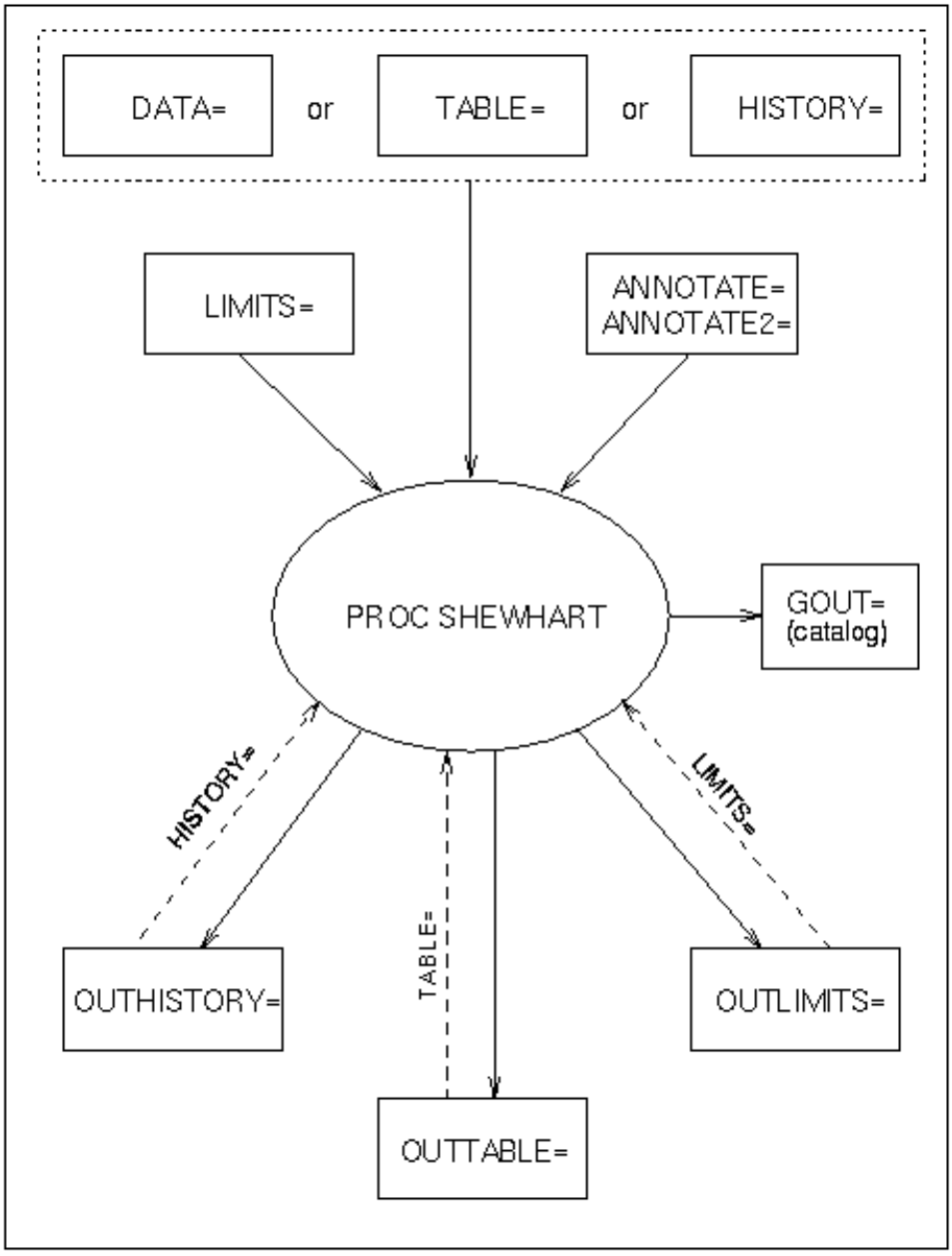
TESTHTML=*SAS-data-set*
names an input data set for creating links associated with tests for special causes when traditional graphics output is directed into HTML. A TESTHTML= data set contains variables _TEST_, _CHART_, and _URL_. _TEST_ and _CHART_ are numeric variables identifying a test for special causes (1-8) and the primary or secondary chart (1 or 2). _URL_ is a character variable containing the HTML syntax to create links associated with subgroups for which the given test on the given chart is positive. This option is ignored if you are not producing traditional graphics. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2183 for more information.

TESTURLS=*SAS-data-set*
names an input data set for associating URLs with tests for special causes when ODS Graphics output is directed into HTML. A TESTURLS= data set contains variables _TEST_, _CHART_, and _URL_. _TEST_ and _CHART_ are numeric variables identifying a test for special causes (1-8) and the primary or secondary chart (1 or 2). _URL_ is a character variable containing the URL to be associated with subgroups for which the given test on the given chart is positive. This option is ignored when ODS Graphics is disabled. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2183 for more information.

Input and Output Data Sets: SHEWHART Procedure

Figure 19.2 summarizes the input and output data sets used with the SHEWHART procedure.

Figure 19.2 Input and Output Data Sets in the SHEWHART Procedure



BOXCHART Statement: SHEWHART Procedure

Overview: BOXCHART Statement

The BOXCHART statement creates an $\bar{X}$ chart for subgroup means superimposed with box-and-whisker plots of the measurements in each subgroup. Throughout this chapter, a chart of this type is referred to as a *box chart*. This chart is recommended for large subgroup sample sizes (typically greater than ten). You can also use the BOXCHART statement to create standard side-by-side box-and-whisker plots (see Example 19.2 and Example 19.3).

You can use options in the BOXCHART statement to

- specify control limits for subgroup means or medians
- compute control limits from the data based on a multiple of the standard error of the means (or medians) or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 2100)
- specify one of several methods for calculating quantile statistics (percentiles)
- control the style of the box-and-whisker plots
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing box charts with the BOXCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: BOXCHART Statement

This section introduces the BOXCHART statement with simple examples that illustrate commonly used options. Complete syntax for the BOXCHART statement is presented in the section "Syntax: BOXCHART Statement" on page 1432, and advanced examples are given in the section "Examples: BOXCHART Statement" on page 1461.

Creating Box Charts from Raw Data

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

A petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set that contains the power output measurements for 20 days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   label KWatts='Average Power Output';
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
   end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
07JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457
07JUL94 3411 3350 3417 3629 3400 3381 3309 3608 3438 3567
08JUL94 3568 2968 3514 3465 3175 3358 3460 3851 3845 2983
08JUL94 3410 3274 3590 3527 3509 3284 3457 3729 3916 3633
09JUL94 3153 3408 3741 3203 3047 3580 3571 3579 3602 3335
```

```
09JUL94 3494 3662 3586 3628 3881 3443 3456 3593 3827 3573
10JUL94 3594 3711 3369 3341 3611 3496 3554 3400 3295 3002
10JUL94 3495 3368 3726 3738 3250 3632 3415 3591 3787 3478
11JUL94 3482 3546 3196 3379 3559 3235 3549 3445 3413 3859
11JUL94 3330 3465 3994 3362 3309 3781 3211 3550 3637 3626
12JUL94 3152 3269 3431 3438 3575 3476 3115 3146 3731 3171
12JUL94 3206 3140 3562 3592 3722 3421 3471 3621 3361 3370
13JUL94 3421 3381 4040 3467 3475 3285 3619 3325 3317 3472
13JUL94 3296 3501 3366 3492 3367 3619 3550 3263 3355 3510
14JUL94 3795 3872 3559 3432 3322 3587 3336 3732 3451 3215
14JUL94 3594 3410 3335 3216 3336 3638 3419 3515 3399 3709
15JUL94 3850 3431 3460 3623 3516 3810 3671 3602 3480 3388
15JUL94 3365 3845 3520 3708 3202 3365 3731 3840 3182 3677
16JUL94 3711 3648 3212 3664 3281 3371 3416 3636 3701 3385
16JUL94 3769 3586 3540 3703 3320 3323 3480 3750 3490 3395
17JUL94 3596 3436 3757 3288 3417 3331 3475 3600 3690 3534
17JUL94 3306 3077 3357 3528 3530 3327 3113 3812 3711 3599
18JUL94 3428 3760 3641 3393 3182 3381 3425 3467 3451 3189
18JUL94 3588 3484 3759 3292 3063 3442 3712 3061 3815 3339
19JUL94 3746 3426 3320 3819 3584 3877 3779 3506 3787 3676
19JUL94 3727 3366 3288 3684 3500 3501 3427 3508 3392 3814
20JUL94 3676 3475 3595 3122 3429 3474 3125 3307 3467 3832
20JUL94 3383 3114 3431 3693 3363 3486 3928 3753 3552 3524
21JUL94 3349 3422 3674 3501 3639 3682 3354 3595 3407 3400
21JUL94 3401 3359 3167 3524 3561 3801 3496 3476 3480 3570
22JUL94 3618 3324 3475 3621 3376 3540 3585 3320 3256 3443
22JUL94 3415 3445 3561 3494 3140 3090 3561 3800 3056 3536
23JUL94 3421 3787 3454 3699 3307 3917 3292 3310 3283 3536
23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
```

A partial listing of Turbine is shown in Figure 19.3. This data set is said to be in "strung-out" form because each observation contains the day and power output for a single heating. The first 20 observations contain the outputs for the first day, the second 20 observations contain the outputs for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWatts contains the output measurements and is referred to as the *process variable* (or *process* for short).

Figure 19.3 Partial Listing of the Data Set Turbine

Kilowatt Power Output Data

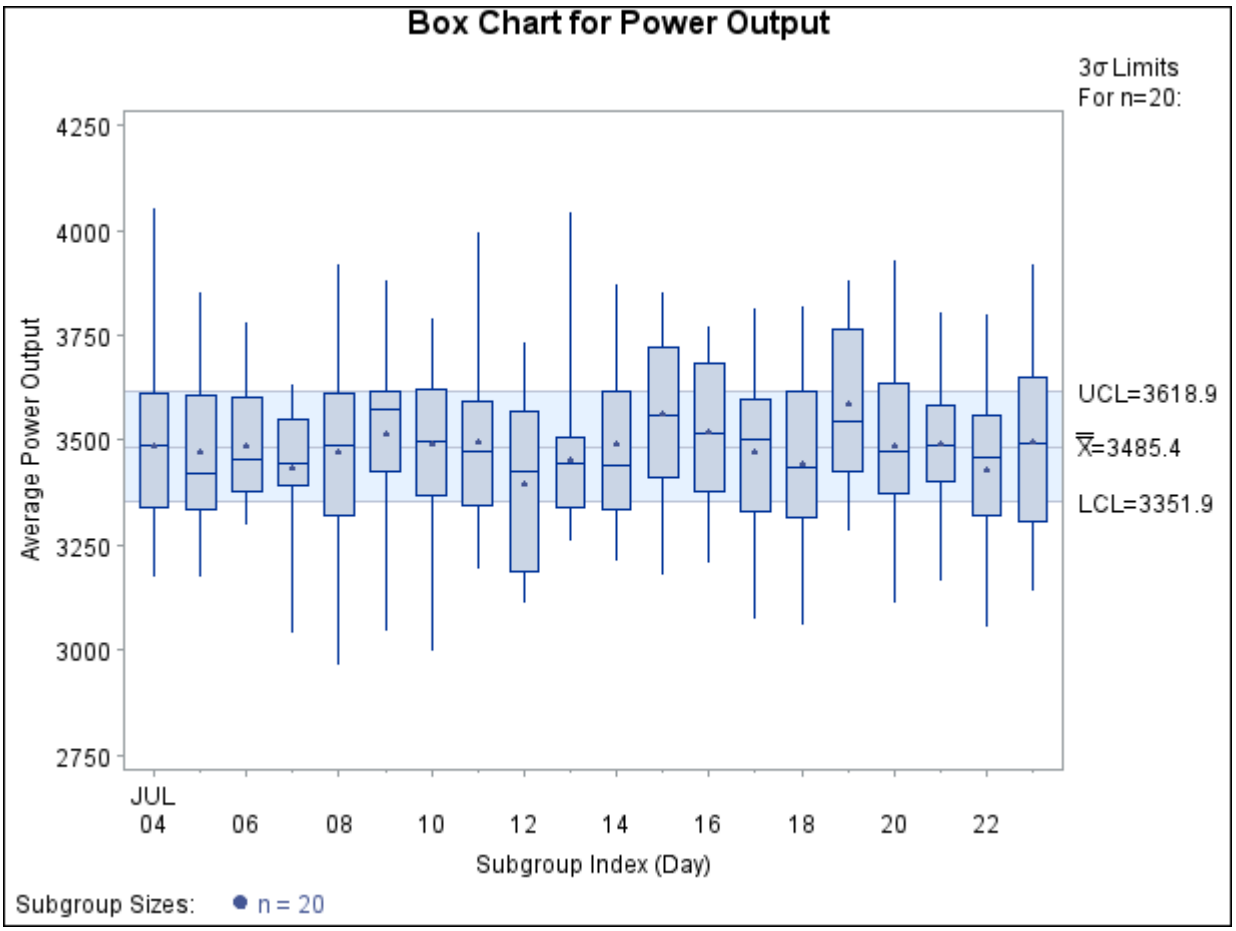
Obs	Day	KWatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
5	04JUL	3583
6	04JUL	3617
7	04JUL	3789
8	04JUL	3180
9	04JUL	3505
10	04JUL	3454
11	04JUL	3417
12	04JUL	3199
13	04JUL	3613
14	04JUL	3384
15	04JUL	3475
16	04JUL	3316
17	04JUL	3556
18	04JUL	3607
19	04JUL	3364
20	04JUL	3721
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
24	05JUL	3193
25	05JUL	3635

You can use a box chart to examine the distribution of power output for each day and to determine whether the mean level of the heating process is in control. The following statements create the box chart shown in Figure 19.4:

```
ods graphics off;
title 'Box Chart for Power Output';
symbol v=dot;
proc shewhart data=Turbine;
   boxchart KWatts*Day;
run;
```

This example illustrates the basic form of the BOXCHART statement. After the keyword BOXCHART, you specify the *process* to analyze (in this case, KWatts), followed by an asterisk and the *subgroup-variable* (Day).

Figure 19.4 Box Chart for Power Output Data (Traditional Graphics)



The input data set is specified with the DATA= option in the PROC SHEWHART statement.

By default, the BOXCHART statement requests an $\bar{X}$ chart superimposed with box-and-whisker plots for each subgroup. Table 19.4 lists the summary statistics represented by each plot. For details on the computation of percentiles, see "Percentile Definitions" on page 1460.

Table 19.4 Summary Statistics Represented by Box-and-Whisker Plots

Subgroup Summary Statistic	Feature of Box-and-Whisker Plot
Maximum	Endpoint of upper whisker
Third quartile (75th percentile)	Upper edge of box
Median (50th percentile)	Line inside box
Mean	Symbol marker (in this example, a dot)
First quartile (25th percentile)	Lower edge of box
Minimum	Endpoint of lower whisker

The within-subgroup variation in power output is stable, as indicated in Figure 19.4 by the edges of the boxes and the endpoints of the whiskers. Because the subgroup means, indicated by the dots, lie within the control limits, you can conclude that the heating process is in statistical control.

The skeletal style of the box-and-whisker plots shown in Figure 19.4 is the default. You can request different styles, as illustrated in Example 19.2. By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.7.

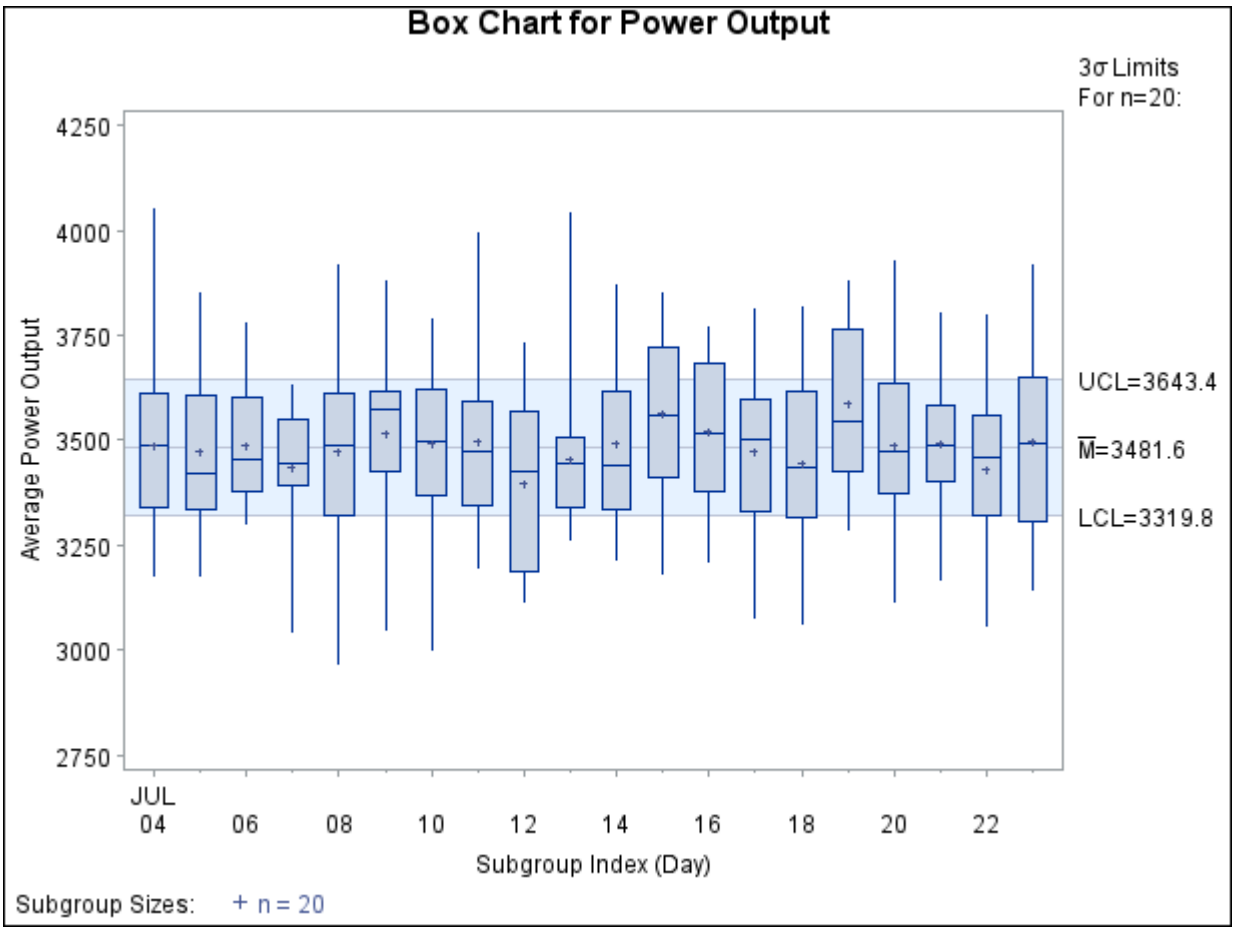
You can also create box charts in which the control limits apply to the subgroup medians. For example, the following statements create the chart shown in Figure 19.5:

```
title 'Box Chart for Power Output';
proc shewhart data=Turbine;
   boxchart KWatts*Day / controlstat = median;
run;
```

The CONTROLSTAT=MEDIAN option requests control limits that apply to the medians. Alternatively, you can specify the NOLIMITS option to suppress the display of control limits and create ordinary side-by-side box-and-whisker plots. See Example 19.2.

Options such as CONTROLSTAT= and NOLIMITS are specified after the slash (/) in the BOXCHART statement. A complete list of options is presented in the section "Syntax: BOXCHART Statement" on page 1432.

Figure 19.5 Box Chart for Power Output Data (Traditional Graphics)



Creating Box Charts from Subgroup Summary Data

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create box charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the BOXCHART statement with data of this type.

The following data set (Oilsum) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day).

```
data Oilsum;
   input Day KWattsL KWatts1 KWattsX KWattsM
             KWatts3 KWattsH KWattsR KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day    ='Date of Measurement'
         KWattsL='Minimum Power Output'
         KWatts1='25th Percentile'
         KWattsX='Average Power Output'
         KWattsM='Median Power Output'
         KWatts3='75th Percentile'
         KWattsH='Maximum Power Output'
         KWattsR='Range of Power Output'
         KWattsN='Subgroup Sample Size';
   datalines;
04JUL94 3180 3340.0 3487.40 3490.0 3610.0 4050 870 20
05JUL94 3179 3333.5 3471.65 3419.5 3605.0 3849 670 20
06JUL94 3304 3376.0 3488.30 3456.5 3604.5 3781 477 20
07JUL94 3045 3390.5 3434.20 3447.0 3550.0 3629 584 20
08JUL94 2968 3321.0 3475.80 3487.0 3611.5 3916 948 20
09JUL94 3047 3425.5 3518.10 3576.0 3615.0 3881 834 20
10JUL94 3002 3368.5 3492.65 3495.5 3621.5 3787 785 20
11JUL94 3196 3346.0 3496.40 3473.5 3592.5 3994 798 20
12JUL94 3115 3188.5 3398.50 3426.0 3568.5 3731 616 20
13JUL94 3263 3340.0 3456.05 3444.0 3505.5 4040 777 20
14JUL94 3215 3336.0 3493.60 3441.5 3616.0 3872 657 20
15JUL94 3182 3409.5 3563.30 3561.0 3719.5 3850 668 20
16JUL94 3212 3378.0 3519.05 3515.0 3682.5 3769 557 20
17JUL94 3077 3329.0 3474.20 3501.5 3599.5 3812 735 20
18JUL94 3061 3315.5 3443.60 3435.0 3614.5 3815 754 20
19JUL94 3288 3426.5 3586.35 3546.0 3762.5 3877 589 20
20JUL94 3114 3373.0 3486.45 3474.5 3635.5 3928 814 20
21JUL94 3167 3400.5 3492.90 3488.0 3582.5 3801 634 20
22JUL94 3056 3322.0 3432.80 3460.0 3561.0 3800 744 20
23JUL94 3145 3308.5 3496.90 3495.0 3652.0 3917 772 20
;
```

A partial listing of Oilsum is shown in Figure 19.6.

Figure 19.6 The Summary Data Set Oilsum

Summary Data Set for Power Outputs

Day	KWattsL	KWatts1	KWattsX	KWattsM	KWatts3	KWattsH	KWattsR	KWattsN
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	870	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	670	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	477	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	584	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	948	20

There are eight summary variables in Oilsum.

- KWattsL contains the subgroup minimums (low values).
- KWatts1 contains the 25th percentile (first quartile) for each subgroup.
- KWattsX contains the subgroup means.
- KWattsM contains the subgroup medians.
- KWatts3 contains the 75th percentile (third quartile) for each subgroup.
- KWattsH contains the subgroup maximums (high values).
- KWattsR contains the subgroup ranges.
- KWattsN contains the subgroup sample sizes.

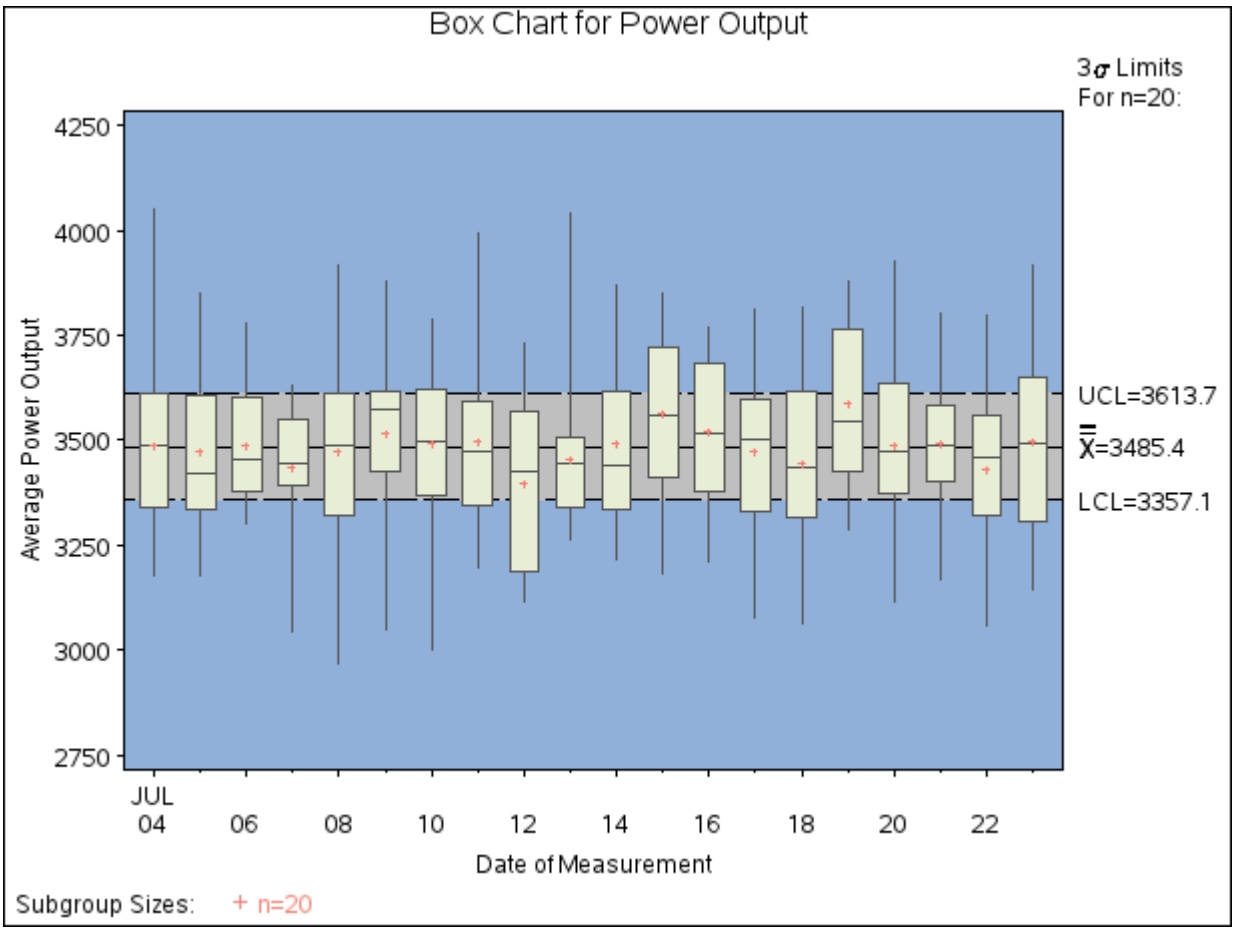
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as illustrated by the following statements, which create the box chart shown in Figure 19.7:

```
options nogstyle;
goptions ftext='albany amt';
symbol color = salmon h = .8;
title 'Box Chart for Power Output';
proc shewhart history=Oilsum;
   boxchart KWatts*Day / cinfill  = ligr
                         cboxfill = ywh
                         cboxes   = dagr
                         cframe   = vligb
                         ranges;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and BOXCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently.

Note that the *process* KWatts is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the eight summary variables. The suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *R*, and *N* indicate the contents of the variable. For example, the suffix characters *1* and *3* indicate first and third quartiles. The name Day specified after the asterisk is the name of the *subgroup-variable*.

Figure 19.7 Box Chart for Power Output Data (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the BOXCHART statement must contain the following variables:

- subgroup variable
- subgroup minimum variable
- subgroup first quartile variable
- subgroup mean variable
- subgroup median variable
- subgroup third quartile variable
- subgroup maximum variable
- subgroup sample size variable
- either a subgroup standard deviation variable or a subgroup range variable

Furthermore, the names of the summary variables must begin with the *process* name specified in the BOXCHART statement and end with the appropriate suffix character. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1885).

If you specify the RANGES option in the BOXCHART statement, the HISTORY= data set must contain a subgroup range variable; otherwise, the HISTORY= data set must contain a subgroup standard deviation variable. The RANGES option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup ranges rather than subgroup standard deviations. For example, in the following statements, the data set Oilsum2 must contain a subgroup standard deviation variable named KWattsS, because the RANGES option not specified:

```
title 'Box Chart for Power Output';
proc shewhart history=Oilsum2;
   boxchart KWatts*Day;
run;
```

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1455.

Saving Summary Statistics

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

In this example, the BOXCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Turbine and create a summary data set named Turbhist:

```
title 'Summary Data Set for Power Output';
proc shewhart data=Turbine;
   boxchart KWatts*Day / outhistory = Turbhist
                         nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.4.

Figure 19.8 contains a partial listing of Turbhist.

Figure 19.8 The Summary Data Set Turbhist

Summary Data Set for Power Output

Obs	Day	KWattsL	KWatts1	KWattsX	KWattsM	KWatts3	KWattsH	KWattsS	KWattsN
1	04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	220.260	20
2	05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	210.427	20
3	06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	147.025	20
4	07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	157.637	20
5	08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	258.949	20

There are nine variables in the data set Turbhist.

- Day is the subgroup variable.
- KWattsL contains the subgroup minimums.
- KWatts1 contains the first quartiles for each subgroup.
- KWattsX contains the subgroup means.
- KWattsM contains the subgroup medians.
- KWatts3 contains the third quartiles for each subgroup.
- KWattsH contains the subgroup maximums.
- KWattsS contains the subgroup standard deviations.
- KWattsN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* to the *process* KWatts specified in the BOXCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the RANGES option, the OUTHISTORY= data set includes a subgroup range variable, rather than a subgroup standard deviation variable, as demonstrated by the following statements:

```
proc shewhart data=Turbine;
   boxchart KWatts*Day / outhistory = Turbhist2
                         ranges
                         nochart;
run;
```

Figure 19.9 contains a partial listing of Turbhist2. The variable KWattsR contains the subgroup ranges.

The RANGES option is not recommended when the subgroup sample sizes are greater than 10, nor when you use the NOLIMITS option to create standard side-by-side box-and-whisker plots.

For more information, see "OUTHISTORY= Data Set" on page 1451.

Figure 19.9 The Summary Data Set Turbhist2

Summary Data Set for Power Output

Day	KWattsL	KWatts1	KWattsX	KWattsM	KWatts3	KWattsH	KWattsR	KWattsN
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	870	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	670	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	477	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	584	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	948	20

Saving Control Limits

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a box chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1431) or modify the limits with a DATA step program.

The following statements read measurements from the data set Turbine (see "Creating Box Charts from Raw Data" on page 1418) and save the control limits displayed in Figure 19.4 in a data set named Turblim:

```
proc shewhart data=Turbine;
   boxchart KWatts*Day / outlimits=Turblim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Turblim is listed in Figure 19.10.

Figure 19.10 The Data Set Turblim Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	.002699796	3	3351.92	3485.41	3618.90

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

The data set Turblim contains one observation with the limits for *process* KWatts. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLS_, _S_, and _UCLS_ are not used to create box charts, but they are included so that the data set Turblim can be used to create an *s* chart; see "XSCHART Statement: SHEWHART Procedure" on page 1925. If you specify the RANGES option in the BOXCHART statement, the variables _LCLR_, _R_, and _UCLR_, rather than the variables _LCLS_, _S_, and _UCLS_, are included in the OUTLIMITS= data set. These variables can be used to create an *R* chart; see "XRCHART Statement: SHEWHART Procedure" on page 1881.

If you specify CONTROLSTAT=MEDIAN to request control limits for medians, the variables _LCLM_ and _UCLM_, rather than the variables _LCLX_ and _UCLX_, are included in the OUTLIMITS= data set as demonstrated by the following statements:

```
proc shewhart data=Turbine;
   boxchart KWatts*Day / outlimits   = Turblim2
                         controlstat = median
                         nochart;
run;
```

Turblim2 is listed in Figure 19.11. For more information, see "OUTLIMITS= Data Set" on page 1448.

Figure 19.11 The Data Set Turblim2 Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_
KWatts	Day	ESTIMATE	20	.002776264	3	3319.85	3481.63	3643.40

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limit Information';
proc shewhart data=Turbine;
   boxchart KWatts*Day / outtable=Turbtab
                         nochart;
run;
```

The data set Turbtab is partially listed in Figure 19.12.

Figure 19.12 The OUTTABLE= Data Set Turbtab

Summary Statistics and Control Limit Information

VAR	Day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
KWatts	04JUL	3	20	20	3351.92	3487.40	3485.41	3618.90	198.996
KWatts	05JUL	3	20	20	3351.92	3471.65	3485.41	3618.90	198.996
KWatts	06JUL	3	20	20	3351.92	3488.30	3485.41	3618.90	198.996
KWatts	07JUL	3	20	20	3351.92	3434.20	3485.41	3618.90	198.996
KWatts	08JUL	3	20	20	3351.92	3475.80	3485.41	3618.90	198.996
KWatts	09JUL	3	20	20	3351.92	3518.10	3485.41	3618.90	198.996
KWatts	10JUL	3	20	20	3351.92	3492.65	3485.41	3618.90	198.996
KWatts	11JUL	3	20	20	3351.92	3496.40	3485.41	3618.90	198.996
KWatts	12JUL	3	20	20	3351.92	3398.50	3485.41	3618.90	198.996
KWatts	13JUL	3	20	20	3351.92	3456.05	3485.41	3618.90	198.996
KWatts	14JUL	3	20	20	3351.92	3493.60	3485.41	3618.90	198.996
KWatts	15JUL	3	20	20	3351.92	3563.30	3485.41	3618.90	198.996
KWatts	16JUL	3	20	20	3351.92	3519.05	3485.41	3618.90	198.996
KWatts	17JUL	3	20	20	3351.92	3474.20	3485.41	3618.90	198.996
KWatts	18JUL	3	20	20	3351.92	3443.60	3485.41	3618.90	198.996
KWatts	19JUL	3	20	20	3351.92	3586.35	3485.41	3618.90	198.996
KWatts	20JUL	3	20	20	3351.92	3486.45	3485.41	3618.90	198.996
KWatts	21JUL	3	20	20	3351.92	3492.90	3485.41	3618.90	198.996
KWatts	22JUL	3	20	20	3351.92	3432.80	3485.41	3618.90	198.996
KWatts	23JUL	3	20	20	3351.92	3496.90	3485.41	3618.90	198.996

EXLIM	_SUBMIN_	_SUBQ1_	_SUBMED_	_SUBQ3_	_SUBMAX_
	3180	3340.0	3490.0	3610.0	4050
	3179	3333.5	3419.5	3605.0	3849
	3304	3376.0	3456.5	3604.5	3781
	3045	3390.5	3447.0	3550.0	3629
	2968	3321.0	3487.0	3611.5	3916
	3047	3425.5	3576.0	3615.0	3881
	3002	3368.5	3495.5	3621.5	3787
	3196	3346.0	3473.5	3592.5	3994
	3115	3188.5	3426.0	3568.5	3731
	3263	3340.0	3444.0	3505.5	4040
	3215	3336.0	3441.5	3616.0	3872
	3182	3409.5	3561.0	3719.5	3850
	3212	3378.0	3515.0	3682.5	3769
	3077	3329.0	3501.5	3599.5	3812
	3061	3315.5	3435.0	3614.5	3815
	3288	3426.5	3546.0	3762.5	3877
	3114	3373.0	3474.5	3635.5	3928
	3167	3400.5	3488.0	3582.5	3801
	3056	3322.0	3460.0	3561.0	3800
	3145	3308.5	3495.0	3652.0	3917

This data set contains one observation for each subgroup sample. The variable _SUBMIN_ contains the subgroup minimums, and the variable _SUBQ1_ contains the first quartile for each subgroup. The variable _SUBX_ contains the subgroup means, and the variable _SUBMED_ contains the subgroup medians. The variable _SUBQ3_ contains the third quartiles, and the variable _SUBMAX_ contains the subgroup maximums. The variable _SUBN_ contains the subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means. The variable _MEAN_ contains the central line. The variables _VAR_ and Day contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1452.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Turbtab and display a box chart (not shown here) that is identical to the chart in Figure 19.4:

```
title 'Box Chart for Power Output';
proc shewhart table=Turbtab;
   boxchart KWatts*Day;
label _SUBX_ = 'Average Power Output';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143).

For more information, see "TABLE= Data Set" on page 1456.

Reading Preestablished Control Limits

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Turblim saved control limits computed from the measurements in Turbine. This example shows how these limits can be applied to new data. The following statements create the box chart in Figure 19.13 using new measurements in a data set named Turbine2 (not listed here) and the control limits in Turblim:

```
title 'Box Chart for Power Output';
ods graphics on;
proc shewhart data=Turbine2 limits=Turblim;
   boxchart KWatts*Day / odstitle=title;
run;
```

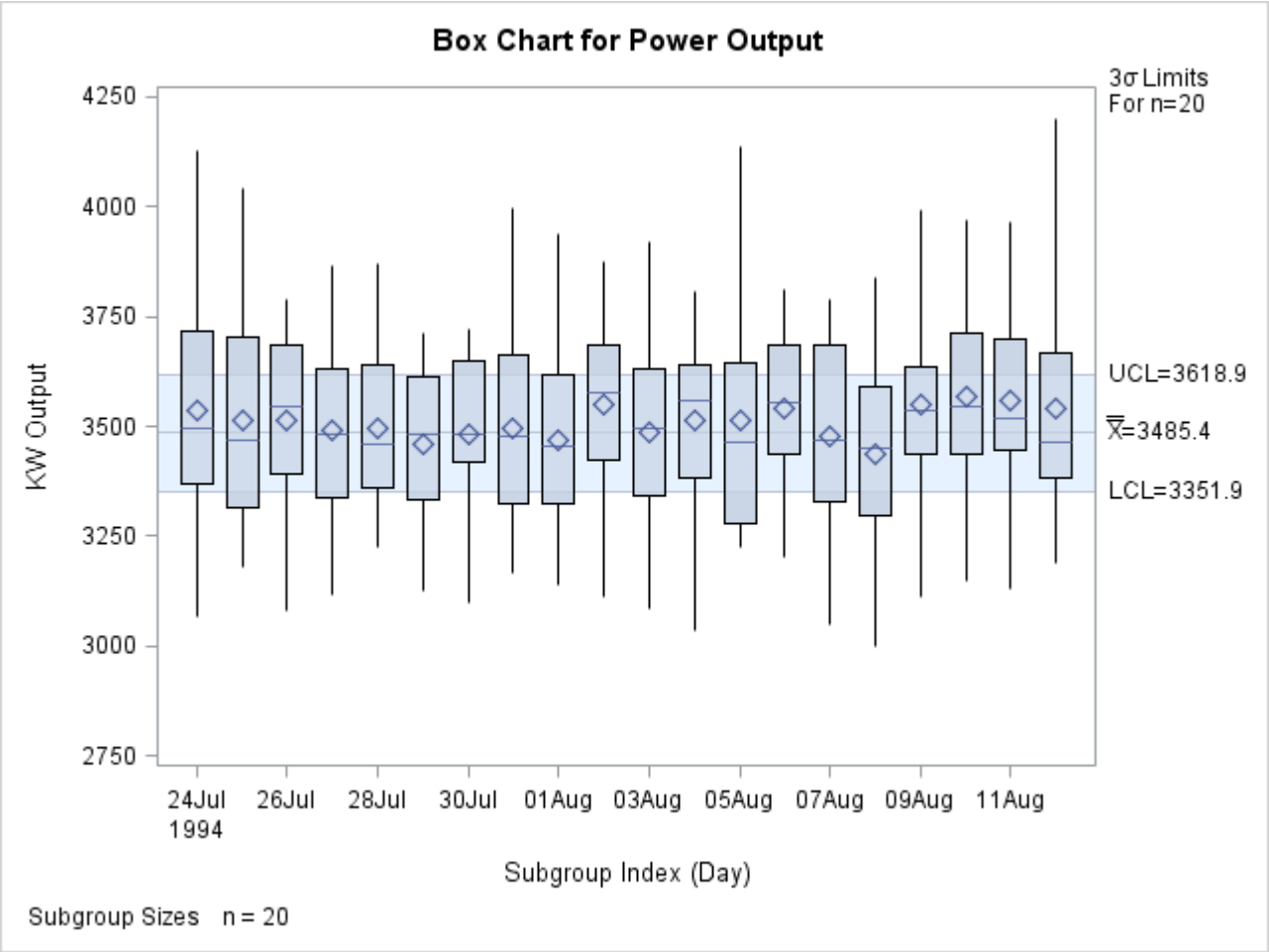
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the box chart is created by using ODS Graphics instead of traditional graphics.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWatts
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

The chart reveals an increase in variability beginning on August 1.

Figure 19.13 Box Chart for Second Set of Power Outputs (ODS Graphics)



In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1454 for details concerning the variables that you must provide.

Syntax: BOXCHART Statement

The basic syntax for the BOXCHART statement is as follows:

BOXCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

BOXCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of BOXCHART statements in the SHEWHART procedure. The components of the BOXCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Box Charts from Raw Data" on page 1418.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Box Charts from Subgroup Summary Data" on page 1423.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1428.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct box charts for Weight, Length, and Width:

```
proc shewhart data=summary;
   boxchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding BOXCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a box chart using an asterisk (*) to plot the means:

```
proc shewhart data=values lineprinter;
   boxchart weight*day='*';
run;
```

options

enhance the appearance of the box chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the BOXCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.5 BOXCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for control charts
CONTROLSTAT=	Specifies whether control limits are computed for subgroup means or subgroup medians
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than from a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	Reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted statistic
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit on box chart
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default labels for control limits and central line in box chart

Table 19.5 *continued*

Option	Description
NOCTL	Suppresses display of central line in box chart
NOLCL	Suppresses display of lower control limit in box chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit in box chart
UCLLABEL=	Specifies label for upper control limit in box chart
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line in box chart
Process Mean and Standard Deviation Options	
MEDCENTRAL=	Specifies method for estimating process mean μ
MU0=	Specifies known value of μ_0 for process mean μ
RANGES	Specifies that estimate of process standard deviation σ is to be calculated from subgroup ranges
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Controlling Box Appearance	
BOXCONNECT=	Connects subgroup means, medians, maximum values, minimum values, or quartiles in box-and-whisker plots
BOXSTYLE=	Specifies style of box-and-whisker plots
BOXWIDTH=	Specifies width of box-and-whisker plots
BOXWIDTHSCALE=	Specifies that widths of box-and-whisker plots vary proportionately to subgroup sample size
CBOXES=	Specifies color for outlines of box-and-whisker plots
CBOXFILL=	Specifies fill color for interior of box-and-whisker plots
IDCOLOR=	Specifies outlier symbol color in schematic box-and-whisker plots
IDCTEXT=	Specifies text color to label outliers or process variable values
IDFONT=	Specifies text font to label outliers or process variable values
IDHEIGHT=	Specifies text height to label outliers or process variable values

Table 19.5 *continued*

Option	Description
IDSYMBOL=	Specifies outlier symbol in schematic box-and-whisker plots
IDSYMBOLHEIGHT=	Specifies outlier symbol height in schematic box-and-whisker plots
LBOXES=	Specifies line types for outlines of box-and-whisker plots
NOTCHES	Specifies that box-and-whisker plots are to be notched
PCTLDEF=	Specifies percentile definition used for box-and-whisker plots
SERIFS	Adds serifs to the whiskers of skeletal box-and-whisker plots
WHISKERPERCENTILE=	Specifies that whiskers be drawn to percentile values
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on box chart
ALLLABEL2=	Labels every point on trend chart
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CLABEL=	Specifies color for labels
COUT=	Specifies color for portions of line segments that connect points outside control limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NOCONNECT	Suppresses line segments that connect points on chart
NOTRENDCONNECT	Suppresses line segments that connect points on trend chart
OUTLABEL=	Labels points outside control limits on box chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes for the box chart
TEST2RUN=	Specifies length of pattern for Test 2

Table 19.5 *continued*

Option	Description
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset for the box chart
WESTGARD=	Requests that Westgard rules be applied to the box chart
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines to box chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis

Table 19.5 *continued*

Option	Description
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of box chart
VAXIS2=	Specifies major tick mark values for vertical axis of trend chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots summary statistics for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process variable only when exceptions occur
INTERVAL=	Specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable

Table 19.5 *continued*

Option	Description
MAXPANELS=	Specifies maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of box chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
TRENDVAR=	Specifies list of trend variables
YPCT1=	Specifies length of vertical axis on box chart as a percentage of sum of lengths of vertical axes for box and trend charts
ZEROSTD	Displays box chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on box chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on box chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on box chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on trend chart

Table 19.5 *continued*

Option	Description
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	Specifies position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to box chart
ANNOTATE2=	Specifies annotate data set that adds features to trend chart
DESCRIPTION=	Specifies description of box chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of box chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
WTREND=	Specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies variables whose values define colors for filling background of *block-variable* legend
BOXES=	Specifies variables whose values define colors box outlines

Table 19.5 *continued*

Option	Description
BOXFILL=	Specifies variables whose values define colors for filling boxes
CFRAMELAB	Draws a frame around labeled points
CPHASEBOX	Requests boxes enclosing all plotted points for a phase
CPHASEBOXCONNECT	Requests lines connecting adjacent enclosing boxes
CPHASEBOXFILL	Fills boxes enclosing all plotted points for a phase
CPHASEMEANCONNECT	Requests lines connecting phase average value points
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
BOXTRANSPARENCY=	Specifies the box fill transparency for box-and-whisker charts
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOBOXFILLLEGEND	Suppresses legend for levels of a BOXFILL= variable
NOFILLLEGEND	Suppresses legend for levels of a BOXFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTHIGHURL=	Specifies variable whose values are URLs to be associated with outliers above the upper fence on a schematic box chart
OUTLOWURL=	Specifies variable whose values are URLs to be associated with outliers below the lower fence on a schematic box chart
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEBOXLABELS	Draws phase labels as titles along the top of phase boxes
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases

Table 19.5 *continued*

Option	Description
POINTSURL=	Specifies variable whose values are URLs to be associated with points representing individual observations
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
WBOXES=	Specifies width of box outlines for box-and-whisker charts
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTBOX=	Creates output data set containing subgroup summary statistics, control limits, and outlier values for box chart
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup values, subgroup sample sizes, subgroup summary statistics, and control limits
TABLEALL	is equivalent to the options TABLE, TABLEBOX, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETEST
TABLEBOX	Augments basic table with columns for minimum, 25th percentile, median, 75th percentile, and maximum of observations in a subgroup
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded

Table 19.5 *continued*

Option	Description
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASEBOX=	Specifies color for box enclosing all plotted points for a phase
CPHASEBOXCONNECT=	Specifies color for line segments connecting adjacent enclosing boxes
CPHASEBOXFILL=	Specifies fill color for box enclosing all plotted points for a phase
CPHASELEG=	Specifies text color for *phase* legend
CPHASEMEANCONNECT=	Specifies color for line segments connecting average value points within a phase
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEMEANSYMBOL=	Specifies symbol marker for average of values within a phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set

Table 19.5 *continued*

Option	Description
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies variable whose values create links to be associated with subgroups on primary chart
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies variable whose values create links to be associated with symbols in the symbol legend
OUTHIGHHTML=	Specifies variable whose values create links to be associated with outliers above the upper fence on a schematic box chart

Table 19.5 *continued*

Option	Description
OUTLOWHTML=	Specifies variable whose values create links to be associated with outliers below the lower fence on a schematic box chart
POINTSHTML=	Specifies variable whose values create links to be associated with points representing individual observations
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: BOXCHART Statement

The following sections provide details that are specific to the BOXCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Box Charts

The following notation is used in this section:

μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	Mean of measurements in ith subgroup
n_i	Sample size of ith subgroup
N	The number of subgroups
x_{ij}	jth measurement in the ith subgroup, $j = 1, 2, 3 \ldots, n_i$
$x_{i(j)}$	jth largest measurement in the ith subgroup:

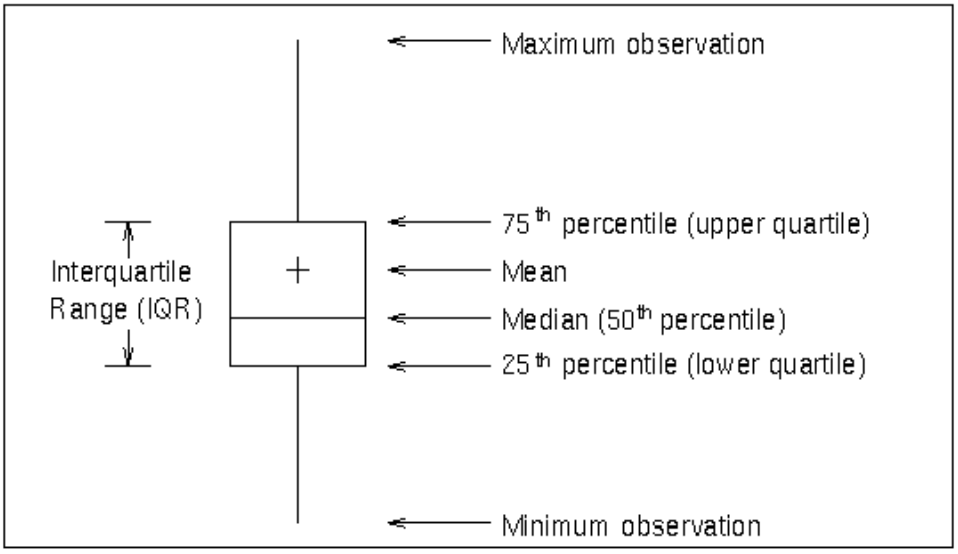
$$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$

$\overline{\overline{X}}$	Weighted average of subgroup means
M_i	Median of the measurements in the ith subgroup: $$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$
$\bar{M}$	Average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$
$\tilde{M}$	Median of the subgroup medians. Denote the jth largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	Standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	$100 \times p$th percentile ($0 < p < 1$) of the distribution of the median of n independent observations from a normal population with unit standard deviation
z_p	$100 \times p$th percentile of the standard normal distribution
$D_p(n)$	$100 \times p$th percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Elements of Box-and-Whisker Plots

A box-and-whisker plot is displayed for the measurements in each subgroup on the box chart. Figure 19.14 illustrates the elements of each plot.

Figure 19.14 Box-and-Whisker Plot



The skeletal style of the box-and-whisker plot shown in Figure 19.14 is the default. You can specify alternative styles with the BOXSTYLE= option; see Example 19.2 or the entry for BOXSTYLE= in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Control Limits and Central Line

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of $\bar{X}_i$ (or M_i) above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ (or M_i) exceeds the limits

The CONTROLSTAT= option specifies whether control limits are computed for subgroup means (the default) or subgroup medians. Table 19.7 provides the formulas for the limits.

Table 19.7 Limits and Central Line for Box Charts

Control Limits **CONTROLSTAT=MEAN**	**CONTROLSTAT=MEDIAN**
LCLX = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$	LCLM = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$
Central Line = $\overline{\overline{X}}$	Central Line = $\bar{M}$
UCLX = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$	UCLM = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$
Probability Limits **CONTROLSTAT=MEAN**	**CONTROLSTAT=MEDIAN**
LCLX = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$	LCLM = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$
Central Line = $\overline{\overline{X}}$	Central Line = $\bar{M}$
UCLX = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$	UCLM = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$

In the preceding tables, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN in addition to CONTROLSTAT=MEDIAN. Likewise, replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED in addition to CONTROLSTAT=MEDIAN. If standard values μ_0 and σ_0 are available for μ and σ, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 19.7.

Note that the limits vary with n_i. The formulas for median limits assume that the data are normally distributed.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

NOTE: You can suppress the display of the control limits with the NOLIMITS option. This is useful for creating standard side-by-side box-and-whisker plots.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 19.8 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_CP_	Capability index C_p
CPK	Capability index C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLM	Lower control limit for subgroup median
LCLR	Lower control limit for subgroup range
LCLS	Lower control limit for subgroup standard deviation
LCLX	Lower control limit for subgroup mean
LIMITN	Nominal sample size associated with the control limits
LSL	Lower specification limit
MEAN	Process mean (value of central line on box chart)
_R_	Value of central line on R chart
_S_	Value of central line on s chart

Table 19.8 *continued*

Variable	Description
SIGMAS	Multiple (k) of standard error of $\bar{X}_i$ or M_i
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the BOXCHART statement
TARGET	Target value
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	Upper control limit for subgroup median
UCLR	Upper control limit for subgroup range
UCLS	Upper control limit for subgroup standard deviation
UCLX	Upper control limit for subgroup mean
USL	Upper specification limit
VAR	*Process* specified in the BOXCHART statement

Notes:

1. The variables _LCLM_ and _UCLM_ are included if you specify CONTROLSTAT=MEDIAN; otherwise, the variables _LCLX_ and _UCLX_ are included.

2. The variables _LCLR_, _R_, and _UCLR_ are included if you specify the RANGES option; otherwise, the variables _LCLS_, _S_, and _UCLS_ are included. These variables are not used to create box charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the XRCHART, XSCHART, MRCHART, SCHART, and RCHART statements.

3. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLM_, _UCLM_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.

4. If the limits are defined in terms of a multiple k of the standard error of $\bar{X}_i$, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function. If the limits are defined in terms of a multiple k of the standard error of M_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _ALPHA_.

5. If the limits for means are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1-\alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function. If the limits for medians are probability limits, the value of _SIGMAS_ is computed as $k = F^{-1}_{med}(1-\alpha/2, n)$, where $F^{-1}_{med}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.

6. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

7. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the BOXCHART statement. For an example, see "Saving Control Limits" on page 1428.

OUTBOX= Data Set

The OUTBOX= data set saves subgroup summary statistics, control limits, and outlier values. The following variables can be saved:

- the *subgroup-variable*
- the variable _VAR_, containing the process variable name
- the variable _TYPE_, identifying features of box-and-whisker plots
- the variable _VALUE_, containing values of box-and-whisker plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing links associated with box-and-whisker plot features

_ID_ is included in the OUTBOX= data set only if one of the keywords SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if one or more of the HTML=, OUTHIGHHTML=, OUTLOWHTML=, or POINTSHTML= options are specified.

Each observation in an OUTBOX= data set records the value of a single feature of one subgroup's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. Table 19.9 lists valid _TYPE_ variable values:

Table 19.9 Valid _TYPE_

Value	Description
N	Subgroup size
SIGMAS	Multiple (k) of standard error of $\bar{X}_i$ or M_i
ALPHA	Probability (α) of exceeding limits
LIMITN	Nominal sample size associated with control limits
LCLM	Lower control limit for subgroup median
LCLX	Lower control limit for subgroup mean
UCLM	Upper control limit for subgroup median
UCLX	Upper control limit for subgroup mean
PROCMED	Process median
PROCMEAN	Process mean
EXLIM	Control limit exceeded on box chart
TREND	Trend variable value
MIN	Minimum subgroup value
Q1	Subgroup first quartile
MEDIAN	Subgroup median
MEAN	Subgroup mean
Q3	Subgroup third quartile
MAX	Subgroup maximum value
LOW	Low outlier value

Table 19.9 *continued*

Value	Description
HIGH	High outlier value
LOWHISKR	Low whisker value, if different from MIN
HIWHISKR	High whisker value, if different from MAX
FARLOW	Low far outlier value
FARHIGH	High far outlier value

Additionally, the following variables, if specified, are included:

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup minimum variable named by the prefix *process* suffixed with *L*
- a subgroup first-quartile variable named by the prefix *process* suffixed with *1*
- a subgroup mean variable named by the prefix *process* suffixed with *X*
- a subgroup median variable named by the prefix *process* suffixed with *M*
- a subgroup third-quartile variable named by the prefix *process* suffixed with *3*
- a subgroup maximum variable named by the prefix *process* suffixed with *H*
- a subgroup sample size variable named by the prefix *process* suffixed with *N*
- a subgroup range variable named by the prefix *process* suffixed with *R* or a subgroup standard deviation variable named by *process* suffixed with *S*

A subgroup range variable is included if you specify the RANGES option; otherwise, a subgroup standard deviation variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the BOXCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   boxchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthL, Width1, WidthM, WidthX, Width3, WidthH, WidthS, WidthN, DiameterL, Diameter1, DiameterM, DiameterX, Diameter3, DiameterH, DiameterS, and DiameterN.

The variables WidthS and DiameterS are included because the RANGES option is not specified. If you specified the RANGES option, the data set Summary would contain the variables WidthR and DiameterR rather than WidthS and DiameterS.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1426.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.10 list the variables that can be saved.

Table 19.10 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on box chart
LCLM	Lower control limit for median
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBMAX	Subgroup maximum
SUBMED	Subgroup median
SUBMIN	Subgroup minimum
SUBN	Subgroup sample size
SUBQ1	Subgroup first quartile (25th percentile)
SUBQ3	Subgroup third quartile (75th percentile)
SUBX	Subgroup mean
TESTS	Tests for special causes signaled on box chart

Variable	Description
UCLM	Upper control limit for median
UCLX	Upper control limit for mean
VAR	*Process* specified in the BOXCHART statement

The variables _LCLM_ and _UCLM_ are included if you specify CONTROLSTAT=MEDIAN; otherwise, the variables _LCLX_ and _UCLX_ are included. In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1428.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the BOXCHART statement must be a SAS variable in the data set. This variable provides measurements which must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, specified in the BOXCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i measurements, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains 20 items and there are 30 subgroup samples, the DATA= data set should contain 600 observations. Other variables that can be read from a DATA= data set include

- _PHASE_ (if READPHASES= is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating Box Charts from Raw Data" on page 1418.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   boxchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 19.8. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, and _UCLX_ or (if you specify CONTROLSTAT=MEDIAN) the variables _LCLM_, _MEAN_, and _UCLM_. These variables specify the control limits directly.
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 19.7

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1431.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the BOXCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup minimum variable for each *process*
- a subgroup first-quartile variable for each *process*
- a subgroup median variable for each *process*
- a subgroup mean variable for each *process*
- a subgroup third-quartile variable for each *process*
- a subgroup maximum variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or a subgroup standard deviation variable for each *process*

If you specify the RANGES option, the subgroup range variable must be included; otherwise, the subgroup standard deviation variable must be included.

The names of the subgroup summary statistics variables must be the *process* name concatenated with the following special suffix characters:

Subgroup Summary Statistic	Suffix Character
subgroup Minimum	L
subgroup First-quartile	1
subgroup Median	M
subgroup Mean	X
subgroup Third-quartile	3
subgroup Maximum	H
subgroup Sample size	N
subgroup Range	R
subgroup Standard deviation	S

For example, consider the following statements:

```
proc shewhart history=summary;
   boxchart (weight Yieldstrength)*batch;
run;
```

The data set Summary must include the variables Batch, WeightL, Weight1, WeightM, WeightX, Weight3, WeightH, WeightS, WeightN, YieldstrengthL, Yieldstrength1, YieldstrengthM, YieldstrengthX, Yieldstrength3, YieldstrengthH, YieldstrengthS, and YieldstrengthN.

If the RANGES option were specified in the preceding BOXCHART statement, it would be necessary for Summary to include the variables WeightR and YieldstrengthR rather than WeightS and YieldstrengthS.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if READPHASES= is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Box Charts from Subgroup Summary Data" on page 1423.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.11 lists the variables required in a TABLE= data set used with the BOXCHART statement:

Table 19.11 Variables Required in a TABLE= Data Set

Variable	Description
LCLM	Lower control limit for median
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
Subgroup-variable	Values of the *subgroup-variable*
SUBMAX	Subgroup maximum
SUBMIN	Subgroup minimum

Table 19.11 *continued*

Variable	Description
SUBMED	Subgroup median
SUBN	Subgroup sample size
SUBQ1	Subgroup first quartile (25th percentile)
SUBQ3	Subgroup third quartile (75th percentile)
SUBX	Subgroup mean
UCLM	Upper control limit for median
UCLX	Upper control limit for mean

Note that if you specify CONTROLSTAT=MEDIAN, the variables _LCLM_, _SUBMED_, and _UCLM_ are required; otherwise, the variables _LCLX_, _SUBX_, and _UCLX_ are required.

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1428.

BOX= Data Set

You can read summary statistics, control limits, and outlier values from a BOX= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTBOX= data set created in a previous run of the SHEWHART procedure to display a box chart.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the process variable name
- _TYPE_, identifying features of box-and-whisker plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one subgroup's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. Table 19.12 lists valid _TYPE_ variable values:

Table 19.12 Valid _TYPE_ Values in a BOX= Data Set

Value	Description
N	Subgroup size
SIGMAS	Multiple (k) of standard error of $\bar{X}_i$ or M_i
ALPHA	Probability (α) of exceeding limits
LIMITN	Nominal sample size associated with control limits
LCLM	Lower control limit for subgroup median
LCLX	Lower control limit for subgroup mean
UCLM	Upper control limit for subgroup median
UCLX	Upper control limit for subgroup mean
PROCMED	Process median
PROCMEAN	Process mean
EXLIM	Control limit exceeded on box chart
TREND	Trend variable value
MIN	Minimum subgroup value
Q1	Subgroup first quartile
MEDIAN	Subgroup median
MEAN	Subgroup mean
Q3	Subgroup third quartile
MAX	Subgroup maximum value
LOW	Low outlier value
HIGH	High outlier value
LOWHISKR	Low whisker value, if different from MIN
HIWHISKR	High whisker value, if different from MAX
FARLOW	Low far outlier value
FARHIGH	High far outlier value

The features identified by the _TYPE_ values N, LCLM or LCLX, UCLM or UCLX, PROCMED or PROCMEAN, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each subgroup.

Other variables that can be read from a BOX= data set include:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing links to be associated with features on box plots
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_ does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MVLUE and RMSDF) are available for estimating the process standard deviation σ. The method depends on whether you specify the RANGES option. If you specify this option, σ is estimated using subgroup ranges; otherwise, σ is estimated using subgroup standard deviations.

Default Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option, the default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

Default Method Based on Subgroup Ranges

If you specify the RANGES option, the default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option and specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVLUE Method Based on Subgroup Ranges

If you specify the RANGES option and SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option and specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Percentile Definitions

You can use the PCTLDEF= option to specify one of five definitions for computing quantile statistics (percentiles). Let n equal the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the process variable. For the tth percentile, set $p = t/100$, and express np as

$$np = j + g$$

where j is the integer part of np, and g is the fractional part of np.

The tth percentile (call it y) can be defined in five ways, as described in the next five sections.

PCTLDEF=1

This uses the weighted average at x_{np}

$$y = (1 - g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1.

PCTLDEF=2

This uses the observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$.

PCTLDEF=3

This uses the empirical distribution function

$$\begin{array}{ll} y = x_j & \text{if } g = 0 \\ y = x_{j+1} & \text{if } g > 0 \end{array}$$

PCTLDEF=4

This uses the weighted average aimed at $x_{p(n+1)}$

$$y = (1 - g)x_j + gx_{j+1}$$

where $(n + 1)p = j + g$, and where x_{n+1} is taken to be x_n.

PCTLDEF=5

This uses the empirical distribution function with averaging

$$\begin{array}{ll} y = (x_j + x_{j+1})/2 & \text{if } g = 0 \\ y = x_{j+1} & \text{if } g > 0 \end{array}$$

Examples: BOXCHART Statement

This section provides advanced examples of the BOXCHART statement.

Example 19.1: Using Box Charts to Compare Subgroups

NOTE: See *Using Box Charts to Compare Subgroups* in the SAS/QC Sample Library.

In this example, a box chart is used to compare the delay times for airline flights during the Christmas holidays with the delay times prior to the holiday period. The following statements create a data set named Times with the delay times in minutes for 25 flights each day. When a flight is cancelled, the delay is recorded as a missing value.

```
data Times;
   informat Day date7. ;
   format   Day date7. ;
   input Day @ ;
   do Flight=1 to 25;
      input Delay @ ;
      output;
   end;
   datalines;
16DEC88   4  12   2   2  18   5   6  21   0   0
          0  14   3   .   2   3   5   0   6  19
          7   4   9   5  10
17DEC88   1  10   3   3   0   1   5   0   .   .
          1   5   7   1   7   2   2  16   2   1
          3   1  31   5   0
18DEC88   7   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   3  12   8   6
          2   7   2   4   5
19DEC88  15   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  20   8   1  14   3
         10   0   1  11   7
20DEC88   2   1   0   4   4   6   2   2   1   4
          1  11   .   1   0   6   5   5   4   2
          2   6   6   4   0
21DEC88   2   6   6   2   7   7   5   2   5   0
          9   2   4   2   5   1   4   7   5   6
          5   0   4  36  28
22DEC88   3   7  22   1  11  11  39  46   7  33
         19  21   1   3  43  23   9   0  17  35
         50   0   2   1   0
23DEC88   6  11   8  35  36  19  21   .   .   4
          6  63  35   3  12  34   9   0  46   0
          0  36   3   0  14
24DEC88  13   2  10   4   5  22  21  44  66  13
          8   3   4  27   2  12  17  22  19  36
          9  72   2   4   4
25DEC88   4  33  35   0  11  11  10  28  34   3
         24   6  17   0   8   5   7  19   9   7
         21  17  17   2   6
26DEC88   3   8   8   2   7   7   8   2   5   9
          2   8   2  10  16   9   5  14  15   1
         12   2   2  14  18
;
```

First, the MEANS procedure is used to count the number of cancelled flights for each day. This information is then added to the data set Times.

```
proc means data=Times noprint;
   var Delay;
   by Day ;
```

```
   output out=Cancel nmiss=Ncancel;

data Times;
   merge Times cancel;
   by Day;
run;
```

The following statements create a data set named Weather that contains information about possible causes for delays. This data set is merged with the data set Times.

```
data Weather;
   informat Day date7. ;
   format   Day date7. ;
   length Reason $ 16 ;
   input Day Flight Reason & ;
   datalines;
16DEC88  8   Fog
17DEC88  18  Snow Storm
17DEC88  23  Sleet
21DEC88  24  Rain
21DEC88  25  Rain
22DEC88  7   Mechanical
22DEC88  15  Late Arrival
24DEC88  9   Late Arrival
24DEC88  22  Late Arrival
;

data Times;
   merge Times Weather;
   by Day Flight;
run;
```

Next, control limits are established using the delays prior to the holiday period.

```
   proc shewhart data=Times;
      where Day <= '21DEC88'D;
      boxchart Delay * Day /
         nochart
         outlimits=Timelim;
   run;
```

The OUTLIMITS= option names a data set (Timelim) that saves the control limits. The NOCHART option suppresses the display of the chart.

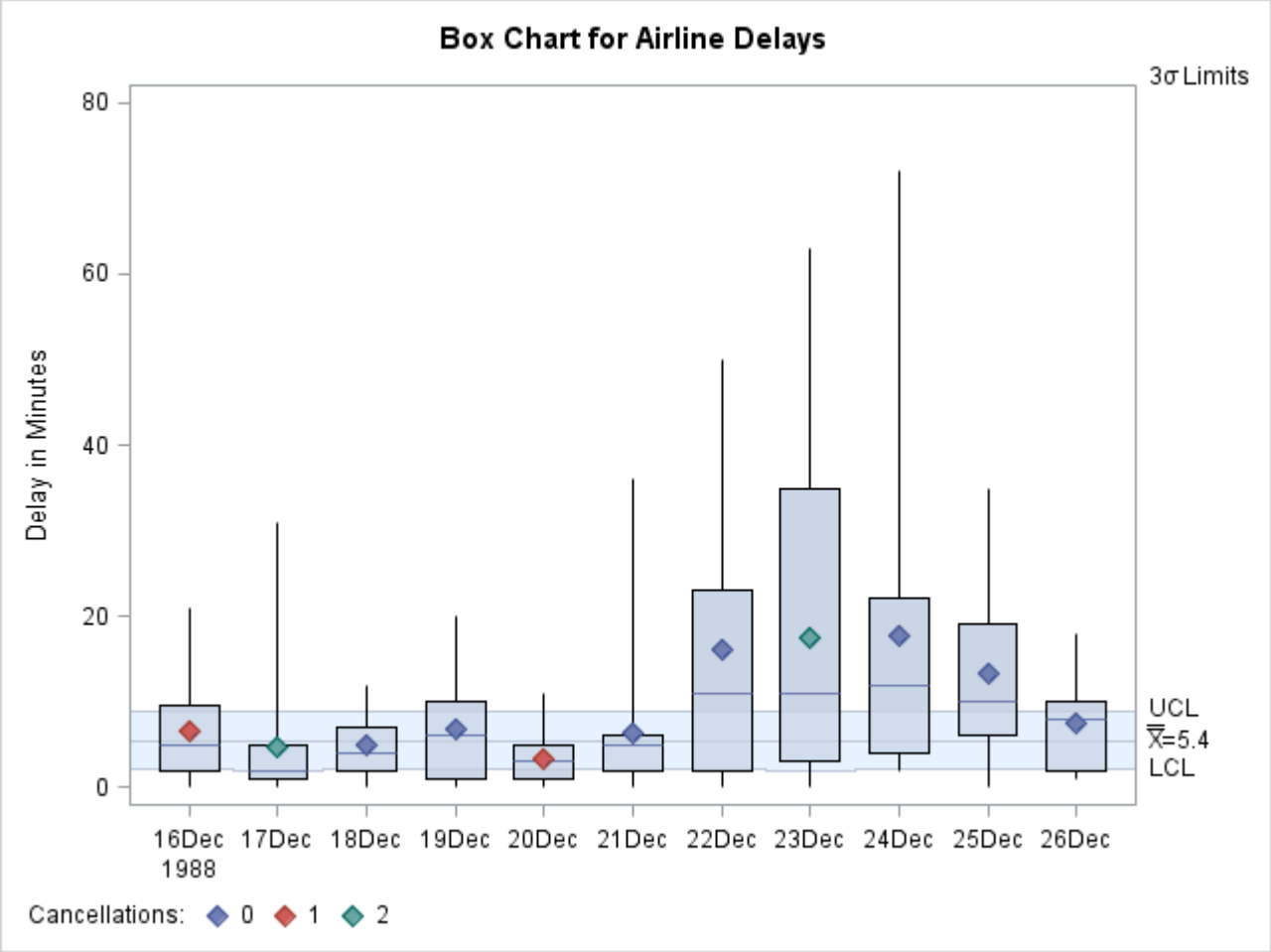
The following statements create a box chart for the complete set of data using the control limits in Timelim:

```
ods graphics on;
title 'Box Chart for Airline Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day = Ncancel /
      readlimits
      nohlabel
      nolegend
      odstitle = title;
```

```
    label Delay   = 'Delay in Minutes'
          Ncancel = 'Cancellations:';
run;
```

The box chart is shown in Output 19.1.1. The level of the *symbol-variable* Ncancel determines the symbol marker for each subgroup mean. The NOHLABEL option suppresses the label for the horizontal axis, and the NOLEGEND option suppresses the default legend for subgroup sample sizes.

Output 19.1.1 Box Chart for Airline Data



The delay distributions from December 22 through December 25 are drastically different from the delay distributions during the pre-holiday period. Both the mean delay and the variability of the delays are much greater during the holiday period.

Example 19.2: Creating Various Styles of Box-and-Whisker Plots

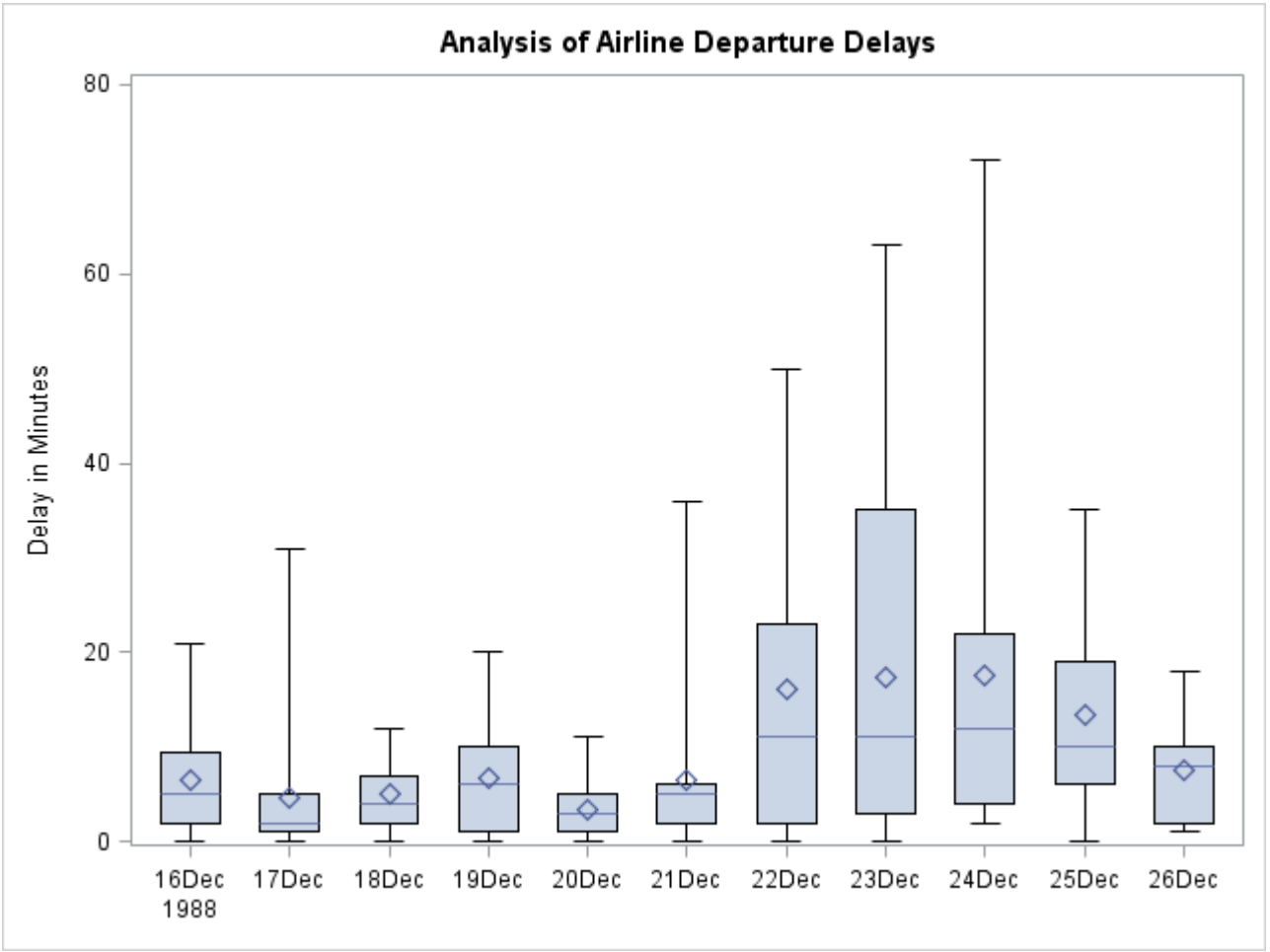
NOTE: See *Creating Various Styles of Box Charts* in the SAS/QC Sample Library.

This example uses the flight delay data of the preceding example to illustrate how you can create box charts with various styles of box-and-whisker plots. For simplicity, the control limits are suppressed. The following statements create a chart, shown in Output 19.2.1, that displays *skeletal box-and-whisker plots*:

```
ods graphics on;
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = skeletal
      serifs
      nolimits
      nohlabel
      nolegend;
   label Delay = 'Delay in Minutes';
run;
```

In a skeletal box-and-whisker plot, the whiskers are drawn from the quartiles to the extreme values of the subgroup sample. You can also request this style by omitting the BOXSTYLE= option, because this style is the default. The SERIFS option adds serifs to the whiskers (by default, serifs are omitted with the skeletal style). The NOLIMITS option suppresses the display of the control limits.

Output 19.2.1 BOXSTYLE=SKELETAL with Serifs

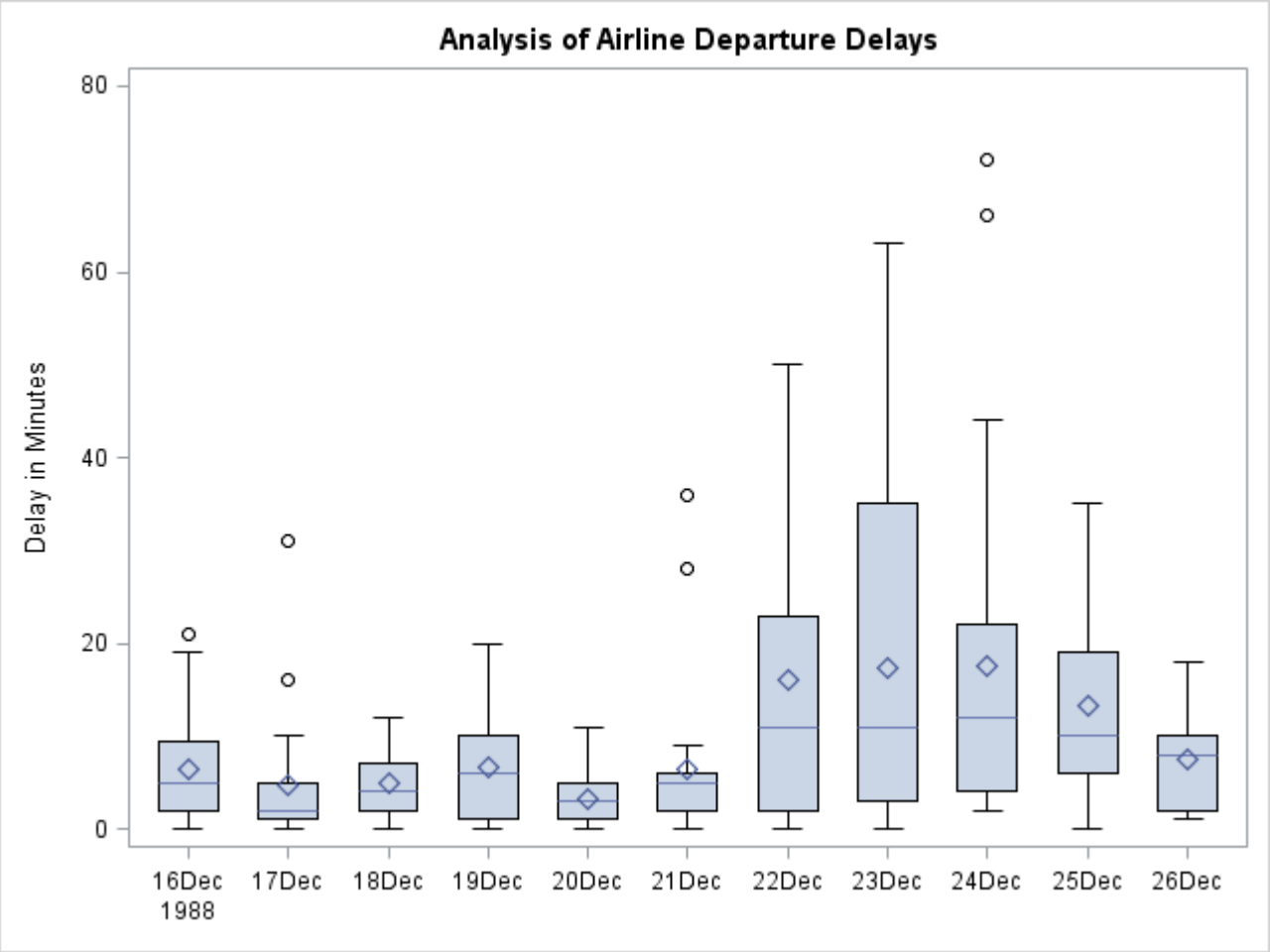


The following statements request a box chart with *schematic box-and-whisker plots*:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematic
      nolimits
      nohlabel
      nolegend;
   label Delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 19.2.2. When BOXSTYLE=SCHEMATIC is specified, the whiskers are drawn to the most extreme points in the subgroup sample that lie within or equal to so-called "fences." The *upper fence* is defined as the third quartile (represented by the upper edge of the box) plus 1.5 times the interquartile range (IQR). The *lower fence* is defined as the first quartile (represented by the lower edge of the box) minus 1.5 times the interquartile range. Observations outside the fences are identified with a special symbol. Serifs are added to the whiskers by default. For further details, see the entry for BOXSTYLE= in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Output 19.2.2 BOXSTYLE=SCHEMATIC

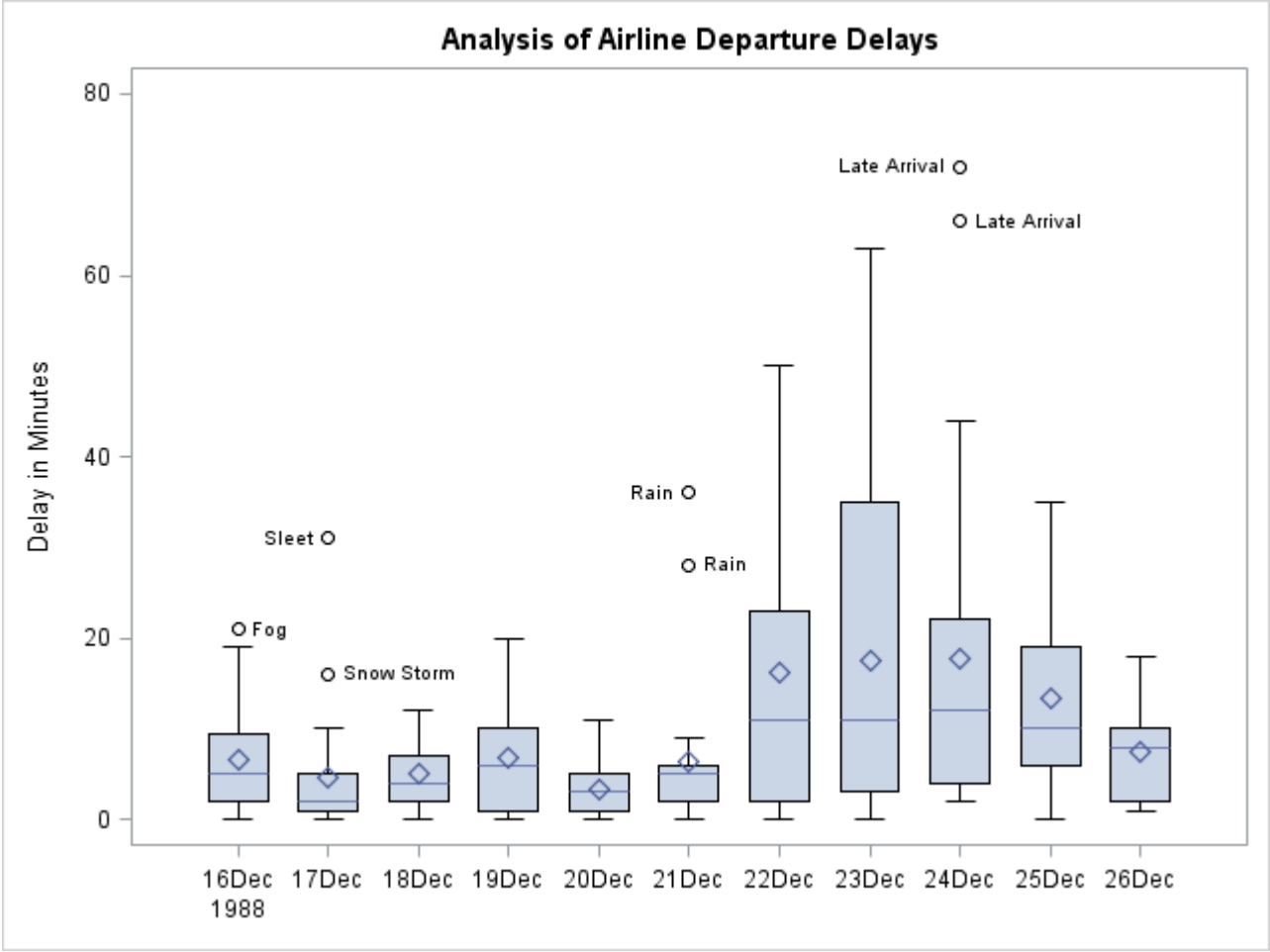


The following statements create a box chart with schematic box-and-whisker plots in which the observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematicid
      llimits  = 20
      nolimits
      nohlabel
      nolegend;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 19.2.3. If you specify BOXSTYLE=SCHEMATICID, schematic box-and-whisker plots are displayed in which the value of the first ID variable (in this case, Reason) is used to label each observation outside the fences.

Output 19.2.3 BOXSTYLE=SCHEMATICID



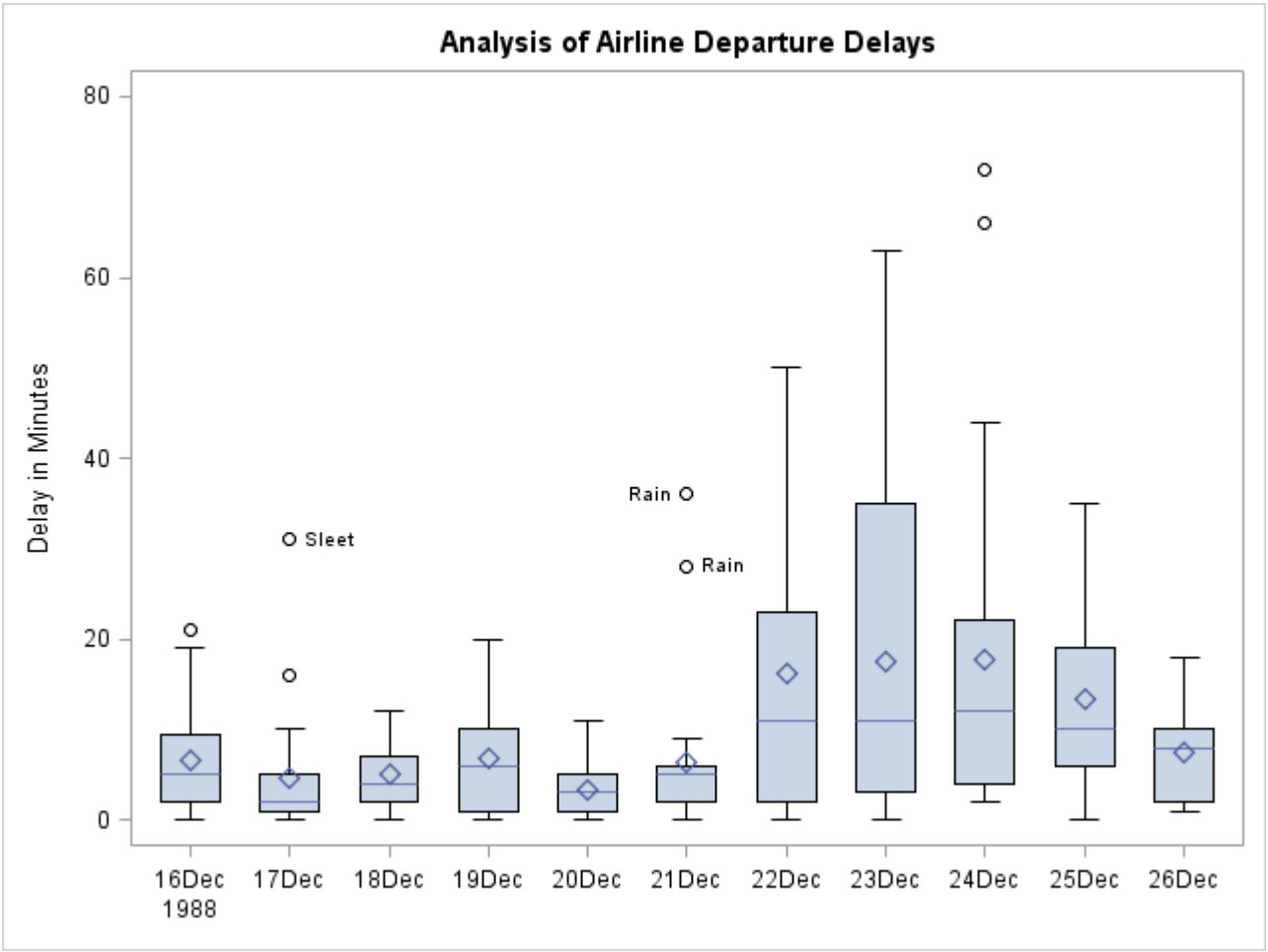
The following statements create a box chart with schematic box-and-whisker plots in which only the extreme

observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematicidfar
      nolimits
      nohlabel
      nolegend;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 19.2.4. If you specify BOXSTYLE=SCHEMATICIDFAR, schematic box-and-whisker plots are displayed in which the value of the first ID variable is used to label each observation outside the *lower* and *upper far fences*. The *lower* and *upper far fences* are located 3×IQR below the 25th percentile and above the 75th percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled.

Output 19.2.4 BOXSTYLE=SCHEMATICIDFAR



Example 19.3: Creating Notched Box-and-Whisker Plots

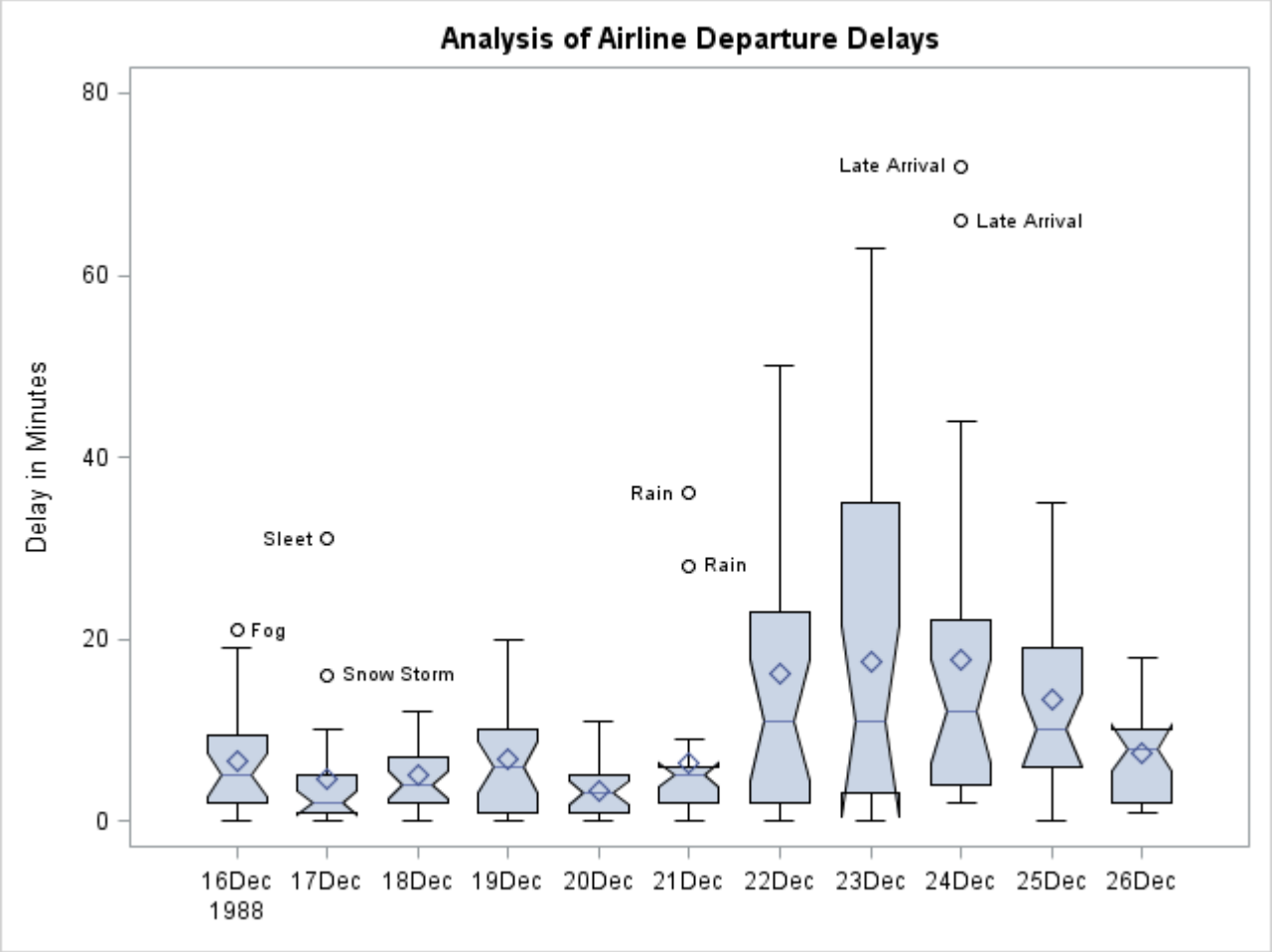
NOTE: See *Using Box Charts to Compare Subgroups* in the SAS/QC Sample Library.

The following statements use the flight delay data of Example 19.1 to illustrate how to create side-by-side box-and-whisker plots with notches:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematicid
      nolimits
      nohlabel
      nolegend
      notches;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

The control limits are suppressed with the NOLIMITS option. The notches, requested with the NOTCHES option, measure the significance of the difference between two medians. The medians are significantly different at approximately the 95% level if the notches do not overlap.

Output 19.3.1 Notched Side-by-Side Box-and-Whisker Plots



Example 19.4: Creating Box-and-Whisker Plots with Varying Widths

NOTE: See *Varying Width Box-and-Whisker Plots* in the SAS/QC Sample Library.

This example shows how to create a box chart with box-and-whisker plots whose widths vary proportionately with the subgroup sample size. The following statements create a SAS data set named Times2 that contains flight departure delays (in minutes) recorded daily for eight consecutive days:

```
data Times2;
   label Delay = 'Delay in Minutes';
   informat Day date7. ;
   format   Day date7. ;
   input Day @ ;
   do Flight=1 to 25;
      input Delay @ ;
      output;
   end;
   datalines;
01MAR90   12  4   2   2  15   8   0  11   0   0
```

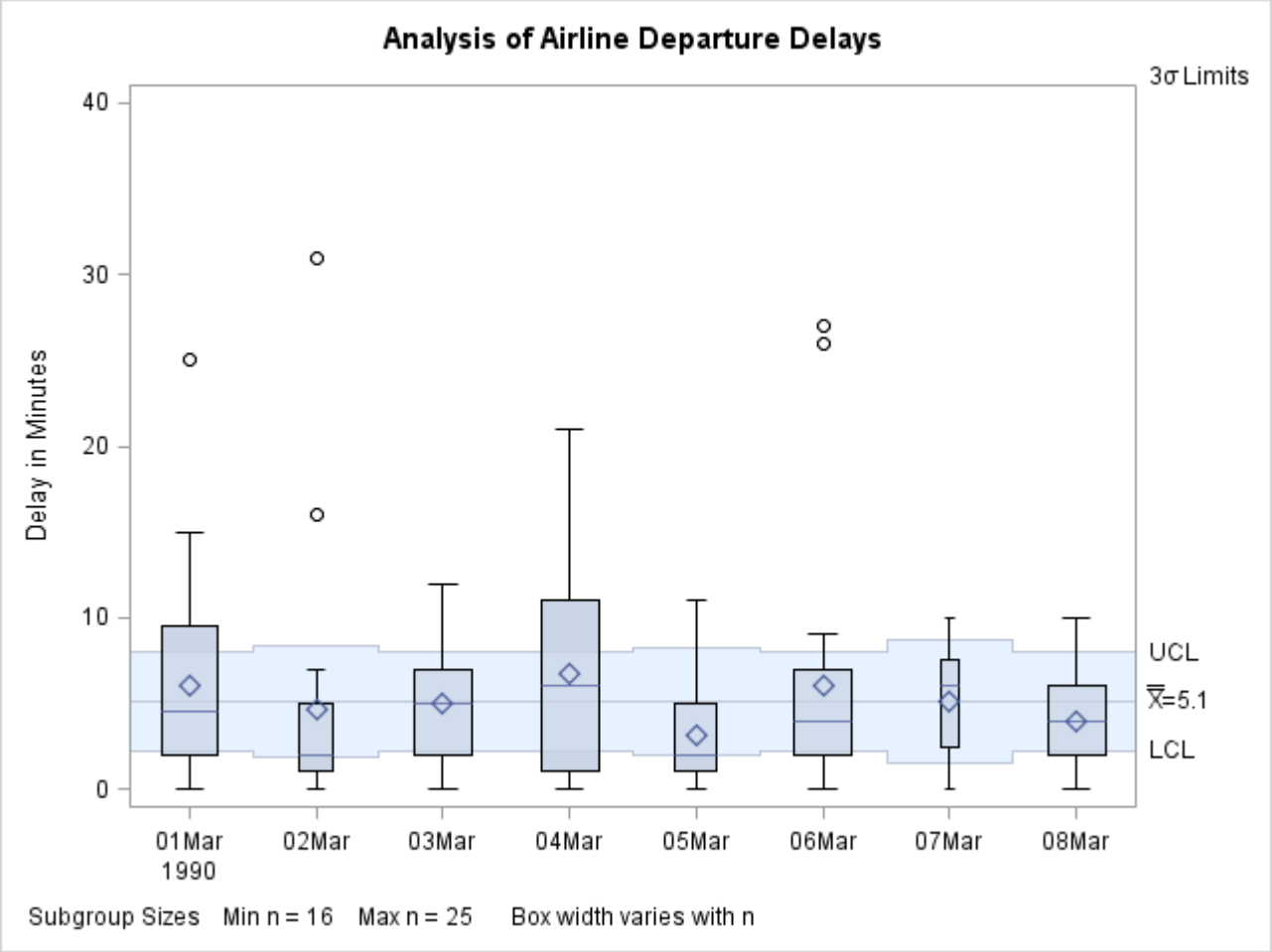
```
            0   12    3    .    2    3    5    0    6   25
            7    4    9    5   10
02MAR90     1    .    3    .    0    1    5    0    .    .
            1    5    7    .    7    2    2   16    2    1
            3    1   31    .    0
03MAR90     6    8    4    2    3    2    7    6   11    3
            2    7    0    1   10    2    5   12    8    6
            2    7    2    4    5
04MAR90    12    6    9    0   15    7    1    1    0    2
            5    6    5   14    7   21    8    1   14    3
           11    0    1   11    7
05MAR90     2    1    0    4    .    6    2    2    1    4
            1   11    .    1    0    .    5    5    .    2
            3    6    6    4    0
06MAR90     8    6    5    2    9    7    4    2    5    1
            2    2    4    2    5    1    3    9    7    8
            1    0    4   26   27
07MAR90     9    6    6    2    7    8    .    .   10    8
            0    2    4    3    .    .    .    7    .    6
            4    0    .    .    .
08MAR90     1    6    6    2    8    8    5    3    5    0
            8    2    4    2    5    1    6    4    5   10
            2    0    4    1    1
;
```

The following statements create the box chart shown in Output 19.4.1:

```
ods graphics on;
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times2;
   boxchart Delay * Day /
      nohlabel
      boxstyle      = schematic
      odstitle      = title
      boxwidthscale = 1 ;
run;
```

The BOXWIDTHSCALE=1 option specifies that the widths of the box-and-whisker plots are to vary proportionately to the subgroup sample size n. This option is useful in situations where the sample size varies widely across subgroups.

Output 19.4.1 Box Chart with Box-and-Whisker Plots of Varying Widths



Example 19.5: Creating Box-and-Whisker Plots with Different Line Styles and Colors

NOTE: See *Varying Width Box-and-Whisker Plots* in the SAS/QC Sample Library.

The control limits in Output 19.4.1 apply to the subgroup means. This example illustrates how you can modify the chart to indicate whether the variability of the process is in control. The following statements create a box chart for Delay in which a dashed outline and a light gray fill color are used for a box-and-whisker plot if the corresponding subgroup standard deviation exceeds its 3σ limits.

First, the SHEWHART procedure is used to create an OUTTABLE= data set (Delaytab) that contains a variable (_EXLIMS_) that records which standard deviations exceed their 3σ limits.

```
proc shewhart data=Times2;
   xschart Delay * Day / nochart
                         outtable = Delaytab;
run;
```

Then, this information is used to set the line styles and fill colors as follows:

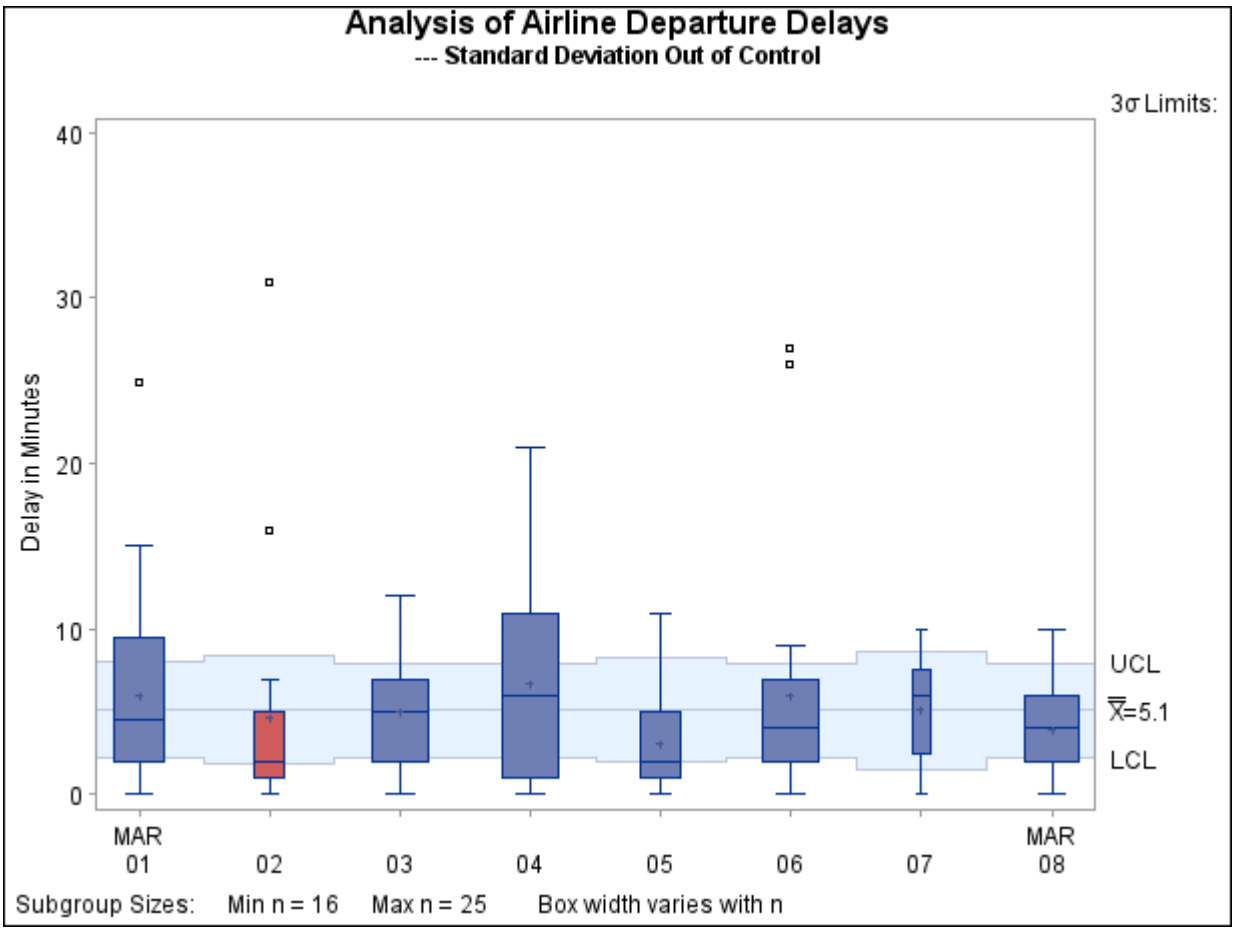
```
data Delaytab;
   length Boxcolor $ 8 ;
   set Delaytab;
   keep Day Boxcolor;
   if _exlims_ = 'UPPER' or _exlims_ = 'LOWER' then do;
      Boxcolor = 'Outside' ;
   end;
   else do;
      Boxcolor = 'Inside' ;
   end;
run;

data Times2;
   merge Times2 Delaytab;
   by Day;
run;
```

The following statements create the modified box chart:

```
ods graphics off;
title  'Analysis of Airline Departure Delays' ;
title2 '--- Standard Deviation Out of Control';
proc shewhart data=Times2;
   boxchart Delay * Day /
        nohlabel
        boxstyle      = schematic
        boxfill       = ( Boxcolor )
        boxwidthscale = 1
        odstitle      = title;
run;
```

The chart is shown in Output 19.5.1. The values of the variable Boxcolor specified with the BOXFILL= option determine the fill colors. The chart indicates that the large variability for March 2 should be checked.

Output 19.5.1 Box Chart Displaying Out-of-Control Subgroup Standard Deviations



Example 19.6: Computing the Control Limits for Subgroup Maximums

NOTE: See *Control Chart for the Subgroup Maximum* in the SAS/QC Sample Library.

This example illustrates how to compute and display control limits for the *maximum* of a subgroup sample. Subgroup samples of 20 metal braces are collected daily, and the lengths of the braces are measured in centimeters. These data are analyzed extensively in Example 19.43. The box chart for LogLength (the log of length) shown in Output 19.43.3 indicates that the subgroup mean is in control and that the subgroup distributions of LogLength are approximately normal. The following statements save the control limits for the mean of the LogLength in a data set named Logllims:

```
data LengthData;
   set LengthData;
   LogLength=log(Length-105);
run;

proc shewhart data=LengthData;
   xchart LogLength*Day /
      nochart
      outlimits=Logllims;
run;
```

The next statements replace the control limits for the mean of LogLength with control limits for the maximum of LogLength:

```
data Maxlim;
   set LengthData;
   set Logllims;
   drop expmax stdmax;
   label _lclx_ = 'Lower Limit for Maximum of 20'
         _uclx_ = 'Upper Limit for Maximum of 20'
         _mean_ = 'Central Line for Maximum of 20';
   expmax = _stddev_*1.86748 + _mean_;
   stdmax = _stddev_*0.52507;
   _lclx_ = expmax - _sigmas_*stdmax;
   _uclx_ = expmax + _sigmas_*stdmax;
   _mean_ = expmax;
   call symput('avgmax',left(put(expmax,8.1)));
run;
```

The control limits are computed using the fact that the maximum of a sample of size 20 from a normal population with zero mean and unit standard deviation has an expected value of 1.86747 and a standard deviation of 0.52509; refer to Teichroew (1962) and see Table 19.13. Finally, the following statements create a box chart for LogLength that displays control limits for the subgroup maximum:

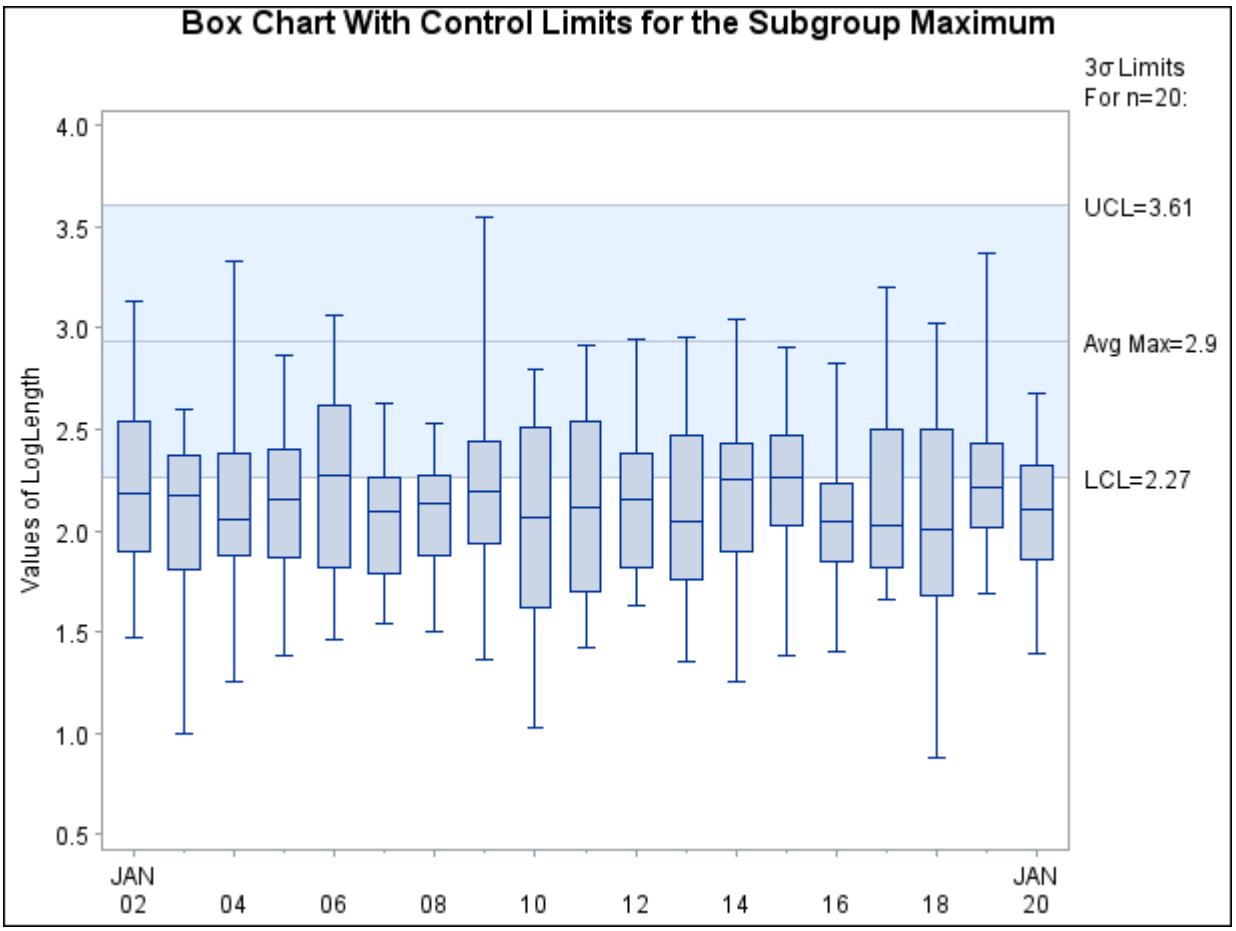
```
ods graphics off;
title 'Box Chart With Control Limits for the Subgroup Maximum';
symbol v=none;
proc shewhart data=LengthData limits=Maxlim;
   boxchart LogLength*Day /
      ranges
      serifs
      nohlabel
      nolegend
      xsymbol="Avg Max=&AVGMAX" ;
   label LogLength='Values of LogLength';
run;
```

The box chart, shown in Output 19.6.1, indicates that the maximum is in control because the tips of the upper whiskers fall within the control limits.

The SYMPUT call is used to pass the value of _MEAN_ in a macro variable to the SHEWHART procedure so that this value can be used to label the central line.

You can apply the variable replacement method shown here to data with sample sizes other than 20 by replacing the constants 1.86747 and 0.52509 with the appropriate values from Table 19.13. Austin (1973) describes a method for approximating these values. You can also use the preceding statements to display control limits for the subgroup minimum by changing the sign of the expected values in Table 19.13.

Output 19.6.1 Box Chart for Subgroup Maximum



The variable replacement method can also be used to create a variety of box charts, including the modifications suggested by Iglewicz and Hoaglin (1987) and Rocke (1989).

Table 19.13 Expected Values and Standard Deviations of Maximum of a Normal Sample

n	Expected Value	Standard Deviation
2	0.56418	0.82565
3	0.84628	0.74798
4	1.02937	0.70123
5	1.16296	0.66899
6	1.26720	0.64494
7	1.35217	0.62605
8	1.42360	0.61065
9	1.48501	0.59780
10	1.53875	0.58681
11	1.58643	0.57730
12	1.62922	0.56891
13	1.66799	0.56144

Table 19.13 *continued*

n	Expected Value	Standard Deviation
14	1.70338	0.55474
15	1.73591	0.54869
16	1.76599	0.54316
17	1.79394	0.53809
18	1.82003	0.53342
19	1.84448	0.52910
20	1.86747	0.52509

Example 19.7: Constructing Multi-Vari Charts

"Multi-vari" charts[3] are used in a variety of industries to analyze process data with nested (hierarchical) patterns of variation

- within-sample variation (for example, position within wafer)
- sample-to-sample variation within batches of samples (for example, wafer within lot)
- batch-to-batch variation (for example, across lots)

This example illustrates the construction of a "multi-vari" display. The following statements create a SAS data set named Parm that contains the value of a measured parameter (Measure) recorded at each of five positions on wafers produced in lots.

```
data Parm;
   length _phase_ $ 5 Wafer $ 2 Position $ 1;
   input  _phase_ $  &  Wafer $ &  Position $  Measure ;
   datalines;
   Lot A      01         L        2.42435
   Lot A      01         B        2.44150
   Lot A      01         C        2.42143
   Lot A      01         T        2.44960
   Lot A      01         R        2.50050
   Lot A      02         L        2.68188
   Lot A      02         B        2.57195
   Lot A      02         C        2.54678
   Lot A      02         T        2.65978
   Lot A      02         R        2.69208
   Lot A      03         L        2.18005
   Lot A      03         B        2.13593
   Lot A      03         C        2.44303
   Lot A      03         T        2.29052
   Lot A      03         R        2.25963
   Lot B      01         L        2.46573
   Lot B      01         B        2.44898
```

[3] Multi-vari charts should not be confused with multivariate control charts.

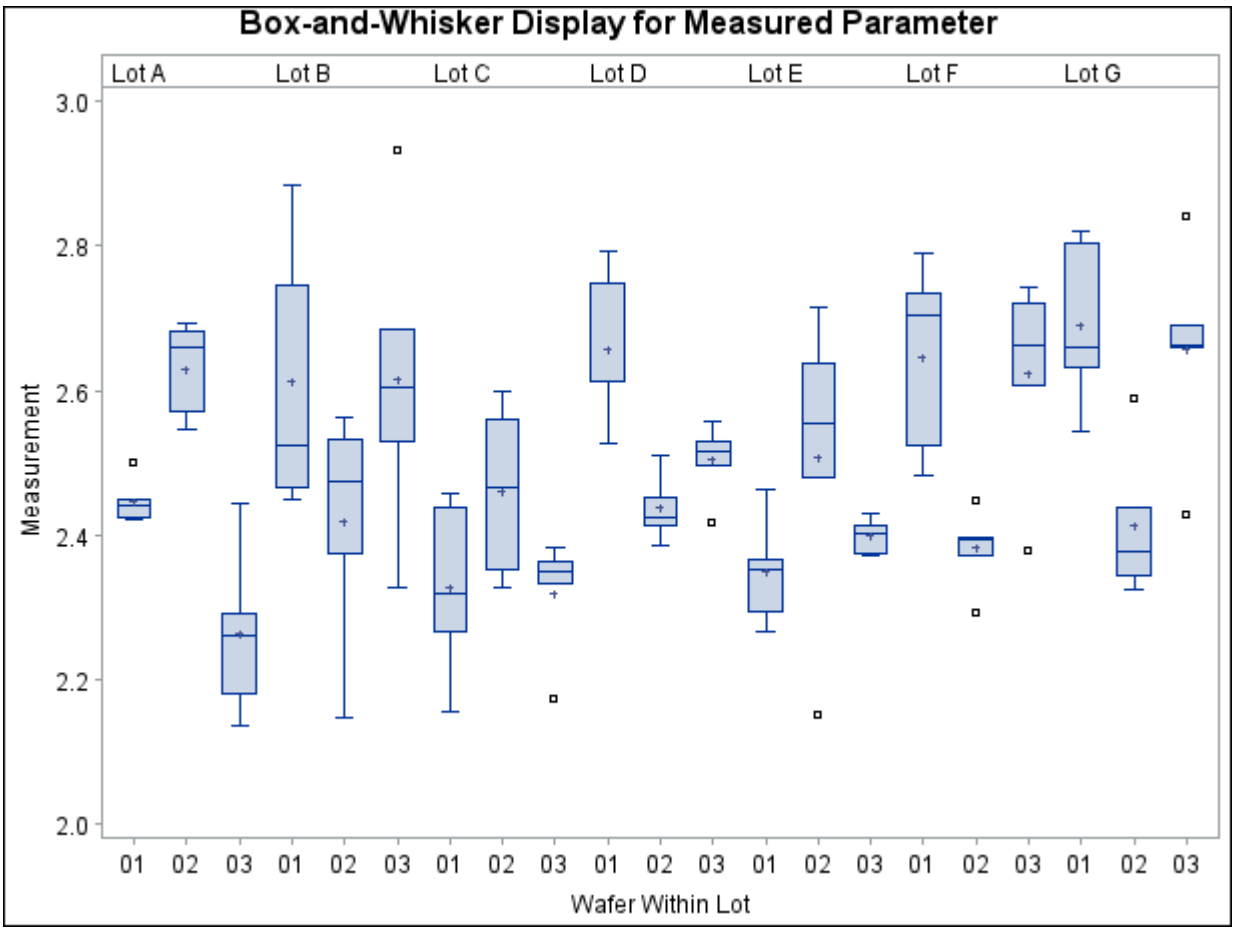
```
   Lot B      01         C        2.52365

   ... more lines ...

   Lot G      03         C        2.66303
   Lot G      03         T        2.65913
   Lot G      03         R        2.84378
;
```

The following statements create an ordinary side-by-side box-and-whisker display for the measurements.

```
ods graphics off;
title 'Box-and-Whisker Display for Measured Parameter';
proc shewhart data=Parm;
   boxchart Measure*Wafer /
         nolimits
         boxstyle  = schematic
         idsymbol  = square
         readphase = all
         phaselegend
         nolegend;
label Measure = 'Measurement'
      Wafer   = 'Wafer Within Lot';
run;
```

The display is shown in Output 19.7.1. Here, the *subgroup-variable* is Wafer, and the option BOXSTYLE=SCHEMATIC is specified to request schematic box-and-whisker plots for the measurements in each subgroup (wafer) sample. The lot values are provided as the values of the special variable _PHASE_, which is read when the option READPHASE=ALL is specified. The option PHASELEGEND requests the legend for phase (lot) values at the top of the chart, and the NOLEGEND option suppresses the default legend for sample sizes. The NOLIMITS option suppresses the display of control limits. This option is recommended whenever you are using the BOXCHART statement to create side-by-side box-and-whisker plots.

Output 19.7.1 Box-and-Whisker Plot Using BOXSTYLE=SCHEMATIC

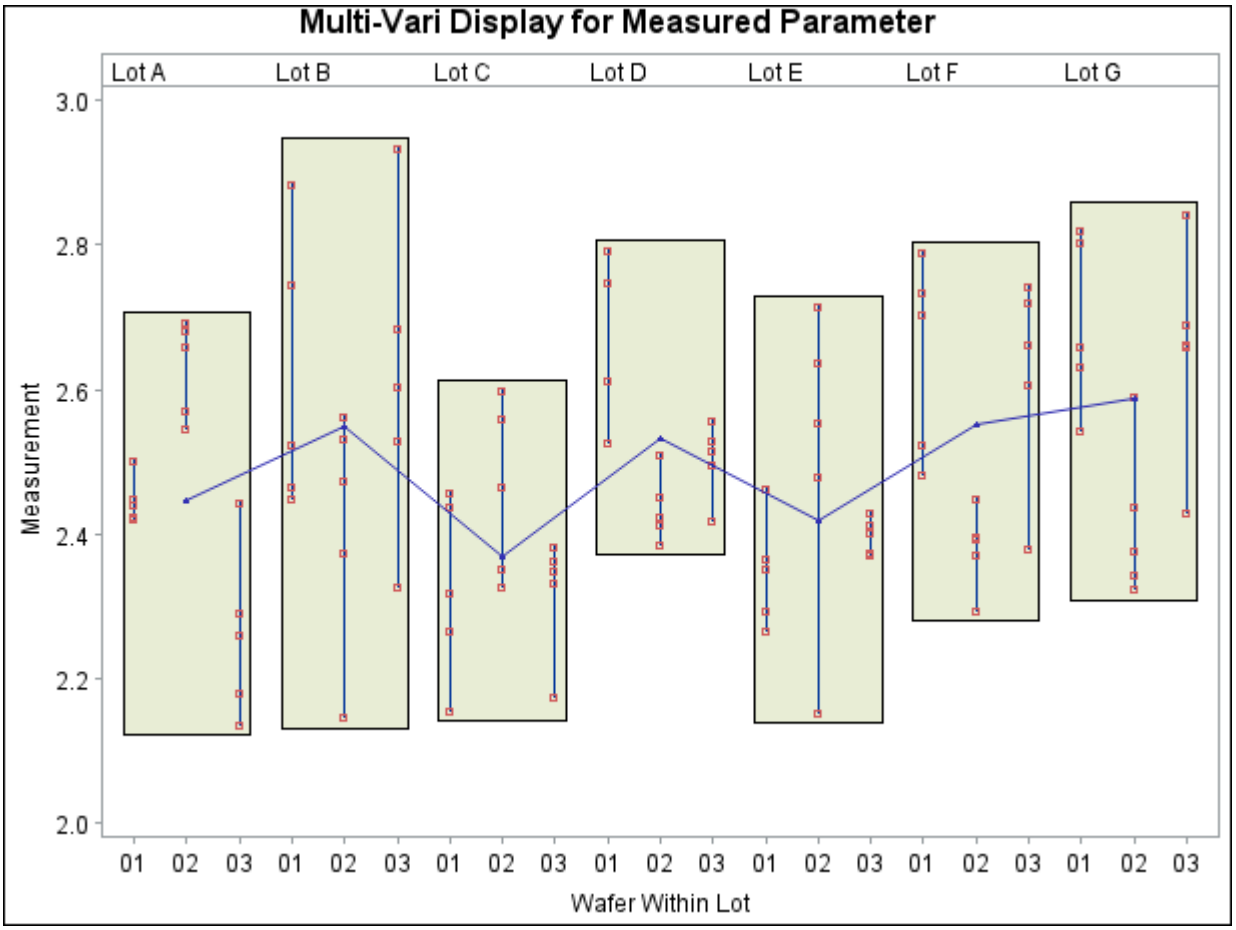


The box-and-whisker display in Output 19.7.1 is not particularly appropriate for these data because there are only five measurements in each wafer and because the variation within each wafer might depend on the position, which is not indicated. The next statements use the BOXCHART statement to produce a multi-vari chart for the same data.

```
symbol v=none;
title 'Multi-Vari Display for Measured Parameter';
proc shewhart data=Parm;
   boxchart Measure*Wafer /
      nolimits
      boxstyle          = pointsjoin
      idsymbol          = square
      cphaseboxfill     = ywh
      cphasebox         = black
      cphasemeanconnect = bib
      phasemeansymbol   = dot
      readphase         = all
      phaselegend
      nolegend;
label Measure = 'Measurement'
      Wafer   = 'Wafer Within Lot';
run;
```

The display is shown in Output 19.7.2.

Output 19.7.2 Multi-Vari Chart Using BOXSTYLE=POINTSJOIN



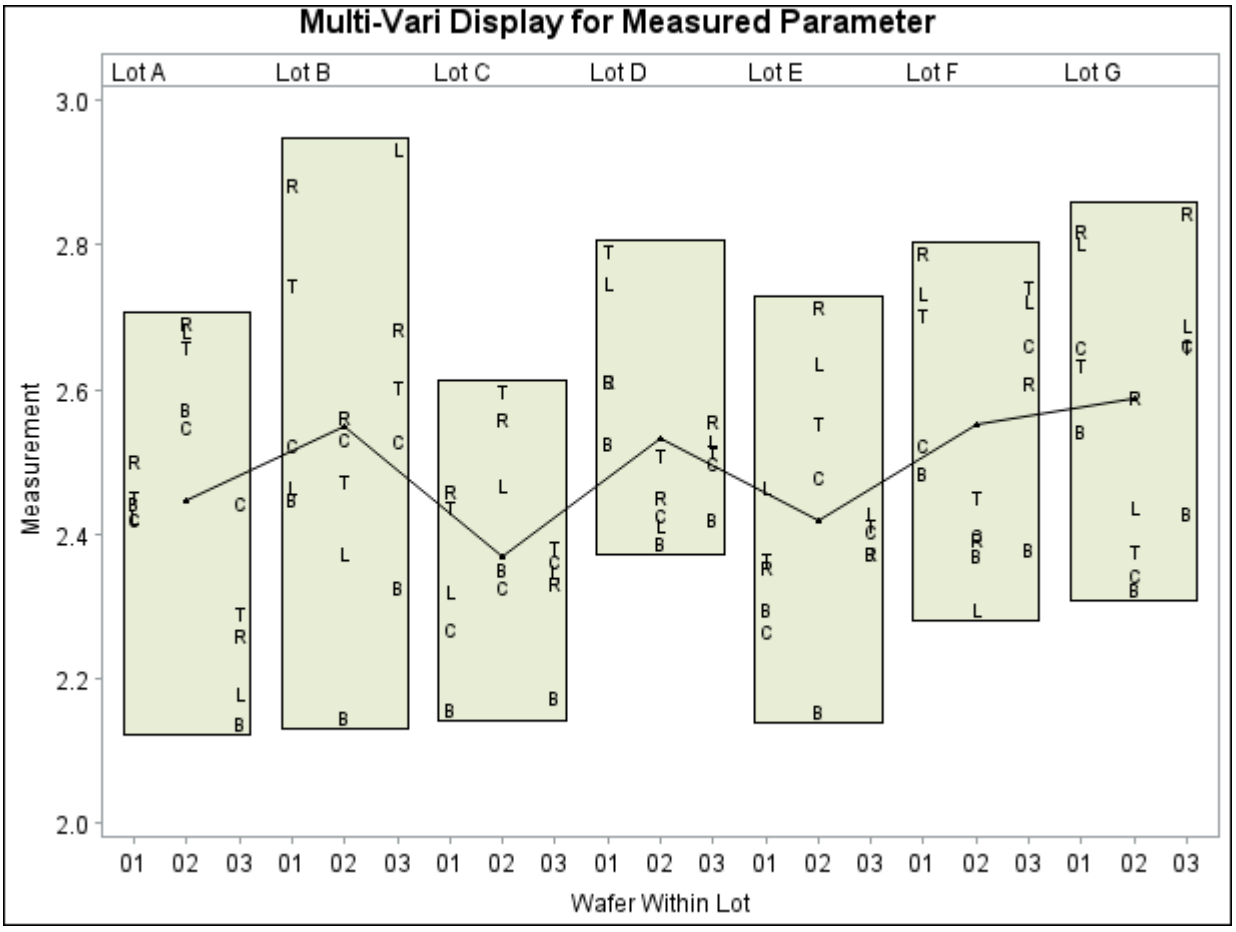
The option BOXSTYLE=POINTSJOIN specifies that the values for each wafer are to be displayed as points joined by a vertical line. The IDSYMBOL= option specifies the symbol marker for the points. The option V=NONE in the SYMBOL statement is specified to suppress the symbol for the wafer averages shown in Output 19.7.1. The option CPHASEBOX=BLACK specifies that the points for each lot are to be enclosed in a black box, and the CPHASEBOXFILL= option specifies the fill color for the box. The option CPHASEMEANCONNECT=BLACK specifies that the means of the lots are to be connected with black lines, and the PHASEMEANSYMBOL= option specifies the symbol marker for the lot means.

The following statements create a slightly different multi-vari chart using the values of the variable Position to identify the measurements for each wafer. Note that the option BOXSTYLE=POINTSID is specified and that Position is specified as the ID variable. The display is shown in Output 19.7.3.

```
symbol v=none;
title 'Multi-Vari Display for Measured Parameter';
proc shewhart data=Parm;
   boxchart Measure*Wafer /
      nolimits
      cphaseboxfill      = ywh
      cphasemeanconnect = black
```

```
      boxstyle          = pointsid
      phasemeansymbol   = dot
      readphase         = all
      phaselegend
      nolegend;
   label Measure = 'Measurement'
         Wafer   = 'Wafer Within Lot';
   id Position;
run;
```

Output 19.7.3 Multi-Vari Chart Using BOXSTYLE=POINTSID



CCHART Statement: SHEWHART Procedure

Overview: CCHART Statement

The CCHART statement creates *c* charts for the numbers of nonconformities (defects) in subgroup samples.

You can use options in the CCHART statement to

- specify the number of inspection units per subgroup. Typically (but not necessarily), each subgroup consists of a single unit.
- compute control limits from the data based on a multiple of the standard error of the counts or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control the layout and appearance of the chart

You have three alternatives for producing *c* charts with the CCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, “SAS/QC Graphics,” for more information about producing these different kinds of graphs.

Getting Started: CCHART Statement

This section introduces the CCHART statement with simple examples that illustrate commonly used options. Complete syntax for the CCHART statement is presented in the section "Syntax: CCHART Statement" on page 1492, and advanced examples are given in the section "Examples: CCHART Statement" on page 1511.

Creating c Charts from Defect Count Data

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

A *c* chart is used to monitor the number of paint defects on new trucks. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named Trucks, which contains the defect counts:

```
data Trucks;
   input TruckID $ Defects @@;
   label TruckID='Truck Identification Number'
         Defects='Number of Paint Defects';
   datalines;
C1   5   C2   4   C3   4   C4   8   C5  7
C6  12   C7   3   C8  11   E4   8   E9  4
E7   9   E6  13   A3   5   A4   4   A7  9
Q1  15   Q2   8   Q3   9   Q9  10   Q4  8
;
```

A partial listing of Trucks is shown in Figure 19.15.

Figure 19.15 The Data Set Trucks

Paint Defects on New Trucks

TruckID	Defects
C1	5
C2	4
C3	4
C4	8
C5	7

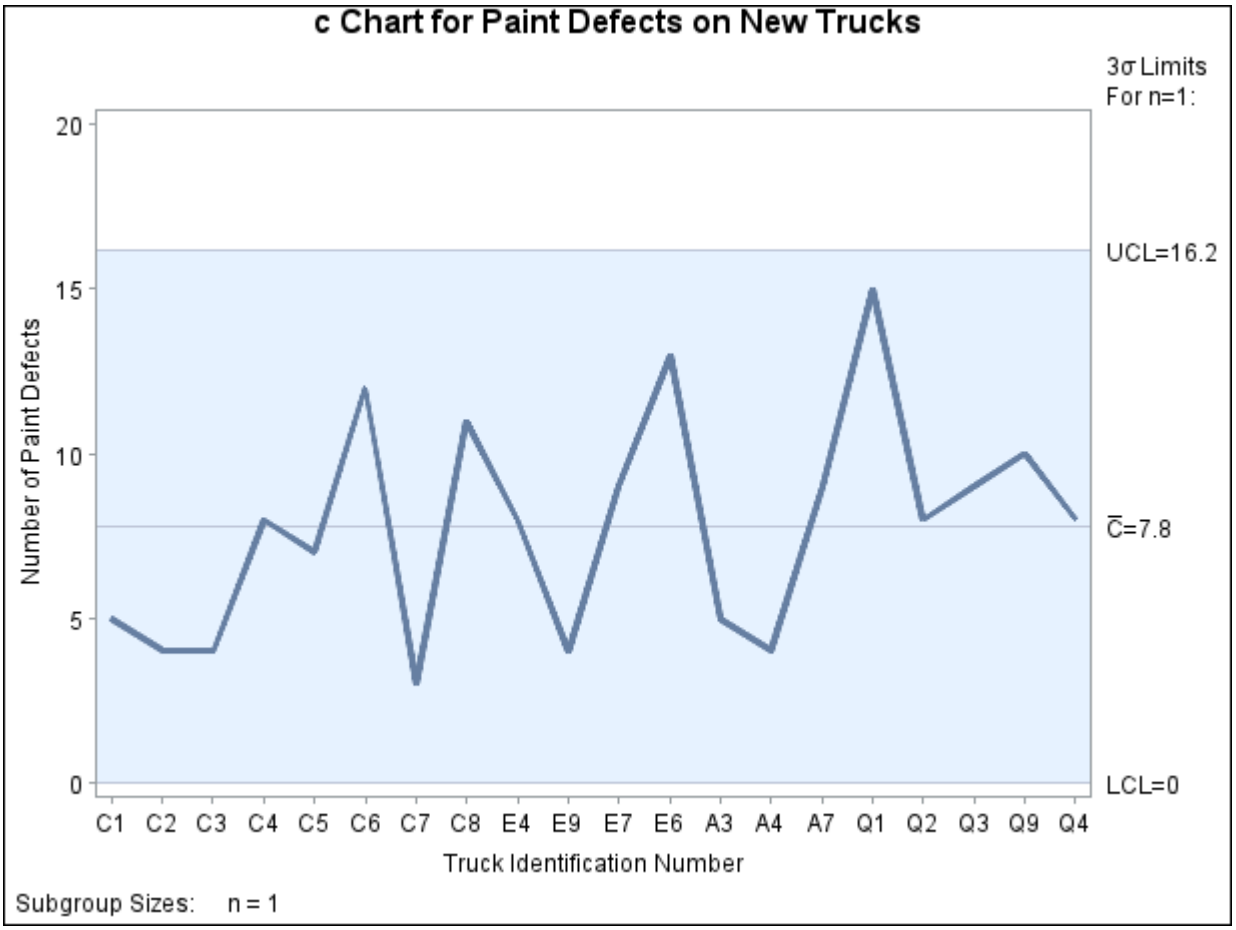
There is a single observation per truck. The variable TruckID identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Defects contains the number of nonconformities in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *c* chart shown in Figure 19.16:

```
ods graphics off;
title 'c Chart for Paint Defects on New Trucks';
proc shewhart data=Trucks;
   cchart Defects*TruckID;
run;
```

This example illustrates the basic form of the CCHART statement. After the keyword CCHART, you specify the *process* to analyze (in this case, Defects) followed by an asterisk and the *subgroup-variable* (TruckID).

Figure 19.16 *c* Chart of Paint Defects (Traditional Graphics)



Each point on the *c* chart represents the number of nonconformities for a particular subgroup. For instance, the value plotted for the first subgroup is 5 (because there are five paint defects on the first truck). By default, the control limits shown are 3σ limits estimated from the data; the formulas are given in "Control Limits" on page 1504. Because none of the points exceed the 3σ limits, the *c* chart indicates that the painting process is in statistical control.

See "Constructing Charts for Numbers of Nonconformities (c Charts)" on page 1503 for details concerning *c* charts. For more details on reading raw data, see "DATA= Data Set" on page 1508.

Saving Control Limits

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a *c* chart in a SAS data set; this enables you to apply the control limits to future data (see the section "Reading Preestablished Control Limits" on page 1487) or subsequently modify the limits with a DATA step program.

The following statements read the data set Trucks introduced in "Creating c Charts from Defect Count Data" on page 1483 and saves the control limit information displayed in Figure 19.16 in a data set named Deflim:

```
proc shewhart data=Trucks;
   cchart Defects*TruckID / outlimits=Deflim
                            nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. Options such as OUTLIMITS= and NOCHART are specified after the slash (/) in the CCHART statement. A complete list of options is presented in the section "Syntax: CCHART Statement" on page 1492. The data set Deflim is listed in Figure 19.17.

Figure 19.17 The Data Set Deflim Containing Control Limit Information

Control Limits Data Set Deflim

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_U_	_LCLC_	_C_	_UCLC_
Defects	TruckID	ESTIMATE	1	.002492887	3	7.8	0	7.8	16.1785

The data set Deflim contains one observation with the limits for the *process* Defects. The variables _LCLC_, and _UCLC_ contain the lower and upper control limits. The variable _C_ contains the central line, and the variable _U_ contains the average number of nonconformities per inspection unit. Because all the subgroups contain a single inspection unit, the values of _C_ and _U_ are the same. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see the section "OUTLIMITS= Data Set" on page 1506.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
title 'Number of Nonconformities and Control Limit Information';
proc shewhart data=Trucks;
   cchart Defects*TruckID / outtable=Trucktab
                            nochart;
run;
```

The OUTTABLE= data set Trucktab is listed in Figure 19.18.

Figure 19.18 The Data Set Trucktab

Number of Nonconformities and Control Limit Information

VAR	TruckID	_SIGMAS_	_LIMITN_	_SUBN_	_LCLC_	_SUBC_	_C_	_UCLC_	_EXLIM_
Defects	C1	3	1	1	0	5	7.8	16.1785	
Defects	C2	3	1	1	0	4	7.8	16.1785	
Defects	C3	3	1	1	0	4	7.8	16.1785	
Defects	C4	3	1	1	0	8	7.8	16.1785	
Defects	C5	3	1	1	0	7	7.8	16.1785	
Defects	C6	3	1	1	0	12	7.8	16.1785	
Defects	C7	3	1	1	0	3	7.8	16.1785	
Defects	C8	3	1	1	0	11	7.8	16.1785	
Defects	E4	3	1	1	0	8	7.8	16.1785	
Defects	E9	3	1	1	0	4	7.8	16.1785	
Defects	E7	3	1	1	0	9	7.8	16.1785	
Defects	E6	3	1	1	0	13	7.8	16.1785	
Defects	A3	3	1	1	0	5	7.8	16.1785	
Defects	A4	3	1	1	0	4	7.8	16.1785	
Defects	A7	3	1	1	0	9	7.8	16.1785	
Defects	Q1	3	1	1	0	15	7.8	16.1785	
Defects	Q2	3	1	1	0	8	7.8	16.1785	
Defects	Q3	3	1	1	0	9	7.8	16.1785	
Defects	Q9	3	1	1	0	10	7.8	16.1785	
Defects	Q4	3	1	1	0	8	7.8	16.1785	

This data set contains one observation for each subgroup sample. The variables _SUBC_ and _SUBN_ contain the number of nonconformities per subgroup and the number of inspection units per subgroup. The variables _LCLC_ and _UCLC_ contain the lower and upper control limits, and the variable _C_ contains the central line. The variables _VAR_ and TruckID contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1507.

An OUTTABLE= data set can be read later as a TABLE= data set in the SHEWHART procedure. For example, the following statements read Trucktab and display a *c* chart (not shown here) identical to the chart in Figure 19.16:

```
title 'c Chart for Paint Defects in New Trucks';
proc shewhart table=Trucktab;
   cchart Defects*Truckid;
   label _SUBC_ = 'Number of Paint Defects';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1510.

Reading Preestablished Control Limits

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

In the previous example, control limits were saved in a SAS data set named Deflim. This example shows how these limits can be applied to defect data for a second group of trucks, which are provided in the following data set:

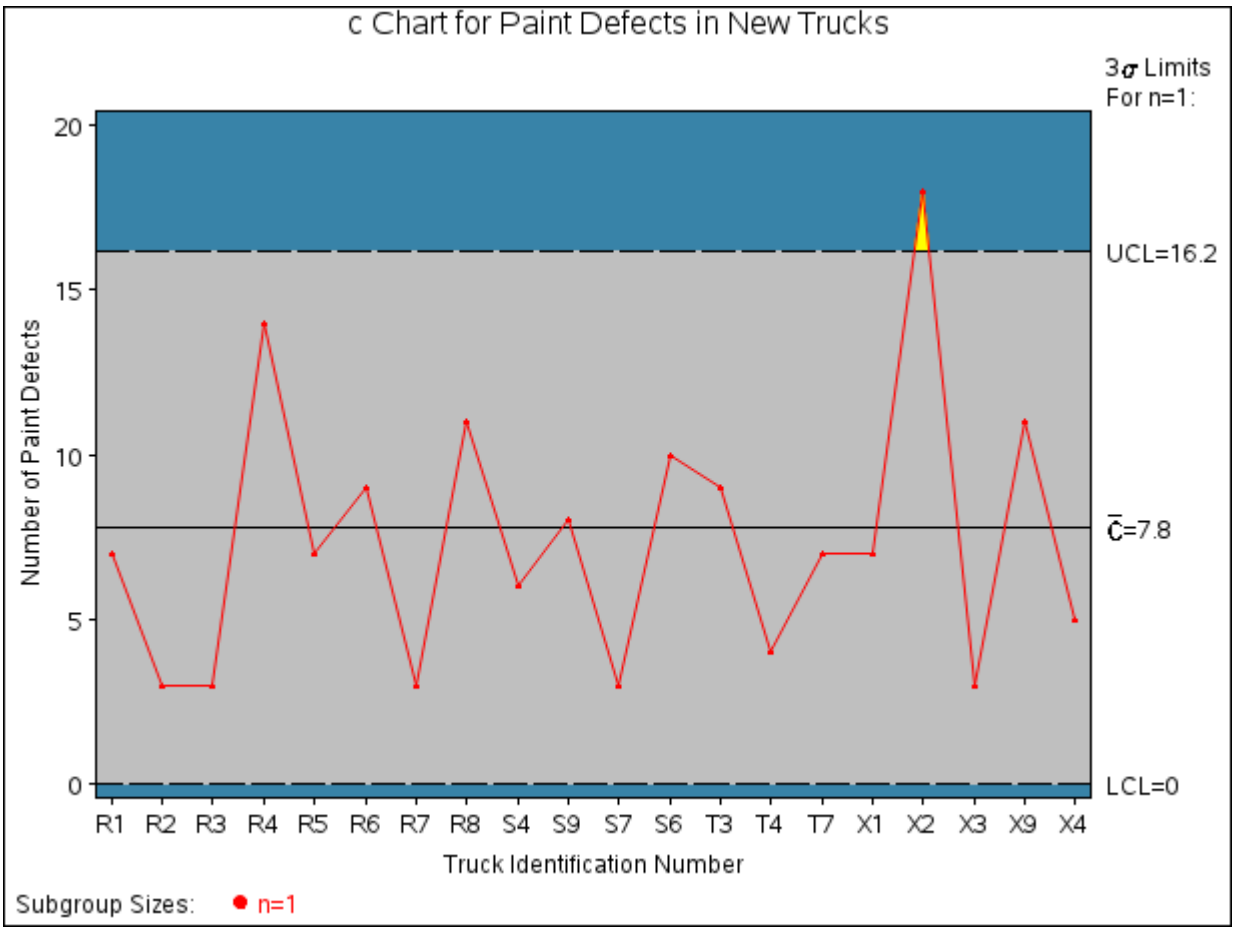
```
data Trucks2;
   input TruckID $ Defects @@;
   label TruckID='Truck Identification Number'
         Defects='Number of Paint Defects';
   datalines;
R1  7   R2   3   R3   3   R4  14   R5  7
R6  9   R7   3   R8  11   S4   6   S9  8
S7  3   S6  10   T3   9   T4   4   T7  7
X1  7   X2  18   X3   3   X9  11   X4  5
;
```

The following statements plot the number of paint defects for the second group of trucks on a *c* chart using the control limits in Deflim. The chart is shown in Figure 19.19.

```
options nogstyle;
goptions ftext='albany amt';
symbol v=dot color=red height=.8;
title 'c Chart for Paint Defects in New Trucks';
proc shewhart data=Trucks2 limits=Deflim;
   cchart Defects*TruckID / cframe   = steel
                            cconnect = red
                            cinfill  = ligr
                            coutfill = yellow ;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and CCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently.

Figure 19.19 *c* Chart for Second Set of Trucks (Traditional Graphics with NOGSTYLE)



Note that the number of defects on the truck with identification number X2 exceeds the upper control limit, indicating that the process is out-of-control. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Defects
- the value of _SUBGRP_ matches the *subgroup-variable* name TruckID

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1509 for details concerning the variables that you must provide.

Creating c Charts from Nonconformities per Unit

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

In the previous example, the input data set provided the number of nonconformities per subgroup sample. However, in some applications, as illustrated here, the data might be provided as the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Because the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in each box is divided by ten and then recorded. The following statements create a SAS data set named Shirts, which contains the average number of flaws per shirt for 25 boxes:

```
data Shirts;
   input Box avgdefu @@;
   avgdefn=10;
   datalines;
 1  0.4   2  0.7   3  0.5   4  1.0   5  0.3
 6  0.2   7  0.0   8  0.4   9  0.4  10  0.6
11  0.2  12  0.7  13  0.3  14  0.1  15  0.3
16  0.6  17  0.6  18  0.3  19  0.7  20  0.3
21  0.0  22  0.1  23  0.5  24  0.6  25  0.4
;
```

A partial listing of Shirts is shown in Figure 19.20.

Figure 19.20 The Data Set Shirts

Average Number of Shirt Flaws

Box	avgdefu	avgdefn
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10

The data set Shirts contains three variables: the box number (Box), the average number of flaws per shirt (avgdefu), and the number of shirts per box (avgdefn). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup consists of ten inspection units.

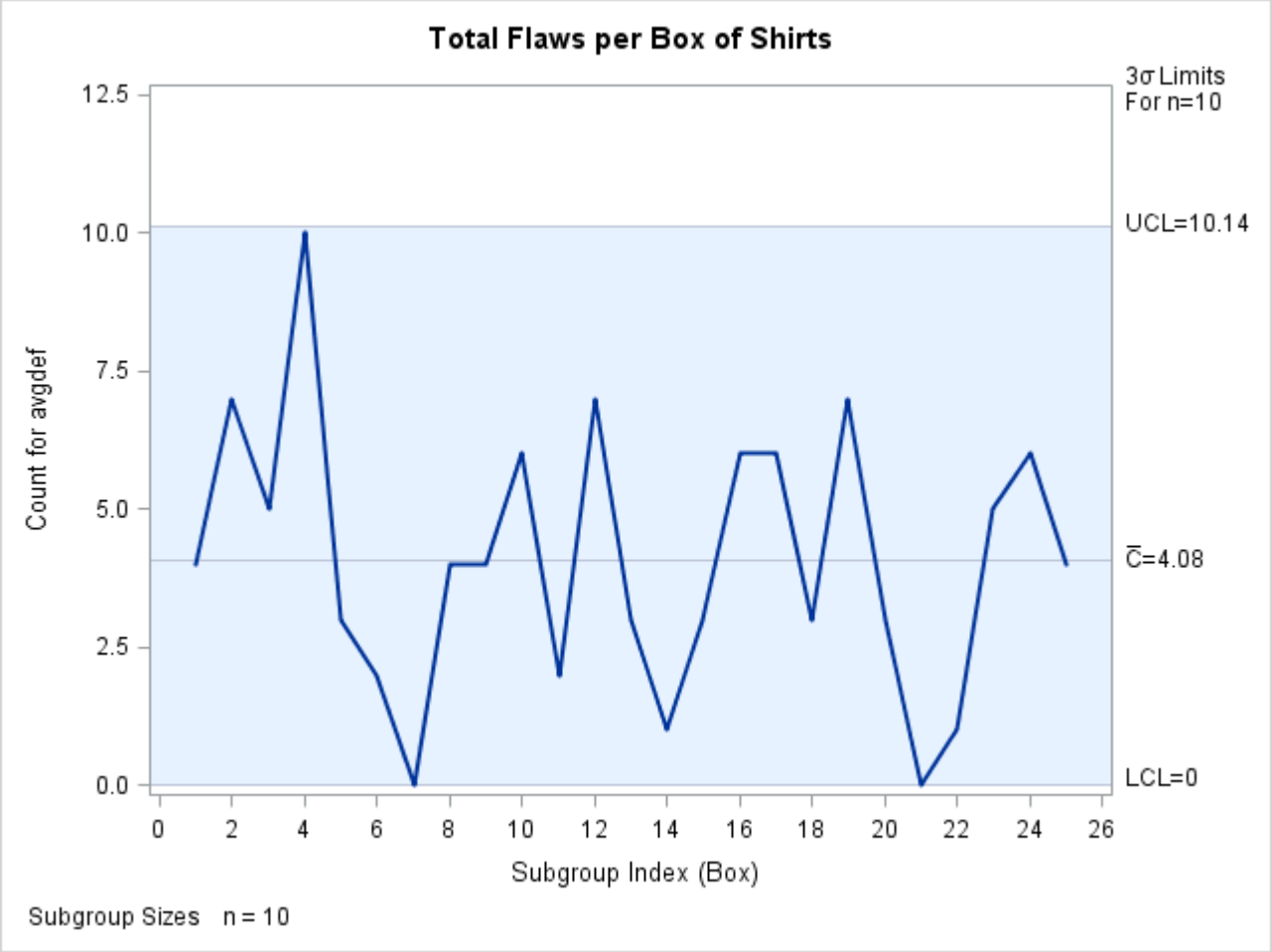
To create a *c* chart plotting the total number of flaws per box (instead of per shirt), you can specify Shirts as a HISTORY= data set.

```
ods graphics on;
title 'Total Flaws per Box of Shirts';
proc shewhart history=Shirts;
   cchart avgdef*Box / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *c* chart is created using ODS Graphics instead of traditional graphics.

Note that avgdef is *not* the name of a SAS variable in the data set but is instead the common prefix for the SAS variable names avgdefu and avgdefn. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name referred to as the *process*. The name Box specified after the asterisk is the name of the *subgroup-variable*. The *c* chart is shown in Figure 19.21.

Figure 19.21 c Chart for Boxes of Shirts (ODS Graphics)



In general, a HISTORY= input data set used with the CCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the CCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables avgdefu and avgdefn, the data set Shirts contained the variables Shirtdef and Sizes. The following statements would temporarily rename Shirtdef and Sizes to avgdefu and avgdefn:

```
proc shewhart
   history=Shirts (rename=(Shirtdef = AvgdefU
                           Sizes    = AvgdefN ));
   cchart Avgdef*Box;
run;
```

For more information, see "HISTORY= Data Set" on page 1509.

Saving Nonconformities per Unit

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named Shirts2, which contains the total defects per box for 20 boxes:

```
data Shirts2;
   input Box Flaws Nshirts @@;
   datalines;
 1  3  10    2  8  10    3 15  25    4 20  25
 5  9  25    6  1  10    7  1  10    8 21  50
 9  3  10   10  7  10   11  1  10   12 21  25
13  9  25   14  3  25   15 12  50   16 18  50
17  7  10   18  4  10   19  8  10   20  4  10
;
```

A partial listing of Shirts2 is shown in Figure 19.22.

Figure 19.22 The Data Set Shirts2

Number of Shirt Flaws per Box

Box	Flaws	Nshirts
1	3	10
2	8	10
3	15	25
4	20	25
5	9	25

The variable Box contains the box number, the variable Flaws contains the number of flaws in each box, and the variable Nshirts contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=Shirts2;
   cchart Flaws*Box / subgroupn  = Nshirts
                      outhistory = shirthist
                      nochart;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number

of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of Shirthist is shown in Figure 19.23.

Figure 19.23 The Data Set Shirthist

Average Defects Per Shirt

Box	FlawsU	FlawsN
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25

There are three variables in the data set Shirthist.

- Box contains the subgroup index.
- FlawsU contains the numbers of nonconformities per inspection unit.
- FlawsN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* Defects specified in the CCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1506.

Syntax: CCHART Statement

The basic syntax for the CCHART statement is as follows:

CCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

CCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of CCHART statements in the SHEWHART procedure. The components of the CCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities.

  For an example, see "Creating c Charts from Defect Count Data" on page 1483.

- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set.

  For an example, see "Creating c Charts from Nonconformities per Unit" on page 1488.

- If numbers of nonconformities per subgroup, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set.

  For an example, see "Saving Control Limits" on page 1484.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *c* charts for Defects and Flaws:

```
proc shewhart data=Info;
   cchart (Defects Flaws)*Sample;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding CCHART statement, SAMPLE is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create a *c* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Info lineprinter;
   cchart Defects*Sample='*';
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the CCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.14 CCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	Requests probability limits at discrete values
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	Displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
CSYMBOL=	Specifies label for central line
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMITLABEL	Suppresses labels for control limits and central line

Table 19.14 *continued*

Option	Description
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
Standard Value Options	
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U0=	Specifies known average number of nonconformities per unit
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on *c* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles

Table 19.14 *continued*

Option	Description
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive

Table 19.14 *continued*

Option	Description
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart

Table 19.14 *continued*

Option	Description
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays c chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= options
CVREF=	Specifies color for lines requested by VREF= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines

Table 19.14 *continued*

Option	Description
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features chart
DESCRIPTION=	Specifies description of *c* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of *c* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines

Table 19.14 *continued*

Option	Description
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
SUBGROUPN	Specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits

Table 19.14 *continued*

Option	Description
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars

Table 19.14 *continued*

Option	Description
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Table 19.14 *continued*

Option	Description
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: CCHART Statement

The following sections provide details that are specific to the CCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Numbers of Nonconformities (c Charts)

The following notation is used in this section:

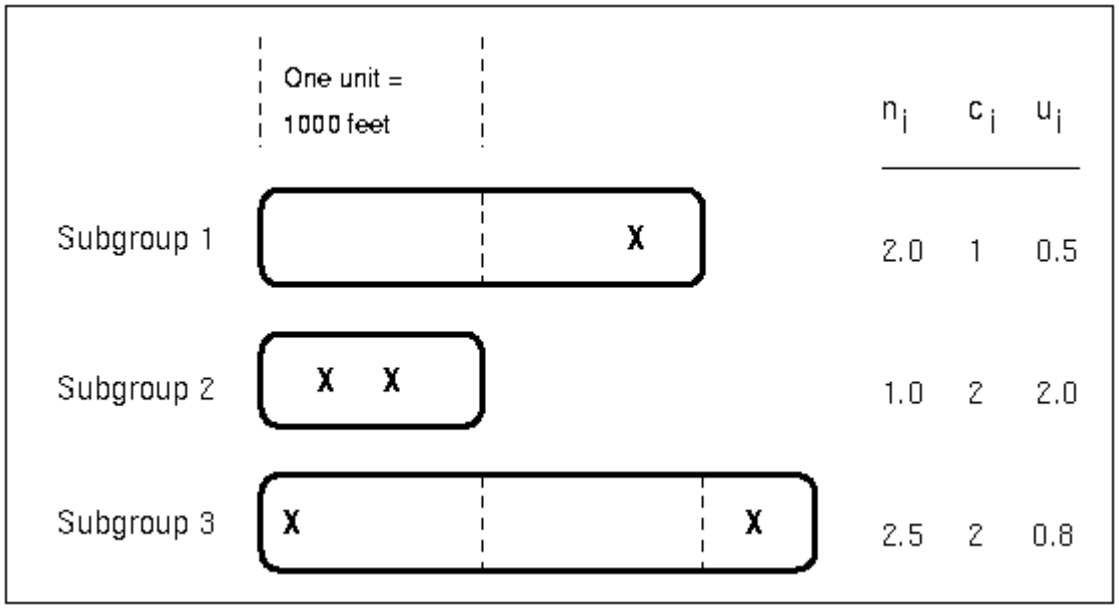
u	expected number of nonconformities per unit produced by the process
u_i	number of nonconformities per unit in the ith subgroup
c_i	total number of nonconformities in the ith subgroup
n_i	number of inspection units in the ith subgroup. Typically, $n_i = 1$ and $u_i = c_i$ for c charts. In general, $u_i = c_i / n_i$.
$\bar{u}$	average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_N u_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
χ^2_ν	has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a c chart represents the total number of nonconformities (c_i) in a subgroup. For example, Figure 19.24 displays three sections of pipeline that are inspected for defective welds (indicated by an X). Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long

sections. The number of units in the ith subgroup is denoted by n_i, which is the subgroup sample size. The value of n_i can be fractional; Figure 19.24 shows $n_3 = 2.5$ units in the third subgroup.

Figure 19.24 Terminology for c Charts and u Charts



The *number of nonconformities* in the ith subgroup is denoted by c_i. The *number of nonconformities per unit* in the ith subgroup is denoted by $u_i = c_i/n_i$. In Figure 19.24, the number of welds per inspection unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A u chart created with the UCHART statement plots the quantity u_i for the ith subgroup (see "UCHART Statement: SHEWHART Procedure" on page 1801). An advantage of a u chart is that the value of the central line at the ith subgroup does not depend on n_i. This is not the case for a c chart, and consequently, a u chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a c chart, the central line indicates an estimate for $n_i u$, which is computed as $n_i \bar{u}$. If you specify a known value (u_0) for u, the central line indicates the value of $n_i u_0$.

Note that the central line varies with subgroup sample size n_i. When $n_i = 1$ for all subgroups, the central line has the constant value $\bar{c} = (c_1 + \cdots + c_N)/N$.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of c_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that c_i exceeds the limits

The lower and upper control limits, LCLC and UCLC respectively, are given by

$$\begin{aligned} \text{LCLC} &= \max\left(n_i\bar{u} - k\sqrt{n_i\bar{u}}\ , 0\right) \\ \text{UCLC} &= n_i\bar{u} + k\sqrt{n_i\bar{u}} \end{aligned}$$

The upper and lower control limits vary with the number of inspection units per subgroup n_i. If $n_i = 1$ for all subgroups, the control limits have constant values.

$$\begin{aligned} \text{LCLC} &= \max\left(\bar{c} - k\sqrt{\bar{c}}\ , 0\right) \\ \text{UCLC} &= \bar{c} + k\sqrt{\bar{c}} \end{aligned}$$

An upper probability limit UCLC for c_i can be determined using the fact that

$$\begin{aligned} P\{c_i > \text{UCLC}\} &= 1 - P\{c_i \leq \text{UCLC}\} \\ &= 1 - P\{\chi^2_{2(\text{UCLC}+1)} \geq 2n_i\bar{u}\} \end{aligned}$$

The upper probability limit UCLC is then calculated by setting

$$1 - P\{\chi^2_{2(\text{UCLC}+1)} \geq 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLC.

A similar approach is used to calculate the lower probability limit LCLC, using the fact that

$$\begin{aligned} P\{c_i < \text{LCLC}\} &= P\{c_i \leq \text{LCLC} - 1\} \\ &= P\{\chi^2_{2((\text{LCLC}-1)+1)} > 2n_i\bar{u}\} \\ &= P\{\chi^2_{2\text{LCLC}} > 2n_i\bar{u}\} \end{aligned}$$

The lower probability limit LCLC is then calculated by setting

$$P\{\chi^2_{2\text{LCLC}} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLC. This assumes that the process is in statistical control and that c_i has a Poisson distribution. For more information, refer to Johnson, Kotz, and Kemp (1992). Note that the probability limits vary with the number of inspection units per subgroup (n_i) and are asymmetric about the central line.

If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits. You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 19.16 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_C_	Value of central line on c chart ($n_i\bar{u}$ or $n_i u_0$)
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLC	Lower control limit for number of nonconformities
LIMITN	Sample size associated with the control limits
SIGMAS	Multiple (k) of standard error of c_i
SUBGRP	*Subgroup-variable* specified in the CCHART statement
TYPE	Type (estimate or standard value) of _U_
_U_	Average number of nonconformities per unit ($\bar{u}$ or u_0)
UCLC	Upper control limit for number of nonconformities
VAR	*Process* specified in the CCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _C_, _LCLC_, _UCLC_, and _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of c_i, the value of _ALPHA_ is computed as $P\{c_i < \text{\_LCLC\_}\} + P\{c_i > \text{\_UCLC\_}\}$. If control limits vary with subgroup sample size and are determined in terms of k, _ALPHA_ is assigned the special missing value V.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLC\_} - \text{\_C\_})/\sqrt{\text{\_C\_}}$. If probability limits vary with subgroup sample size, _SIGMAS_ is assigned the special missing value V.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the CCHART statement. For an example, see "Saving Control Limits" on page 1484.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup sample size variable named by *process* suffixed with N
- a subgroup number of nonconformities per unit variable named by *process* suffixed with U

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the CCHART statement. For example, consider the following statements:

```
proc shewhart data=Fabric;
   cchart (Flaws Ndefects)*lot / outhistory=Summary;
run;
```

The data set Summary contains variables named lot, FlawsU, FlawsN, NdefctsU, and NdefctsN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example that creates an OUTHISTORY= data set, see "Saving Nonconformities per Unit" on page 1491. Note that an OUTHISTORY= data set created with the CCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.17 lists the variables that are saved.

Table 19.17 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
_C_	Average number of nonconformities
EXLIM	Control limit exceeded on *c* chart
LCLC	Lower control limit for number of nonconformities
LIMITN	Nominal sample size associated with the control limits
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBC	Subgroup number of nonconformities
SUBN	Subgroup sample size
TESTS	Tests for special causes signaled on *c* chart
UCLC	Upper control limit for number of nonconformities
VAR	*Process* specified in the CCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for *c* charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1484.

Input Data Sets

DATA= Data Set

You can read the number of nonconformities in subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the CCHART statement must be a SAS variable in the data set. This variable provides the number of nonconformities in subgroup samples indexed by the *subgroup-variable*. Typically (but not necessarily), the subgroup consists of a single inspection unit. The *subgroup-variable*, specified in the CCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set must contain one observation per subgroup. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating c Charts from Defect Count Data" on page 1483.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   cchart Defects*Lot;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLC_, _C_, and _UCLC_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see "Control Limits" on page 1504)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1487.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set. A HISTORY= data set used with the CCHART statement must contain the following variables:

- *subgroup-variable*
- subgroup number of nonconformities per unit variable for each *process*

- subgroup sample size variable (number of units per subgroup) for each *process*

The names of the subgroup number of nonconformities per unit and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *U* and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   cchart (flaws ndefects)*lot;
run;
```

The data set summary must include the variables lot, flawsU, flawsN, ndefctsU, and ndefctsN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character. Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating c Charts from Nonconformities per Unit" on page 1488.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.18 lists the variables required in a TABLE= data set used with the CCHART statement.

Table 19.18 Variables Required in a TABLE= Data Set

Variable	Description
_C_	Average number of nonconformities
LCLC	Lower control limit for nonconformities
LIMITN	Nominal sample size associated with the control limits
Subgroup-variable	Values of the *subgroup-variable*
SUBC	Subgroup number of nonconformities
SUBN	Subgroup sample size
UCLC	Upper control limit for nonconformities

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1484.

Examples: CCHART Statement

This section provides advanced examples of the CCHART statement.

Example 19.8: Applying Tests for Special Causes

NOTE: See *Tests for Special Causes Applied to c Chart* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make *c* charts more sensitive to special causes of variation. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named Trucks3:

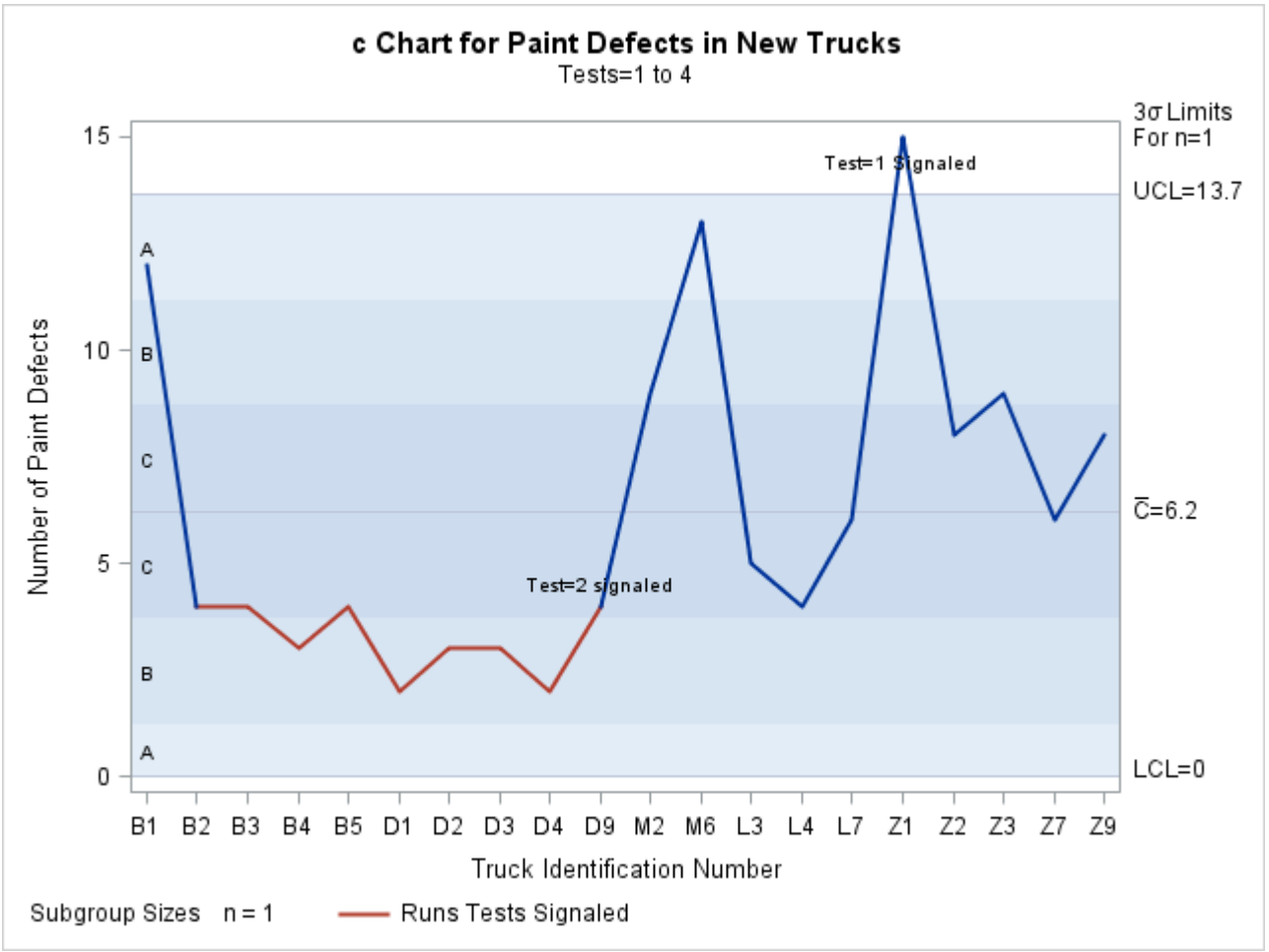
```
data Trucks3;
   input TruckID $ Defects @@;
   label TruckID='Truck Identification Number'
         Defects='Number of Paint Defects';
   datalines;
B1  12    B2  4    B3  4    B4   3
B5   4    D1  2    D2  3    D3   3
D4   2    D9  4    M2  9    M6  13
L3   5    L4  4    L7  6    Z1  15
Z2   8    Z3  9    Z7  6    Z9   8
;
```

The following statements create a *c* chart and tabulate the information on the chart. The chart and table are shown in Output 19.8.1 and Output 19.8.2.

```
ods graphics on;
title1 'c Chart for Paint Defects in New Trucks';
title2 'Tests=1 to 4';
proc shewhart data=Trucks3;
   cchart Defects*TruckID / tests      = 1 to 4
                            testlabel1 = 'Test=1 Signaled'
                            testlabel2 = 'Test=2 signaled'
                            odstitle   = title
                            odstitle2  = title2
                            zonelabels
                            tabletests
                            tablelegend;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. Only Tests 1, 2, 3, and 4 are recommended for *c* charts. The TESTLABEL1= and TESTLABEL2= options specify the labels for points where Tests 1 and 2 are positive. The TESTFONT= option specifies the font for the labels indicating points at which the tests are positive.

Output 19.8.1 Tests for Special Causes Displayed on *c* Chart



Output 19.8.2 Tabular Form of c Chart

c Chart for Paint Defects in New Trucks
Tests=1 to 4

The SHEWHART Procedure

c Chart Summary for Defects

		3 Sigma Limits with n=1 for Count			
TruckID	Subgroup Sample Size	Lower Limit	Subgroup Count	Upper Limit	Special Tests Signaled
B1	1.00000	0	12.000000	13.669940	
B2	1.00000	0	4.000000	13.669940	
B3	1.00000	0	4.000000	13.669940	
B4	1.00000	0	3.000000	13.669940	
B5	1.00000	0	4.000000	13.669940	
D1	1.00000	0	2.000000	13.669940	
D2	1.00000	0	3.000000	13.669940	
D3	1.00000	0	3.000000	13.669940	
D4	1.00000	0	2.000000	13.669940	
D9	1.00000	0	4.000000	13.669940	2
M2	1.00000	0	9.000000	13.669940	
M6	1.00000	0	13.000000	13.669940	
L3	1.00000	0	5.000000	13.669940	
L4	1.00000	0	4.000000	13.669940	
L7	1.00000	0	6.000000	13.669940	
Z1	1.00000	0	15.000000	13.669940	1
Z2	1.00000	0	8.000000	13.669940	
Z3	1.00000	0	9.000000	13.669940	
Z7	1.00000	0	6.000000	13.669940	
Z9	1.00000	0	8.000000	13.669940	

Test Descriptions	
Test 1	One point beyond Zone A (outside control limits)
Test 2	Nine points in a row on one side of center line

The ZONELABELS option requests zone lines and displays zone labels on the chart. The zones are used to define the tests. The TABLETESTS option requests a table of counts of nonconformities, subgroup sample sizes, and control limits, together with a column indicating the subgroups at which the tests are positive. The TABLELEGEND option adds a legend describing the tests that are positive.

Output 19.8.1 and Output 19.8.2 indicate that Test 1 is positive at Truck Z1 and Test 2 is positive at Truck D9.

Example 19.9: Specifying a Known Expected Number of Nonconformities

NOTE: See *c Chart Based on Known (Standard) Value* in the SAS/QC Sample Library.

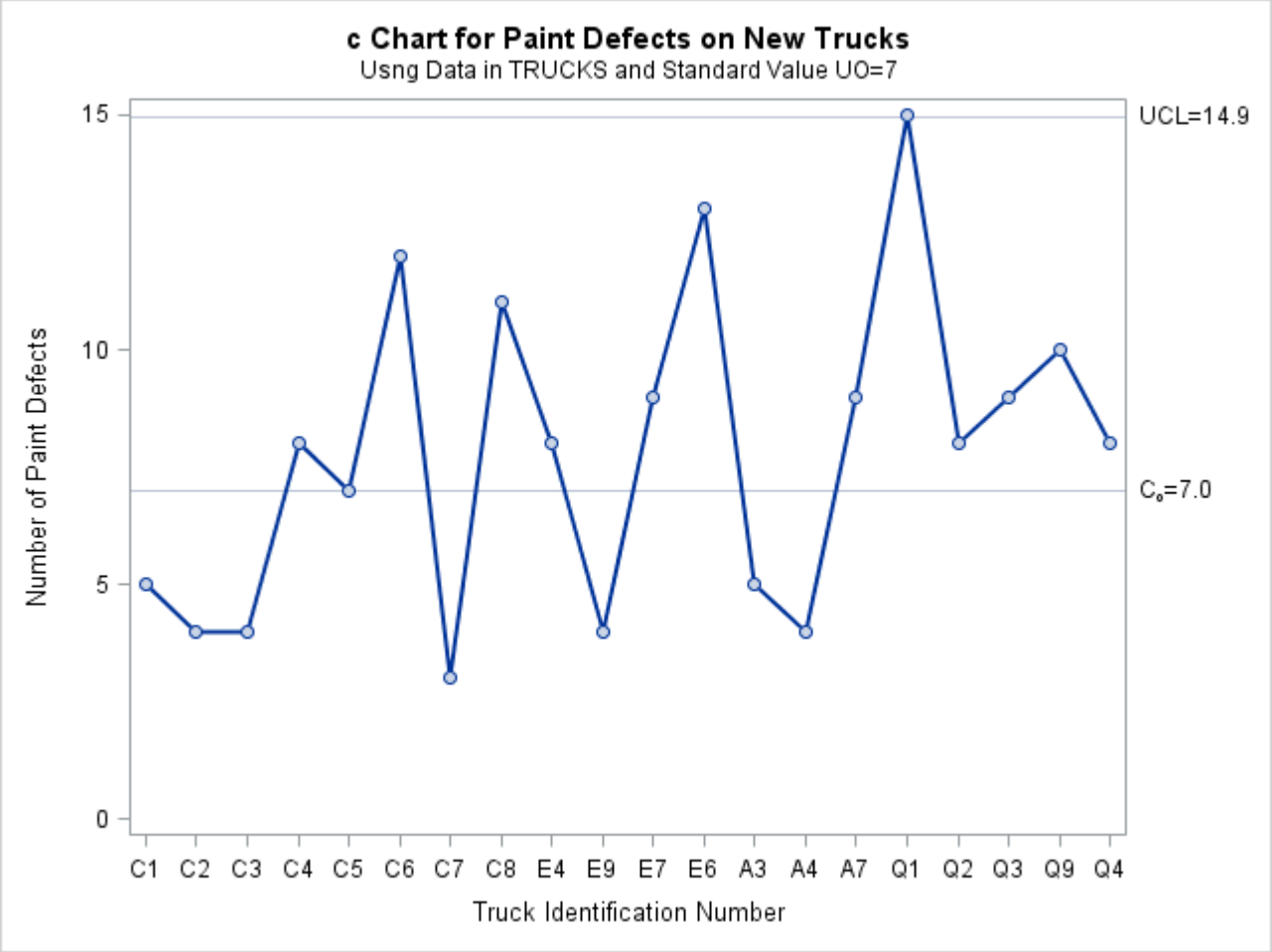
This example illustrates how you can create a *c* chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

A *c* chart is used to monitor the number of paint defects per truck. The defect counts are provided as values of the variable Defects in the data set Trucks given in "Creating c Charts from Defect Count Data" on page 1483. Based on previous testing, it is known that $u_0 = 7$. The following statements create a *c* chart with control limits derived from this value:

```
ods graphics on;
title 'c Chart for Paint Defects on New Trucks';
title2 'Usng Data in TRUCKS and Standard Value UO=7';
proc shewhart data=Trucks;
   cchart Defects*TruckID / u0        = 7
                            csymbol   = c0
                            odstitle  = title
                            odstitle2 = title2
                            markers
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart is shown in Output 19.9.1. The U0= option specifies u_0, and the CSYMBOL= option requests a label for the central line indicating that the line represents a standard value. The NOLEGEND option suppresses the legend for the subgroup sample size, and the NOLIMITSLEGEND option suppresses the legend for the control limits that appears by default in the upper right corner of the chart. The NOLIMIT0 option suppresses the display of the lower limit when it is equal to zero.

Output 19.9.1 A c Chart with Standard Value u_0



The number of paint defects on Truck Q1 exceeds the upper control limit, indicating that the process is out of control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data tlimits;
   length _subgrp_ _var_ _type_ $8;
   _U_      = 7;
   _subgrp_ = 'truckid';
   _var_    = 'defects';
   _limitn_ = 1;
   _type_   = 'STANDARD';

proc shewhart data=trucks limits=tlimits;
   cchart defects*truckid / csymbol=c0
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart produced by these statements is identical to the one in Output 19.9.1.

For further details, see "LIMITS= Data Set" on page 1509.

Example 19.10: Creating c Charts for Varying Numbers of Units

NOTE: See *c Chart for Varying Number of Inspection Units* in the SAS/QC Sample Library.

In applications where the number of inspection units per subgroup is not equal to one, a *u* chart is typically used to analyze the number of nonconformities *per unit* (see "UCHART Statement: SHEWHART Procedure" on page 1801). However, as shown in this example, you can use the CCHART statement to create a *c* chart for this type of data.

Figure 19.25 Difference between *c* Charts and *u* Charts

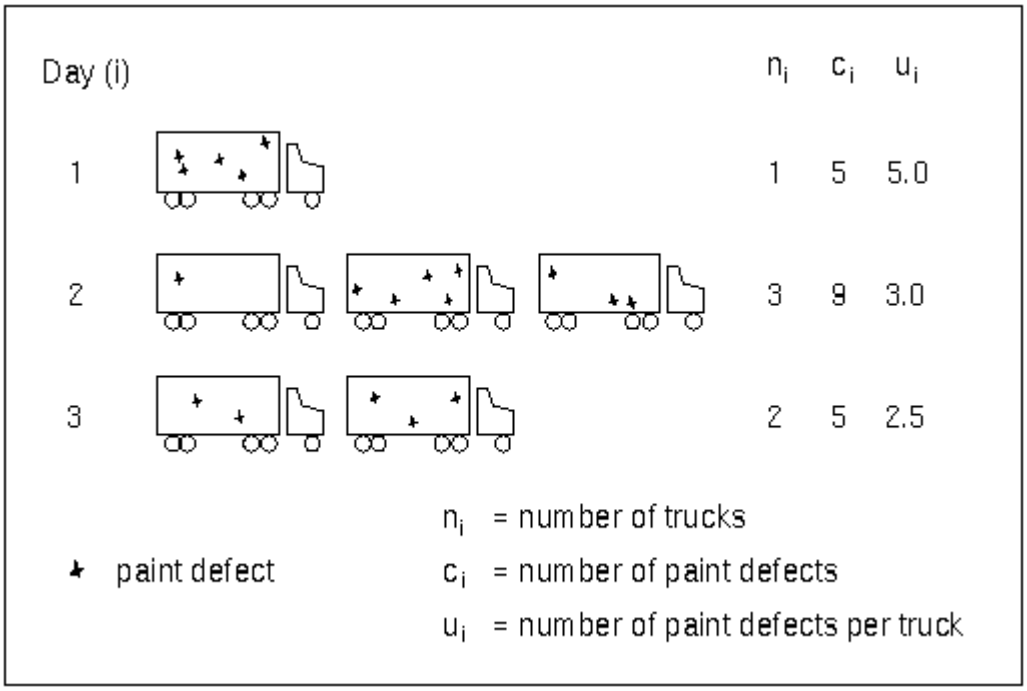


Figure 19.25 illustrates a situation in which varying numbers of trucks are painted each day. Trucks painted on the same day are regarded as *subgroups*, and each truck is regarded as an *inspection unit*. The following statements create a SAS data set named Trucks4, which contains paint defects for trucks painted on 26 days:

```
data Trucks4;
   input Day Defects Ntrucks @@;
   label Day='Day'
         Defects='Number of Paint Defects';
   datalines;
 1   5  1    2   9  3
 3   5  2    4   9  2
 5  24  4    6  10  2
 7  15  3    8  17  3
 9  16  3   10  13  2
```

```
11  28  4   12  18  5
13   8  2   14   7  2
15   5  1   16  17  3
17   2  1   18  17  3
19  15  4   20  19  5
21   6  3   22  23  5
23  27  4   24   6  2
25  12  2   26  12  3
;
```

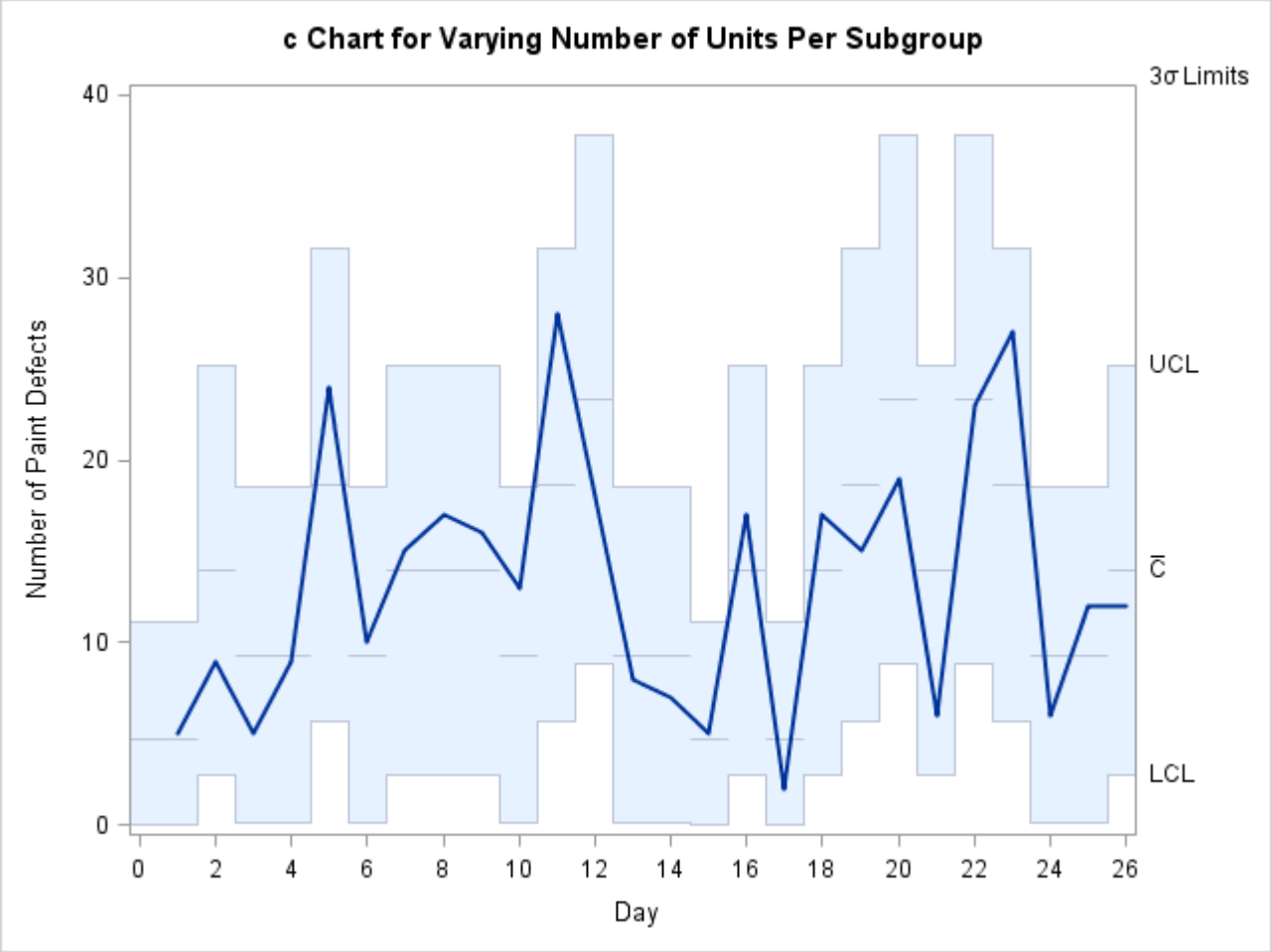
The variable Defects provides the defect count (c_i) for the ith day, and the variable Ntrucks provides the number of inspection units (n_i). The following statements create a c chart for this data:

```
ods graphics on;
title 'c Chart for Varying Number of Units Per Subgroup';
proc shewhart data=Trucks4;
   cchart Defects*Day / subgroupn = Ntrucks
                        odstitle  = title
                        nolegend;
run;
```

The SUBGROUPN= option specifies the subgroup sample size variable Ntrucks (in general, the values of this variable need not be integers). Alternatively, you can specify a fixed value with the SUBGROUPN= option. When this option is not specified, it is assumed that $n_i = 1$.

The chart is shown in Output 19.10.1. Note that the central line and the control limits vary with the number of inspection units.

Output 19.10.1 c Chart for Varying Number of Units



IRCHART Statement: SHEWHART Procedure

Overview: IRCHART Statement

The IRCHART statement creates control charts for individual measurements and moving ranges. These charts are appropriate when only one measurement is available for each subgroup sample and when the measurements are independently and normally distributed.

You can use options in the IRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the individual measurements and moving ranges or as probability limits
- tabulate individual measurements, moving ranges, and control limits
- save control limits in an output data set

- save individual measurements and moving ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) process mean and standard deviation for computing control limits
- specify the number of consecutive measurements to use when computing the moving ranges
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing charts of individual measurements and moving ranges with the IRCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: IRCHART Statement

This section introduces the IRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the IRCHART statement is presented in the section "Syntax: IRCHART Statement" on page 1529, and advanced examples are given in the section "Examples: IRCHART Statement" on page 1551.

Creating Individual Measurements and Moving Range Charts

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

An aeronautics company manufacturing jet engines measures the inner diameter of the forward face of each engine (in centimeters). The following statements create a SAS data set that contains the diameter measurements for 20 engines:

```
data Jets;
   input Engine Diam @@;
   label Engine = "Engine Number";
   datalines;
 1 78.4   2 80.1   3 84.4   4 79.1   5 80.4
 6 83.5   7 73.8   8 83.5   9 75.0  10 76.8
11 70.5  12 80.3  13 82.4  14 79.4  15 86.4
16 90.5  17 77.7  18 82.5  19 79.9  20 83.2
;
```

A partial listing of Jets is shown in Figure 19.26.

Figure 19.26 Partial Listing of the Data Set Jets

The Data Set JETS

Engine	Diam
1	78.4
2	80.1
3	84.4
4	79.1
5	80.4

Each observation contains the diameter measurement and identification number for a particular engine. The variable Engine identifies the sequence of engines and is referred to as the *subgroup-variable*.[4] The variable Diam contains the measurements and is referred to as the *process variable* (or *process* for short).

Because the production rate is low, individual measurements and moving range charts are used to monitor the process. The following statements create the charts shown in Figure 19.27:

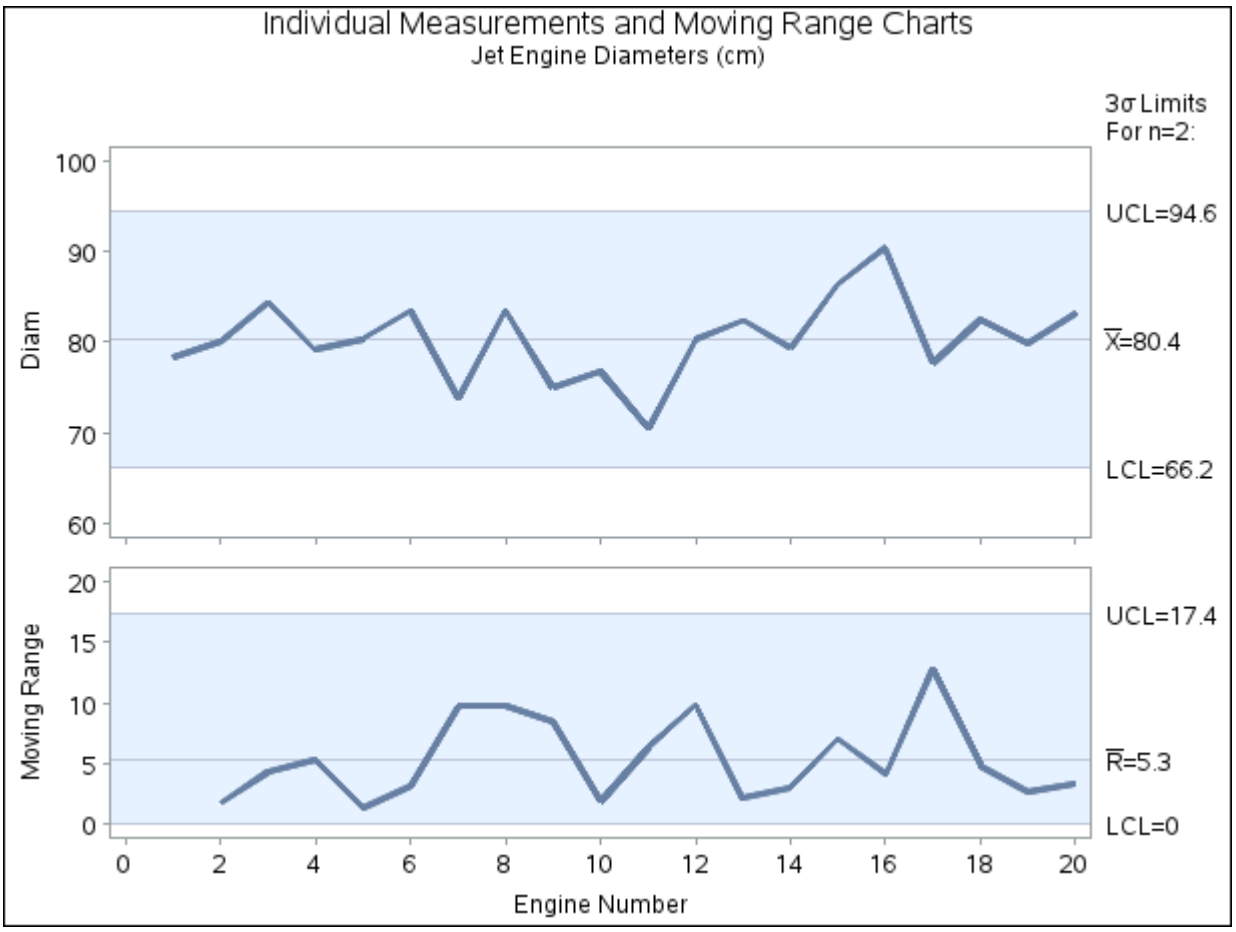
```
ods graphics off;
title 'Individual Measurements and Moving Range Charts';
title2 'Jet Engine Diameters (cm)';
proc shewhart data=Jets;
   irchart Diam*Engine;
run;
```

This example illustrates the basic form of the IRCHART statement. After the keyword IRCHART, you specify the *process* to analyze (in this case, Diam), followed by an asterisk and the *subgroup-variable* (Engine).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

[4] Technically, the data for individual measurements and moving range charts are not arranged in rational subgroups. The term *subgroup-variable* is used for consistency with other chart statements in the SHEWHART procedure, and it is convenient to think of the "subgroups" as consisting of single measurements.

Figure 19.27 Individual Measurements and Moving Range Charts (Traditional Graphics)



Each point on the individual measurements chart indicates the inner diameter of a particular engine. Each point on the moving range chart indicates the range of the two most recent measurements. For instance, the moving range plotted for the second engine is $|78.4 - 80.1| = 1.7$. No moving range is plotted for the first engine. Because all of the individual measurements and moving ranges lie within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in section "Limits for Individual Measurements and Moving Range Charts" on page 1543. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1526.

Saving Individual Measurements and Moving Ranges

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

In this example, the IRCHART statement is used to create an output data set containing individual measurements and moving ranges. The following statements read the diameter measurements from the data set Jets (see "Creating Individual Measurements and Moving Range Charts" on page 1519) and create a data set named Jetinfo:

```
proc shewhart data=Jets;
   irchart Diam*Engine / outhistory = Jetinfo
                         nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to those in Figure 19.27. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the IRCHART statement. A complete list of options is presented in the section "Syntax: IRCHART Statement" on page 1529.

Figure 19.28 contains a partial listing of Jetinfo.

Figure 19.28 The Data Set Jetinfo

Individual Measurements and Moving Ranges for Diameters

Engine	Diam	DiamR
1	78.4	.
2	80.1	1.7
3	84.4	4.3
4	79.1	5.3
5	80.4	1.3

The data set Jetinfo contains one observation for each engine, and it includes three variables.

- Engine contains the subgroup index.
- Diam contains the individual measurements.
- DiamR contains the moving ranges.

Note that the variable containing the moving ranges is named by adding the suffix character *R* to the *process* Diam specified in the IRCHART statement.

For more information, see "OUTHISTORY= Data Set" on page 1545.

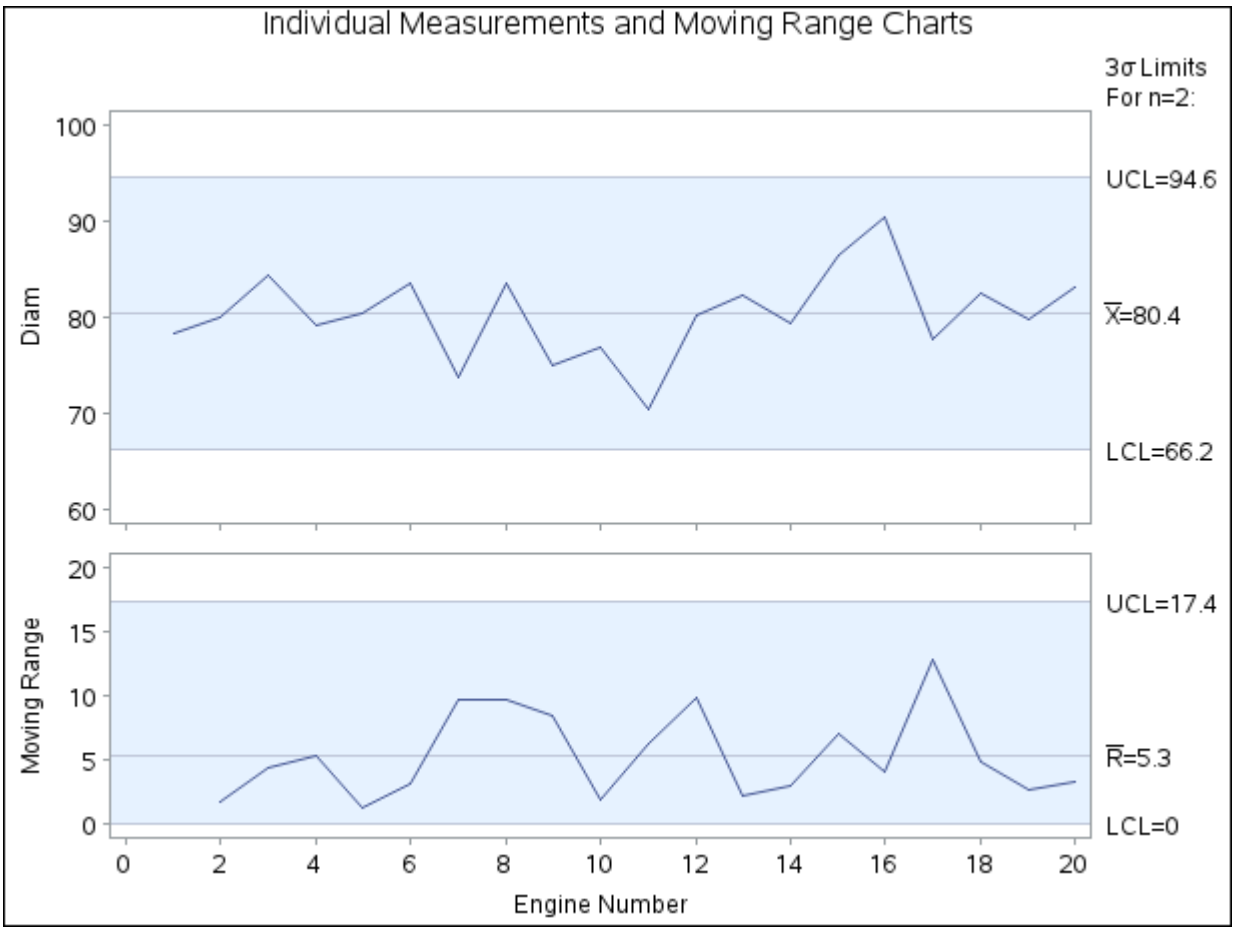
Reading Individual Measurements and Moving Ranges

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

In some applications, both individual measurements and moving ranges might be provided. You can read this type of data set by specifying it with the HISTORY= option in the PROC SHEWHART statement. For example, the following statements read the data set Jetinfo (see Figure 19.28) and create the charts shown in Figure 19.29:

```
symbol h = .8;
title 'Individual Measurements and Moving Range Charts';
proc shewhart history=Jetinfo;
   irchart Diam*Engine;
run;
```

Figure 19.29 Charts Produced from Summary Data Set Jetinfo



A HISTORY= data set used with the IRCHART statement must contain the following variables:

- subgroup variable
- individual measurements variable
- moving range variable

Furthermore, the name of the moving range variable must begin with the *process* name specified in the IRCHART statement and end with the special suffix character *R*. If the name does not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename this variable for the duration of the procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1885). For more information, see "HISTORY= Data Set" on page 1548.

Saving Control Limits

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

You can save the control limits for individual measurements and moving range charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1526) or modify the limits with a DATA step program.

The following statements read the diameter measurements from the data set Jets (see "Creating Individual Measurements and Moving Range Charts" on page 1519) and save the control limits displayed in Figure 19.27 in a data set named Jetlim:

```
proc shewhart data=Jets;
   irchart Diam*Engine / outlimits = Jetlim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Jetlim is listed in Figure 19.30.

Figure 19.30 The Data Set Jetlim Containing Control Limit Information

Control Limits for Diameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_	_UCLI_
Diam	Engine	ESTIMATE	2	.002699796	3	66.2290	80.39	94.5510

LCLR	_R_	_UCLR_	_STDDEV_
0	5.32632	17.3986	4.72032

The data set Jetlim contains one observation with the limits for *process* Diam. The variables _LCLI_ and _UCLI_ contain the control limits for the individual measurements, and the variable _MEAN_ contains the central line. The variables _LCLR_ and _UCLR_ contain the control limits for the moving ranges, and the variable _R_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the number of consecutive measurements used to compute the moving ranges, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1544.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Jets;
   irchart Diam*Engine / outtable=Jtable
                         nochart;
run;
```

The data set Jtable is listed in Figure 19.31.

Figure 19.31 The Data Set Jtable

Summary Statistics and Control Limit Information

VAR	Engine	_SIGMAS_	_LIMITN_	_LCLI_	_SUBI_	_MEAN_	_UCLI_	_STDDEV_	_EXLIM_
Diam	1	3	2	66.2290	78.4	80.39	94.5510	4.72032	
Diam	2	3	2	66.2290	80.1	80.39	94.5510	4.72032	
Diam	3	3	2	66.2290	84.4	80.39	94.5510	4.72032	
Diam	4	3	2	66.2290	79.1	80.39	94.5510	4.72032	
Diam	5	3	2	66.2290	80.4	80.39	94.5510	4.72032	
Diam	6	3	2	66.2290	83.5	80.39	94.5510	4.72032	
Diam	7	3	2	66.2290	73.8	80.39	94.5510	4.72032	
Diam	8	3	2	66.2290	83.5	80.39	94.5510	4.72032	
Diam	9	3	2	66.2290	75.0	80.39	94.5510	4.72032	
Diam	10	3	2	66.2290	76.8	80.39	94.5510	4.72032	
Diam	11	3	2	66.2290	70.5	80.39	94.5510	4.72032	
Diam	12	3	2	66.2290	80.3	80.39	94.5510	4.72032	
Diam	13	3	2	66.2290	82.4	80.39	94.5510	4.72032	
Diam	14	3	2	66.2290	79.4	80.39	94.5510	4.72032	
Diam	15	3	2	66.2290	86.4	80.39	94.5510	4.72032	
Diam	16	3	2	66.2290	90.5	80.39	94.5510	4.72032	
Diam	17	3	2	66.2290	77.7	80.39	94.5510	4.72032	
Diam	18	3	2	66.2290	82.5	80.39	94.5510	4.72032	
Diam	19	3	2	66.2290	79.9	80.39	94.5510	4.72032	
Diam	20	3	2	66.2290	83.2	80.39	94.5510	4.72032	

LCLR	_SUBR_	_R_	_UCLR_	_EXLIMR_
0	.	5.32632	17.3986	
0	1.7	5.32632	17.3986	
0	4.3	5.32632	17.3986	
0	5.3	5.32632	17.3986	
0	1.3	5.32632	17.3986	
0	3.1	5.32632	17.3986	
0	9.7	5.32632	17.3986	
0	9.7	5.32632	17.3986	
0	8.5	5.32632	17.3986	
0	1.8	5.32632	17.3986	
0	6.3	5.32632	17.3986	
0	9.8	5.32632	17.3986	
0	2.1	5.32632	17.3986	
0	3.0	5.32632	17.3986	
0	7.0	5.32632	17.3986	
0	4.1	5.32632	17.3986	
0	12.8	5.32632	17.3986	
0	4.8	5.32632	17.3986	
0	2.6	5.32632	17.3986	
0	3.3	5.32632	17.3986	

This data set contains one observation for each subgroup. The variables _SUBI_ and _SUBR_ contain the individual measurements and moving ranges. The variables _LCLI_ and _UCLI_ contain the lower and upper

control limits for the individual measurements chart, and the variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the moving range chart. The variable _MEAN_ contains the central line of the individual measurements chart, and the variable _R_ contains the central line of the moving range chart. The variables _VAR_ and Engine contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1546.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Jtable and display charts (not shown here) identical to those in Figure 19.27:

```
title 'Individual Measurements and Moving Range Control Charts';
title2 'Jet Engine Diameters (cm)';
proc shewhart table=Jtable;
   irchart Diam*Engine;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143).

For more information, see "TABLE= Data Set" on page 1549.

Reading Preestablished Control Limits

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Jetlim saved control limits computed from the measurements in Jets. This example shows how these limits can be applied to data for an additional 20 jet engines provided in the following data set:

```
data Jets2;
   input Engine Diam @@;
   label Diam   = "Inner Diameter (cm)"
         Engine = "Engine Number";
   datalines;
21 81.8  22 87.5  23 80.0  24 89.3  25 83.9
26 76.3  27 75.8  28 82.4  29 82.6  30 77.7
31 79.3  32 81.4  33 76.8  34 75.9  35 86.3
36 77.4  37 80.9  38 87.1  39 85.7  40 73.3
;
```

The following statements create individual measurements and moving range charts for the data in Jets2 using the control limits in Jetlim:

```
options nogstyle;
goptions ftext='albany amt';
title 'Individual Measurements and Moving Range Control Charts';
proc shewhart data=Jets2 limits=Jetlim;
   irchart Diam*Engine / cframe = vigb
                         cconnect = yellow
                         coutfill = red
                         cinfill = vlib;
run;
options gstyle;
```

The charts are shown in Figure 19.32. The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the IRCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently.

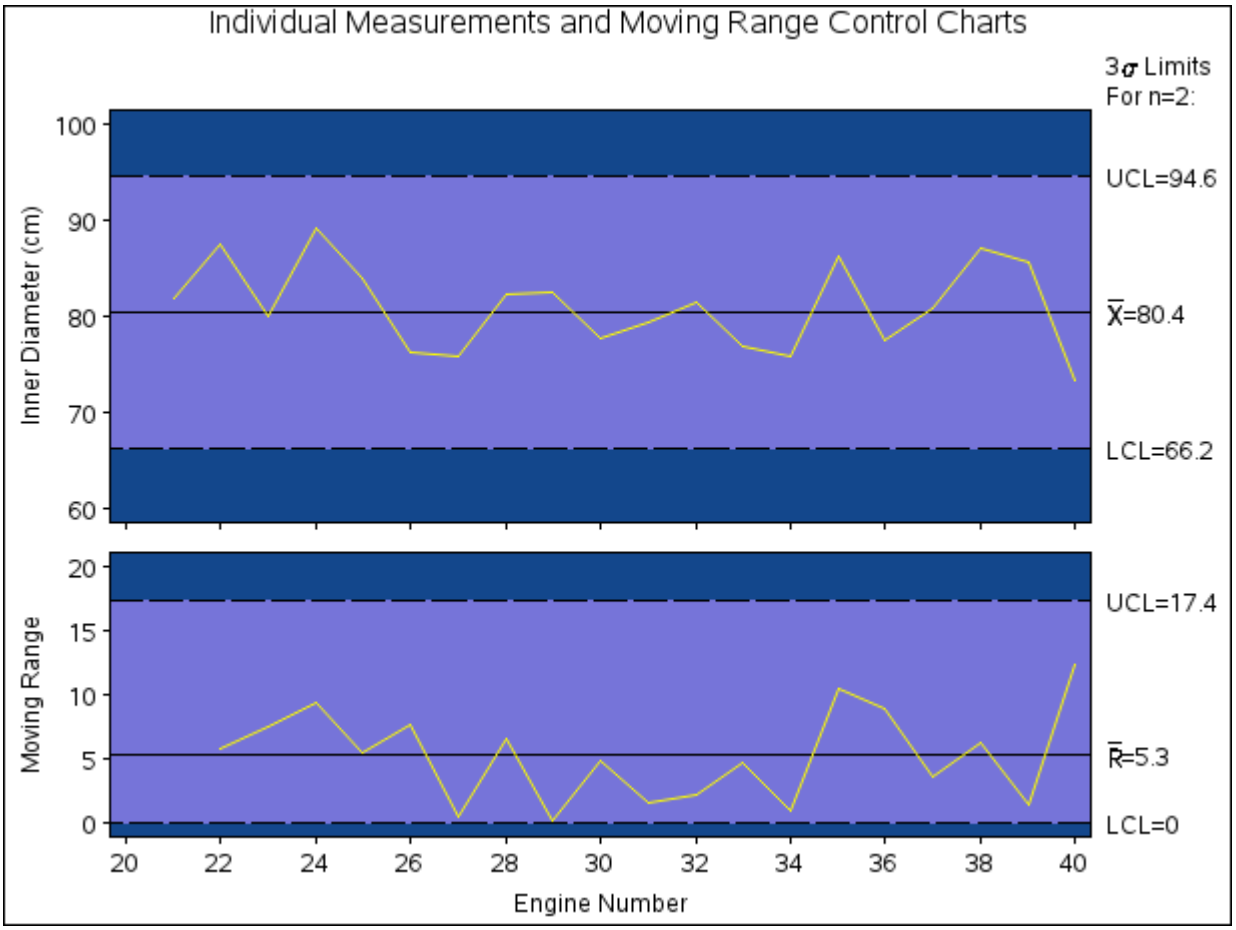
The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Diam
- the value of _SUBGRP_ matches the *subgroup-variable* name Engine

The charts indicate that the process is in control, because all the individual measurements and moving ranges lie within their respective control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1547 for details concerning the variables that you must provide.

Figure 19.32 Charts for Second Set of Engine Noise Levels (Traditional Graphics with NOGSTYLE



Specifying the Computation of the Moving Range

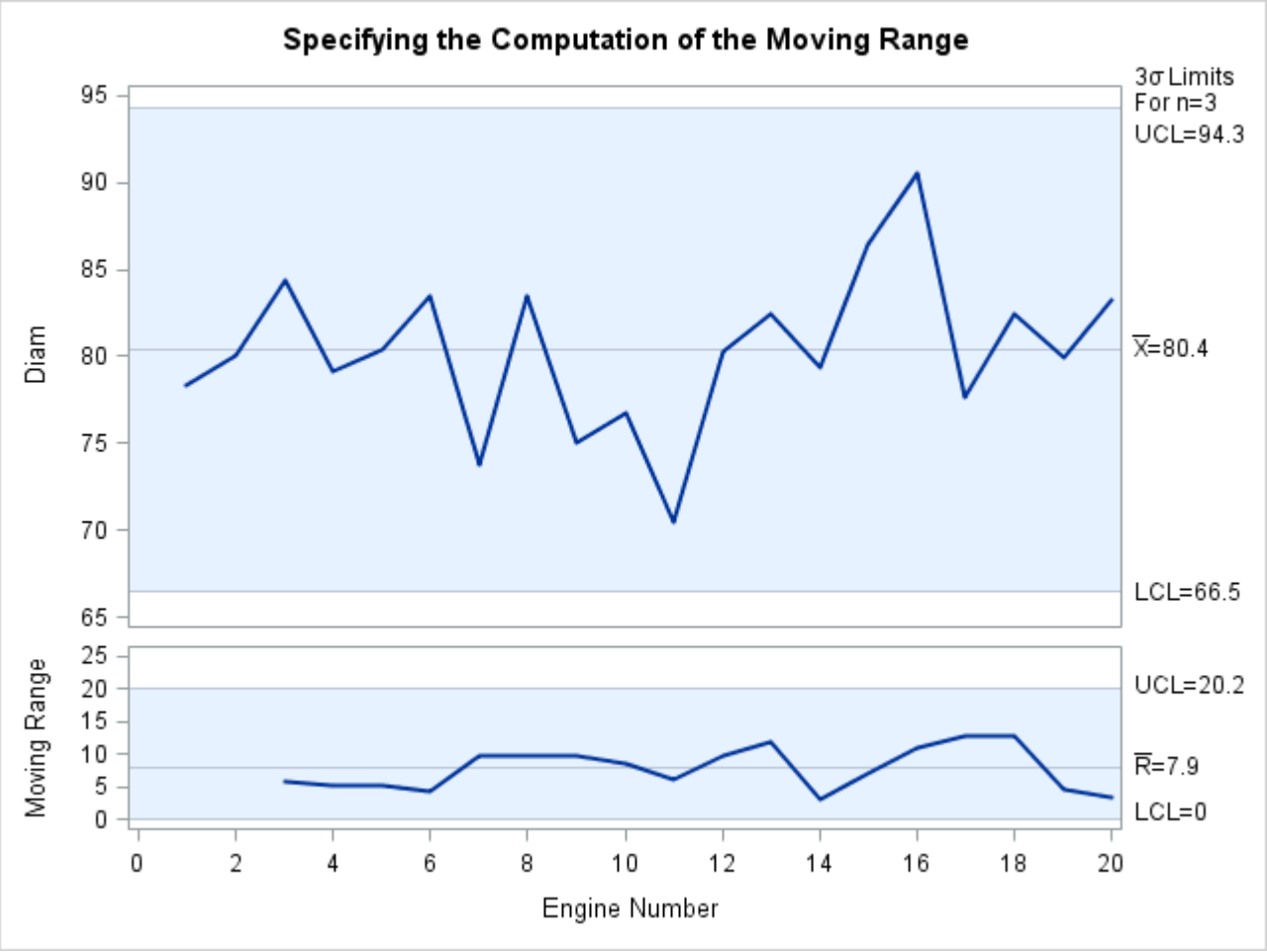
NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

By default, the IRCHART statement uses two consecutive measurements to calculate moving ranges. However, you can specify a different number of measurements to use, as illustrated by the following statements:

```
ods graphics on;
title 'Specifying the Computation of the Moving Range';
proc shewhart data=Jets;
   irchart Diam*Engine / limitn=3 odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the charts are created by using ODS Graphics instead of traditional graphics. The LIMITN= option specifies the number of consecutive measurements used to compute the moving ranges. The resulting charts are shown in Figure 19.33.

Figure 19.33 Computing Moving Ranges from Three Consecutive Measurements (ODS Graphics)



Note that the LIMITN= value is displayed in the legend above the control limit labels. The charts indicate that the process is in control, because all the points lie within the control limits.

Syntax: IRCHART Statement

The basic syntax for the IRCHART statement is as follows:

IRCHART *process* * *subgroup-variable* **;**

The general form of this syntax is as follows:

IRCHART *processes* * *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of IRCHART statements in the SHEWHART procedure. The components of the IRCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the individual measurements. For an example, see "Creating Individual Measurements and Moving Range Charts" on page 1519.
- If individual measurements and moving ranges are read from a HISTORY= data set, *process* must be the name of the variable containing the individual measurements as well as the prefix of the variable containing the moving ranges in the HISTORY= data set. For an example, see "Saving Individual Measurements and Moving Ranges" on page 1521.
- If individual measurements, moving ranges, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1523.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct individual measurements and moving range charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   irchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding IRCHART statement, Day is the subgroup variable. Note that each "subgroup" consists of a single observation. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the individual measurements and moving ranges.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create charts using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   irchart Weight*Day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the IRCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.19 IRCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
MRRESTART	restarts the moving range computation at missing values
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)

Table 19.19 *continued*

Option	Description
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit on individual measurements chart
LCLLABEL2=	Specifies label for lower control limit on moving range chart
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on individual measurements chart
NDECIMAL2=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on moving range chart
NOCTL	Suppresses display of central line on individual measurements chart
NOCTL2	Suppresses display of central line on moving range chart
NOLCL	Suppresses display of lower control limit on individual measurements chart
NOLCL2	Suppresses display of lower control limit on moving range chart
NOLIMIT0	Suppresses display of zero lower control limit on moving range chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit on individual measurements chart
NOUCL2	Suppresses display of upper control limit on moving range chart
RSYMBOL=	Specifies label for central line on moving range chart
UCLLABEL=	Specifies label for upper control limit on individual measurements chart

Table 19.19 *continued*

Option	Description
UCLLABEL2=	Specifies label for upper control limit on moving range chart
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line on individual measurements chart
Process Mean and Standard Deviation Options	
MU0=	Specifies known value of μ_0 for process mean μ
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on individual measurements chart
ALLLABEL2=	Labels every point on moving range chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits on individual measurements chart
OUTLABEL2=	Labels points outside control limits on moving range chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically

Table 19.19 *continued*

Option	Description
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes for the individual measurements chart
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied to the individual measurements chart
ZONELABELS	Adds labels A, B, and C to zone lines for individual measurements chart
ZONES	Adds lines to individual measurements chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels individual measurements chart zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive

Table 19.19 *continued*

Option	Description
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default to moving range chart
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of individual measurements chart
VAXIS2=	Specifies major tick mark values for vertical axis of moving range chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels

Table 19.19 *continued*

Option	Description
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NOCHART	Suppresses creation of charts
NOCHART2	Suppresses creation of moving range chart
NOFRAME	Suppresses frame for plot area
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	Displays individual measurements and moving range charts on separate screens or pages
TOTPANELS=	Specifies number of pages or screens to be used to display chart
YPCT1=	Specifies length of vertical axis on individual measurements chart as a percentage of sum of lengths of vertical axes for individual measurements and moving range charts
ZEROSTD	Displays individual measurements chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on moving range chart

Table 19.19 *continued*

Option	Description
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on moving range chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on individual measurements chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on moving range chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to individual measurements chart

Table 19.19 *continued*

Option	Description
ANNOTATE2=	Specifies annotate data set that adds features to moving range chart
DESCRIPTION=	Specifies description of individual measurements chart's GRSEG catalog entry
DESCRIPTION2=	Specifies description of moving range chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
LTMARGIN=	Specifies width of left margin area for plot requested with LTMPLOT= option
LTMPLOT=	Requests univariate plot in left margin
NAME=	Specifies name of individual measurements chart's GRSEG catalog entry
NAME2=	Specifies name of moving range chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
RTMARGIN=	Specifies width of right margin area for plot requested with LTMPLOT= option
RTMPLOT=	Requests univariate plot in right margin
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers in the individual measurements chart
MARKERDISPLAY2=	Specifies a subset of subgroups to be plotted with markers in the moving range chart

Table 19.19 *continued*

Option	Description
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers in the individual measurements chart
MARKERLABEL2=	Specifies labels for subgroups that are plotted with markers in the moving range chart
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed

Table 19.19 *continued*

Option	Description
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 19.19 *continued*

Option	Description
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on individual measurements chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots

Table 19.19 *continued*

Option	Description
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive

Table 19.19 *continued*

Option	Description
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: IRCHART Statement

The following sections provide details that are specific to the IRCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Individual Measurements and Moving Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
X_i	the ith individual measurement
$\bar{X}$	mean of the individual measurements, computed as $(X_1 + \cdots + X_N)/N$, where N is the number of individual measurements
n	number of consecutive measurements used to calculate the moving ranges (by default, $n = 2$)
R_i	moving range computed for the ith subgroup (corresponding to the ith individual measurement). If $i < n$, then R_i is assigned a missing value. Otherwise, $$R_i = \max(X_i, X_{i-1}, \ldots, X_{i-n+1}) - \min(X_i, X_{i-1}, \ldots, X_{i-n+1})$$ This formula assumes that $X_i, X_{i-1}, \ldots, X_{i-n+1}$ are nonmissing.
$\bar{R}$	average of the nonmissing moving ranges, computed as $$\frac{R_n + R_{n+1} \cdots + R_N}{N + 1 - n}$$
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	$100p$th percentile $(0 < p < 1)$ of the standard normal distribution
$D_p(n)$	$100p$th percentile $(0 < p < 1)$ of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on an individual measurements chart, indicates the value of a measurement (X_i).

Each point on a moving range chart indicates the value of a moving range (R_i). With $n = 2$, for example, if the first three measurements are 3.4, 3.7, and 3.6, the first moving range is missing, the second moving range is $|3.7 - 3.4| = 0.3$, and the third moving range is $|3.6 - 3.7| = 0.1$.

Central Lines

By default, the central line on an individual measurements chart indicates an estimate for μ, which is computed as $\bar{X}$. If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

The central line on a moving range chart indicates an estimate for the expected moving range, computed as $d_2(n)\hat{\sigma}$ where $\hat{\sigma} = \bar{R}/d_2(n)$. If you specify a known value ($\hat{\sigma}_0$) for σ, the central line indicates the value of $d_2(n)\sigma_0$.

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard errors of X_i and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that X_i or R_i exceeds the limits

The following table provides the formulas for the limits:

Table 19.21 Limits for Individual Measurements and Moving Range Charts

Control Limits	
Individual Measurements Chart	LCL = lower control limit = $\bar{X} - k\hat{\sigma}$
	UCL = upper control limit = $\bar{X} + k\hat{\sigma}$
Moving Range Chart	LCL = lower control limit = $\max(d_2(n)\hat{\sigma} - kd_3(n)\hat{\sigma}, 0)$
	UCL = upper control limit = $d_2(n)\hat{\sigma} + kd_3(n)\hat{\sigma}$
Probability Limits	
Individual Measurements Chart	LCL = lower control limit = $\bar{X} - z_{\alpha/2}\hat{\sigma}$
	UCL = upper control limit = $\bar{X} + z_{\alpha/2}\hat{\sigma}$
Moving Range Chart	LCL = lower control limit = $D_{\alpha/2}(n)\hat{\sigma}$
	UCL = upper control limit = $D_{1-\alpha/2}(n)\hat{\sigma}$

The formulas assume that the measurements are normally distributed. Note that the probability limits for the moving range are asymmetric about the central line. If standard values μ_0 and σ_0 are available for μ and σ, replace $\bar{X}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 19.21.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify n with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 19.22 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_CP_	Capability index C_p
CPK	Capability index C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLI	Lower control limit for individual measurements
LCLR	Lower control limit for moving ranges
LIMITN	Number of consecutive measurements used to compute moving ranges
LSL	Lower specification limit
MEAN	Process mean
_R_	Value of central line on moving range chart
SIGMAS	Multiple (k) of standard error of individual measurement or moving range
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the IRCHART statement
TARGET	Target value
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLI	Upper control limit for individual measurements
UCLR	Upper control limit for moving ranges range
USL	Upper specification limit
VAR	*Process* specified in the IRCHART statement

Notes:

1. If the limits are defined in terms of a multiple k of the standard errors of X_i and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.

2. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1-\alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.

3. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the IRCHART statement. For an example, see "Saving Control Limits" on page 1523.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves individual measurements and moving ranges. The following variables are saved:

- the *subgroup-variable*
- an individual measurements variable named by *process*
- a moving range variable named by *process* suffixed with *R*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

A variable containing the moving ranges is created for each *process* specified in the IRCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   irchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, Width, WidthR, Diameter, and DiameterR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Individual Measurements and Moving Ranges" on page 1521.

OUTTABLE= Data Set

The OUTTABLE= data set saves individual measurements, moving ranges, control limits, and related information. Table 19.23 lists the variables that are saved.

Table 19.23 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on individual measurements chart
EXLIMR	Control limit exceeded on moving range chart
LCLI	Lower control limit for individual measurements
LCLR	Lower control limit for moving range
LIMITN	Number of consecutive measurements used to compute moving ranges
MEAN	Process mean
_R_	Average range
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBI	Individual measurement
SUBR	Moving range
TESTS	Tests for special causes signaled on individual measurements chart
UCLI	Upper control limit for individual measurements
UCLR	Upper control limit for moving range
VAR	*Process* specified in the IRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variables _EXLIM_, _EXLIMR_, and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1523.

Input Data Sets

DATA= Data Set

You can read individual measurements from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the IRCHART statement must be a SAS variable in the data set. This variable provides measurements of items indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the IRCHART statement, must also be a SAS variable in the data set. Each observation in a DATA= data set must contain a measurement for each *process* and a value for the *subgroup-variable*. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option in the IRCHART statement (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating Individual Measurements and Moving Range Charts" on page 1519.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=info limits=Conlims;
   irchart Weight*ID;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 19.21. The LIMITS= data set can also be created directly using a DATA step.

When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLI_, _MEAN_, _UCLI_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly

- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 19.21

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'. See Example 19.12 for an illustration.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1526.

HISTORY= Data Set

You can read individual measurements and moving ranges from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure.

A HISTORY= data set used with the IRCHART statement must contain the following:

- the *subgroup-variable*
- an individual measurements variable for each *process*
- a moving range variable for each *process*

The name of the individual measurements variable must be the *process* specified in the IRCHART statement. The name of the moving range variable must be the prefix *process* concatenated with the special suffix character *R*. For example, consider the following statements:

```
proc shewhart history=Summary;
   irchart (Weight Yieldstrength) * ID;
run;
```

The data set Summary must include the variables ID, Weight, WeightR, YieldstrengthN, and YieldstrengthR.

Note that if you specify a *process* name that contains 32 characters, the name of the moving range variable must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with *R*.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Reading Individual Measurements and Moving Ranges" on page 1522.

TABLE= Data Set

You can read individual measurements, moving ranges, and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.24 lists the variables required in a TABLE= data set used with the IRCHART statement.

Table 19.24 Variables Required in a TABLE= Data Set

Variable	Description
LCLI	Lower control limit for individual measurements
LCLR	Lower control limit for moving range
LIMITN	Number of consecutive measurements used to calculate moving ranges
MEAN	Process mean
_R_	Average moving range
Subgroup-variable	Values of the *subgroup-variable*
SUBI	Individual measurements
SUBR	Moving ranges
UCLI	Upper control limit for individual measurements
UCLR	Upper control limit for moving range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1523.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MAD and MMR) are available for estimating the process standard deviation σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive individual measurements used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the observations are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is described in the American Society for Testing and Materials (1976).

MAD Method

If you specify SMETHOD=MAD, a median absolute deviation estimator is computed for σ, as described by Boyles (1997). It is computed as

$$\hat{\sigma} = \text{median}\{|X_i - \tilde{X}|, 1 \leq i \leq N\}/0.6745$$

where $\tilde{X}$ is the sample median.

MMR Method

If you specify SMETHOD=MMR, a median moving range estimator is computed for σ. This estimator is described by Boyles (1997). It is computed as

$$\hat{\sigma} = \tilde{R}/0.954$$

where $\tilde{R}$ is the median of the nonmissing moving ranges.

Interpreting Charts for Individual Measurements and Moving Ranges

Montgomery (1996) points out that a moving range chart should be interpreted with care because "the moving ranges are correlated, and this correlation may often induce a pattern or runs or cycles on the chart." For this reason Nelson (1982) recommends against plotting the moving ranges. Nelson notes that the assumption of normality is more critical for an individual measurements chart than for an $\bar{X}$ chart. You can use the

NOCHART2 option in the IRCHART statement to specify that only the individual measurements chart is to be displayed. See Example 19.13 for an illustration. If, instead, you specify the SEPARATE option, the charts for individual measurements and moving ranges are displayed on separate screens.

An alternative method for creating an individual measurements chart is to use the XCHART statement, which uses an estimate of σ based on moving ranges of two consecutive measurements when the subgroup sample sizes are all equal to one. Note that the XCHART statement displays the control limit legend $n = 1$ to indicate the common subgroup sample size, whereas the IRCHART statement displays a legend that indicates the number of consecutive measurements used to compute the moving ranges (the "pseudo subgroup sample size").

Nelson (1982) explains that the reason for estimating the process standard deviation σ from moving ranges of two consecutive measurements rather than the sample standard deviation of the measurements is that "the moving range of two minimizes inflationary effects on the variability which are caused by trends and oscillations that may be present." Nelson suggests that any moving range that exceeds 3.5 times the average moving range should be removed from the calculation of the average moving range.

Examples: IRCHART Statement

This section provides advanced examples of the IRCHART statement.

Example 19.11: Applying Tests for Special Causes

NOTE: See *IRCHART with Tests for Special Causes* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make an individual measurements chart more sensitive to special causes of variation. The following statements create a data set named Engines, which contains the weights for 25 jet engines:

```
data Engines;
   input ID Weight @@;
   label Weight='Engine Weight (lbs)'
         ID    ='Engine ID Number';
   datalines;
1711 1270   1712 1258   1713 1248   1714 1260
1715 1263   1716 1260   1717 1259   1718 1240
1719 1260   1720 1246   1721 1238   1722 1253
1723 1249   1724 1245   1725 1251   1726 1252
1727 1249   1728 1274   1729 1258   1730 1268
1731 1248   1732 1295   1733 1243   1734 1253
1735 1258
;
```

Individual measurements and moving range charts are used to monitor the weights. The following statements produce the tables shown in Output 19.11.1 and create the charts shown in Output 19.11.2:

```
title 'Tests for Special Causes Applied to Jet Engine Weights';
ods graphics on;
proc shewhart data=Engines;
   irchart Weight*ID /
       tests    = 1 to 8
       test2run = 7
       odstitle = title
       tabletest
       zonelabels
       markers;
run;
```

The TESTS= option applies eight tests for special causes, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. The TEST2RUN= option specifies the length of the pattern for Test 2. The TABLETESTS option requests a table of individual measurements, moving ranges, and control limits, and it adds a column indicating which measurements tested positive for special causes.

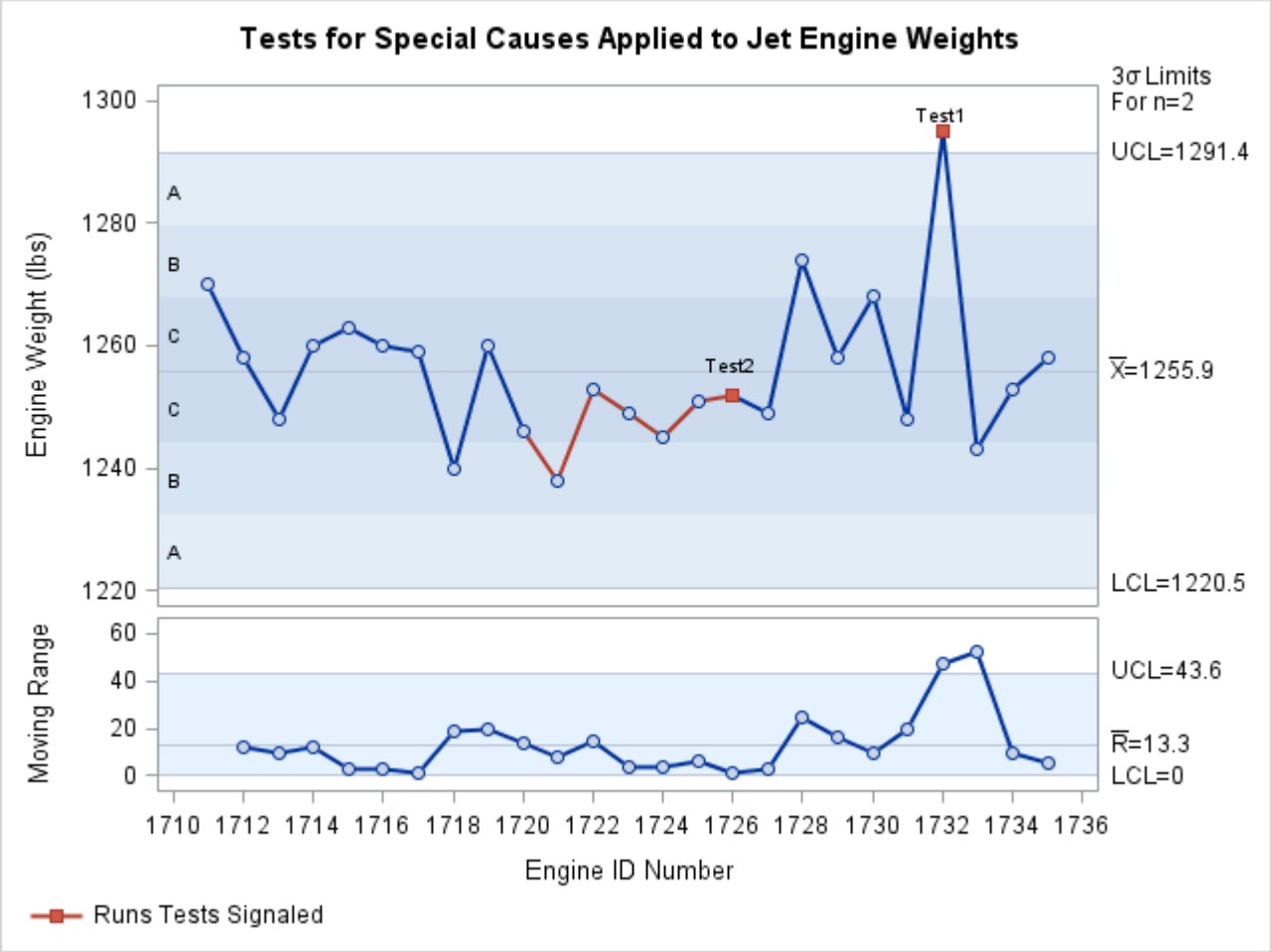
The ZONELABELS option displays zone lines and zone labels on the individual measurements chart. The zones are used to define the tests.

Output 19.11.1 Tabular Form of Individual Measurements and Moving Range Chart

Individual Measurements Chart Summary for Weight

	3 Sigma Limits with n=2 for Weight				3 Sigma Limits with n=2 for Moving Range		
ID	Lower Limit	Weight	Upper Limit	Special Tests Signaled	Lower Limit	Moving Range	Upper Limit
1711	1220.4709	1270.0000	1291.3691		0	.	43.553759
1712	1220.4709	1258.0000	1291.3691		0	12.000000	43.553759
1713	1220.4709	1248.0000	1291.3691		0	10.000000	43.553759
1714	1220.4709	1260.0000	1291.3691		0	12.000000	43.553759
1715	1220.4709	1263.0000	1291.3691		0	3.000000	43.553759
1716	1220.4709	1260.0000	1291.3691		0	3.000000	43.553759
1717	1220.4709	1259.0000	1291.3691		0	1.000000	43.553759
1718	1220.4709	1240.0000	1291.3691		0	19.000000	43.553759
1719	1220.4709	1260.0000	1291.3691		0	20.000000	43.553759
1720	1220.4709	1246.0000	1291.3691		0	14.000000	43.553759
1721	1220.4709	1238.0000	1291.3691		0	8.000000	43.553759
1722	1220.4709	1253.0000	1291.3691		0	15.000000	43.553759
1723	1220.4709	1249.0000	1291.3691		0	4.000000	43.553759
1724	1220.4709	1245.0000	1291.3691		0	4.000000	43.553759
1725	1220.4709	1251.0000	1291.3691		0	6.000000	43.553759
1726	1220.4709	1252.0000	1291.3691	2	0	1.000000	43.553759
1727	1220.4709	1249.0000	1291.3691		0	3.000000	43.553759
1728	1220.4709	1274.0000	1291.3691		0	25.000000	43.553759
1729	1220.4709	1258.0000	1291.3691		0	16.000000	43.553759
1730	1220.4709	1268.0000	1291.3691		0	10.000000	43.553759
1731	1220.4709	1248.0000	1291.3691		0	20.000000	43.553759
1732	1220.4709	1295.0000	1291.3691	1	0	47.000000	43.553759
1733	1220.4709	1243.0000	1291.3691		0	52.000000	43.553759
1734	1220.4709	1253.0000	1291.3691		0	10.000000	43.553759
1735	1220.4709	1258.0000	1291.3691		0	5.000000	43.553759

Output 19.11.2 Tests for Special Causes



Output 19.11.1 and Output 19.11.2 indicate that Test 1 was positive for engine 1732 and Test 2 was positive for engine 1726. Test 1 detects one point beyond Zone A (outside the control limits) and Test 2 detects seven points (TEST2RUN=7) in a row on one side of the central line.

Example 19.12: Specifying Standard Values for the Process Mean and Standard Deviation

NOTE: See *Specifying Known Values for IRCHART* in the SAS/QC Sample Library.

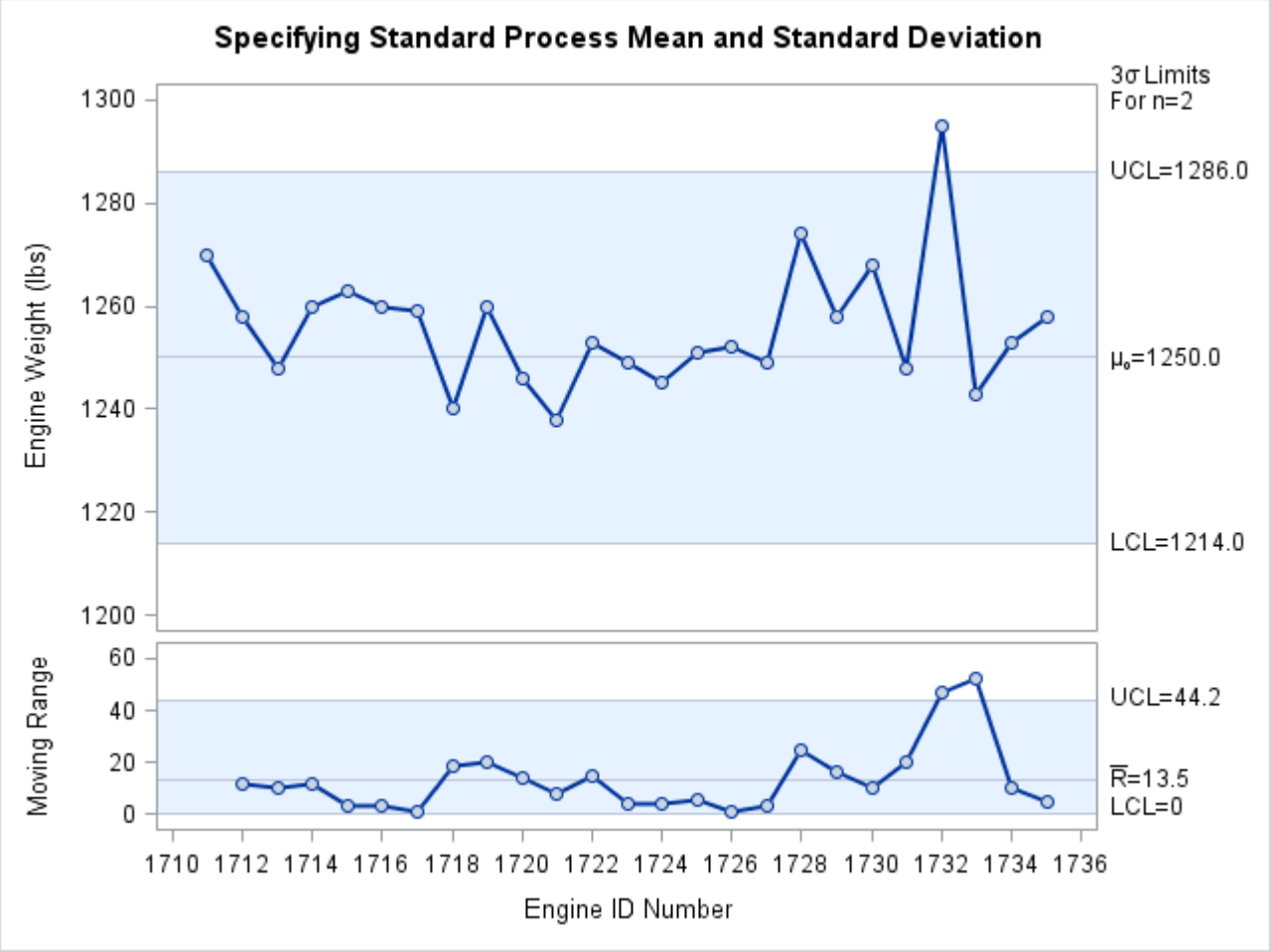
By default, the IRCHART statement estimates the process mean (μ) and standard deviation (σ) from the data, as in the previous example. However, there are applications in which known (standard) values μ_0 and σ_0 are available for these parameters based on previous experience or extensive sampling.

For example, suppose that the manufacturing process described in the previous example produces engines whose weights are normally distributed with a mean of 1250 and a standard deviation of 12. The following statements create individual measurements and moving range charts based on these values:

```
ods graphics on;
title 'Specifying Standard Process Mean and Standard Deviation';
proc shewhart data=Engines;
   irchart Weight*ID /
      odstitle = title
      mu0      = 1250
      sigma0   = 12
      xsymbol  = mu0
      markers;
run;
```

The charts are shown in Output 19.12.1. The MU0= option and SIGMA0= option specify μ_0 and σ_0. The XSYMBOL= option specifies the label for the central line on the individual measurements chart, and the keyword MU0 requests a label indicating that the central line is based on a standard value.

Output 19.12.1 Specifying Standard Values with MU0= and SIGMA0=



You can also specify μ_0 and σ_0 as the values of the variables _MEAN_ and _STDDEV_ in a LIMITS= data set. For example, the following statements create a LIMITS= data set with the standard values specified in the preceding IRCHART statement:

```
data Enginelimits;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Weight';
   _subgrp_ = 'id';
   _limitn_ = 2;
   _type_   = 'STANDARD';
   _mean_   = 1250;
   _stddev_ = 12;
run;
```

The variables _VAR_ and _SUBGRP_ are required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the IRCHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values. See "LIMITS= Data Set" on page 1547 for details.

The following statements read Enginelimits as a LIMITS= data set:

```
proc shewhart data=Engines limits=Enginelimits;
   irchart Weight*ID / xsymbol=mu0;
run;
```

The resulting charts (not shown here) are identical to those shown in Output 19.12.1.

Example 19.13: Displaying Distributional Plots in the Margin

NOTE: See *IRCHARTS with Margin Plots* in the SAS/QC Sample Library.

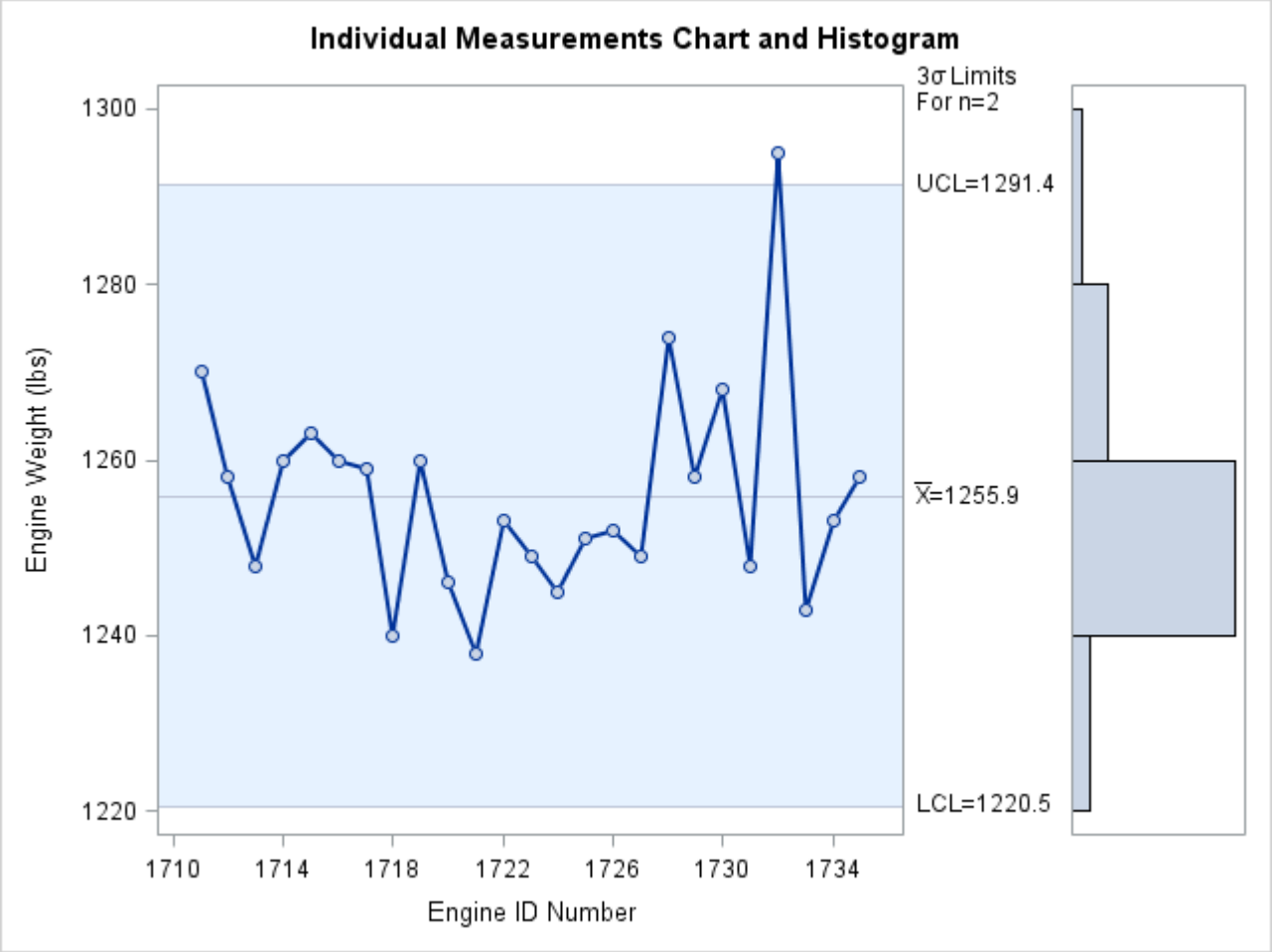
You can augment a chart for individual measurements with one of several graphical displays, such as a histogram or a box-and-whisker plot. These displays summarize the measurements plotted on the chart, and, if the process is in statistical control, they provide a view of the process distribution.

For example, the following statements create an individual measurements chart for the engine weight measurements in the data set Engines (see Example 19.11) augmented with a histogram of the weights:

```
ods graphics on;
title 'Individual Measurements Chart and Histogram';
proc shewhart data=Engines;
   irchart Weight*ID /
      odstitle = title
      rtmplot  = histogram
      markers
      nochart2;
run;
```

The chart is shown in Output 19.13.1. The RTMPLOT= option requests a histogram in the right margin. The NOCHART2 option suppresses the display of the moving range chart.

Output 19.13.1 Histogram in Right Margin



The following *keywords*, requesting different types of plots, are available with the RTMPLOT= option:

Keyword	Marginal Plot
HISTOGRAM	Histogram
DIGIDOT	Digidot plot
SKELETAL	Skeletal box-and-whisker plot
SCHEMATIC	Schematic box-and-whisker plot
SCHEMATICID	Schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	Schematic box-and-whisker plot with far outliers labeled

See the entry for the BOXSTYLE= option in "Dictionary of Options: SHEWHART Procedure" on page 1993 for a description of the various box-and-whisker plots.

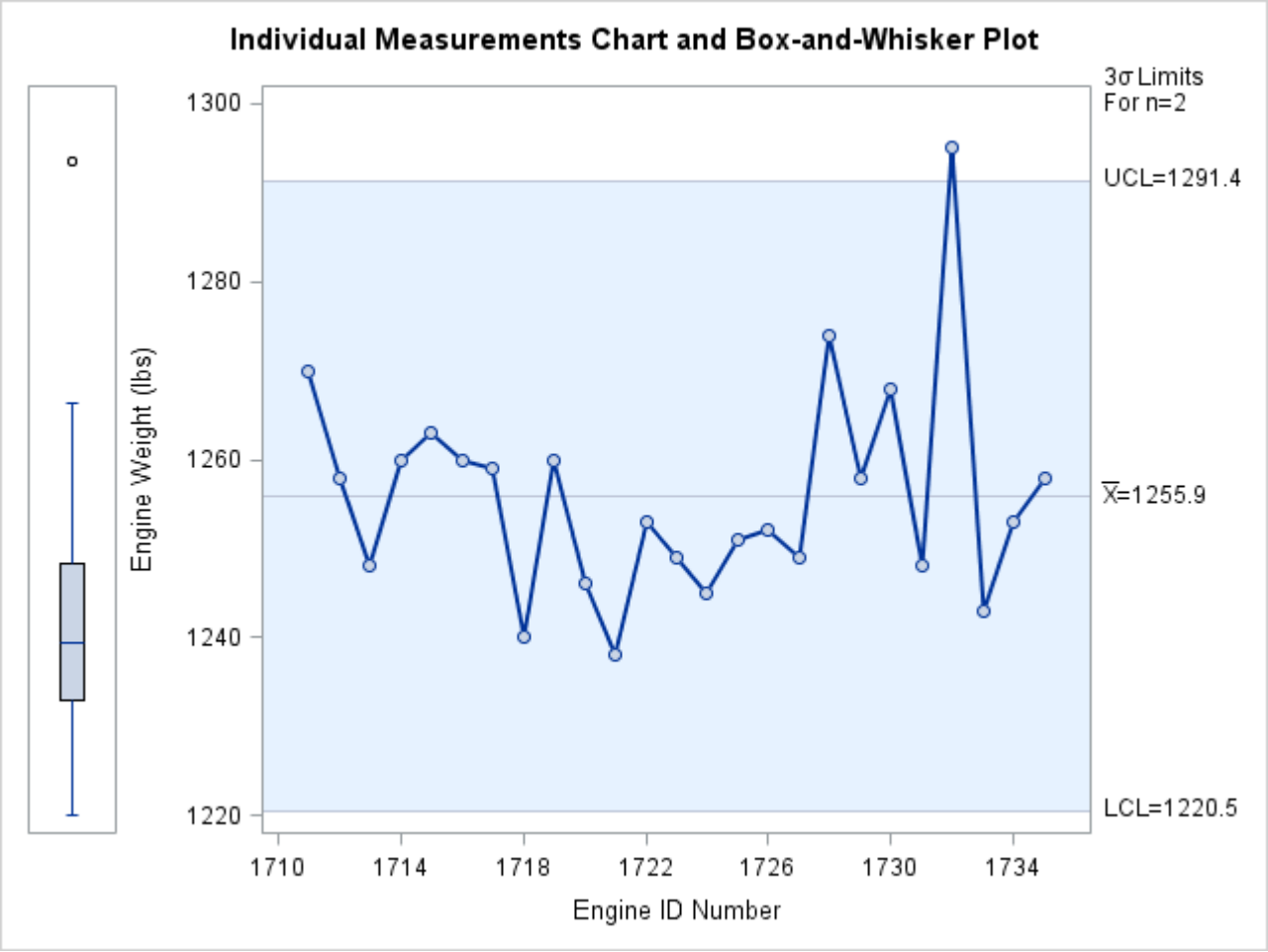
You can also use the LTMPLOT= option to request univariate plots in the left margin. The following statements request an individual measurements chart with a box-and-whisker plot in the left margin:

```
title 'Individual Measurements Chart and Box-and-Whisker Plot';
proc shewhart data=Engines;
   irchart Weight*ID /
      odstitle = title
```

```
        ltmplot  = schematic
        ltmargin = 8
        markers
        nochart2;
run;
```

The chart is shown in Output 19.13.2. The same *keywords* that are available with the RTMPLOT= option can be specified with the LTMPLOT= option. The LTMARGIN= option specifies the width (in horizontal percent screen units) of the left margin.

Output 19.13.2 Box-and-Whisker Plot in Left Margin



MCHART Statement: SHEWHART Procedure

Overview: MCHART Statement

The MCHART statement creates a chart for subgroup medians, which is used to monitor the central tendency of a process.

You can use options in the MCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted medians or as probability limits
- tabulate subgroup sample sizes, subgroup medians, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and subgroup medians in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 2100)
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing medians charts with the MCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.

- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: When analyzing variables data, you should examine the variability of the process as well as the mean level. You can use the MRCHART statement in the SHEWHART procedure to monitor both the mean level and variability.

Getting Started: MCHART Statement

This section introduces the MCHART statement with simple examples that illustrate commonly used options. Complete syntax for the MCHART statement is presented in the section "Syntax: MCHART Statement" on page 1573.

Creating Charts for Medians from Raw Data

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

A consumer products company weighs detergent boxes (in pounds) to determine whether the fill process is in control. The following statements create a SAS data set named Detergent, which contains the weights for five boxes in each of 28 lots. A lot is considered a rational subgroup.

```
data Detergent;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
 1 17.39 26.93 19.34 22.56 24.49
 2 23.63 23.57 23.54 20.56 22.17
 3 24.35 24.58 23.79 26.20 21.55
 4 25.52 28.02 28.44 25.07 23.39
 5 23.25 21.76 29.80 23.09 23.70
 6 23.01 22.67 24.70 20.02 26.35
 7 23.86 24.19 24.61 26.05 24.18
 8 26.00 26.82 28.03 26.27 25.85
 9 21.58 22.31 25.03 20.86 26.94
10 22.64 21.05 22.66 29.26 25.02
11 26.38 27.50 23.91 26.80 22.53
12 23.01 23.71 25.26 20.21 22.38
13 23.15 23.53 22.98 21.62 26.99
14 26.83 23.14 24.73 24.57 28.09
15 26.15 26.13 20.57 25.86 24.70
16 25.81 23.22 23.99 23.91 27.57
17 25.53 22.87 25.22 24.30 20.29
18 24.88 24.15 25.29 29.02 24.46
```

```
19 22.32 25.96 29.54 25.92 23.44
20 25.63 26.83 20.95 24.80 27.25
21 21.68 21.11 26.07 25.17 27.63
22 26.72 27.05 24.90 30.08 25.22
23 31.58 22.41 23.67 23.47 24.90
24 28.06 23.44 24.92 24.64 27.42
25 21.10 22.34 24.96 26.50 24.51
26 23.80 24.03 24.75 24.82 27.21
27 25.10 26.09 27.21 24.28 22.45
28 25.53 22.79 26.26 25.85 25.64
;
```

A partial listing of Detergent is shown in Figure 19.34.

Figure 19.34 Partial Listing of the Data Set Detergent

The Data Set DETERGENT

Lot	Weight
1	17.39
1	26.93
1	19.34
1	22.56
1	24.49
2	23.63
2	23.57
2	23.54
2	20.56
2	22.17
3	24.35
3	24.58
3	23.79
3	26.20
3	21.55
4	25.52

The data set Detergent is said to be in "strung-out" form, because each observation contains the lot number and weight of a single box. The first five observations contain the weights for the first lot, the second five observations contain the weights for the second lot, and so on. Because the variable Lot classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Weight contains the weights and is referred to as the *process variable* (or *process* for short).

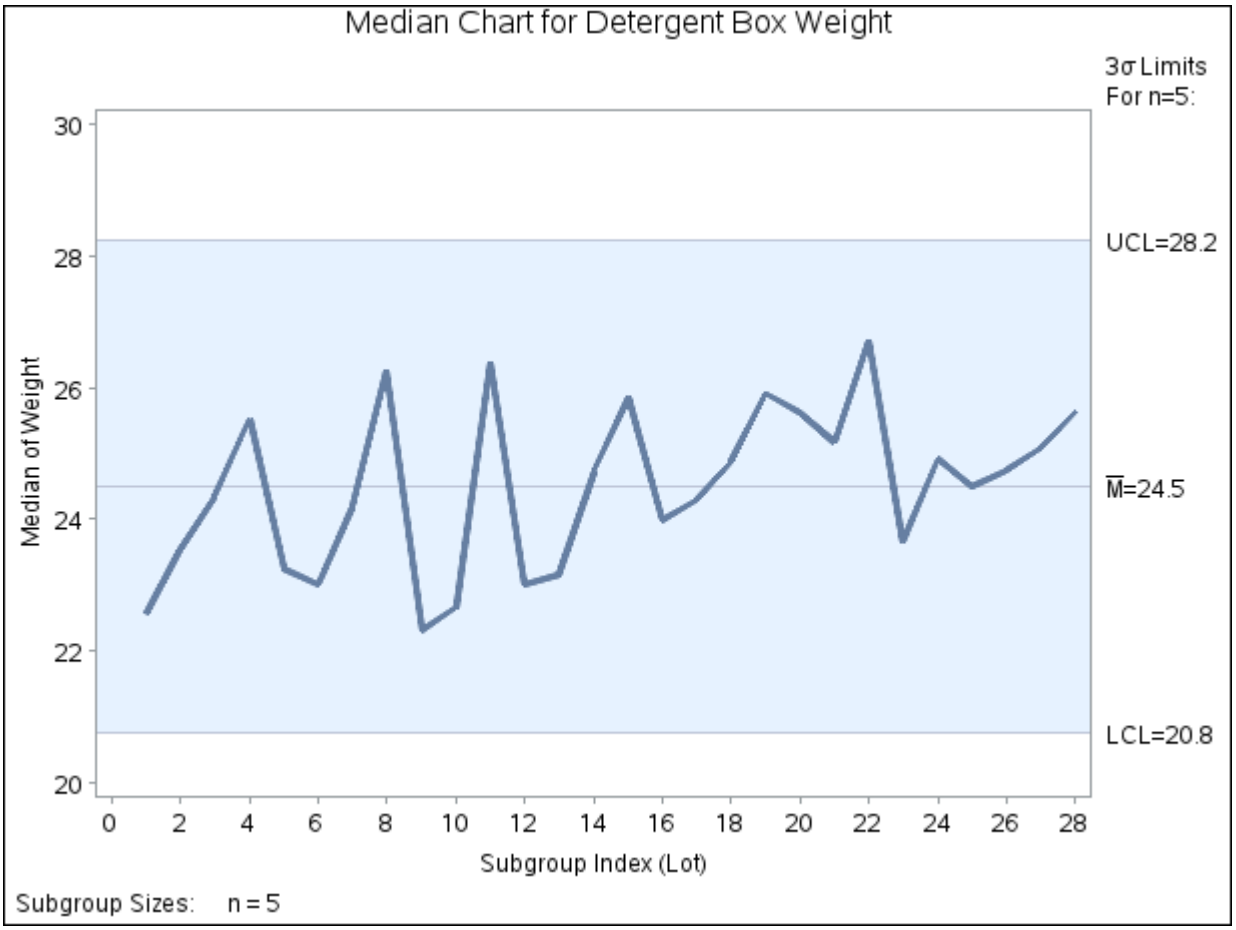
The within-subgroup variability of the weights is known to be stable. You can use a median chart to determine whether the mean level of the weights is in control. The following statements create the median chart shown in Figure 19.35:

```
ods graphics off;
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot;
run;
```

This example illustrates the basic form of the MCHART statement. After the keyword MCHART, you specify the *process* to analyze (in this case, Weight) followed by an asterisk and the *subgroup-variable* (Lot).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 19.35 Median Chart for Detergent Box Weight Data (Traditional Graphics)



Each point on the chart represents the median of the weights for a particular lot. For instance, the weights for the first lot are 17.39, 19.34, 22.56, 24.49, and 26.93, and consequently, the median plotted for this lot is 22.56.

Because all of the subgroup medians lie within the control limits, you can conclude that the process is in statistical control. By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.27. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1571.

For computational details, see "Constructing Median Charts" on page 1585. For more details on reading raw measurements, see "DATA= Data Set" on page 1591.

Creating Charts for Medians from Subgroup Summary Data

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create median charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the MCHART statement with data of this type.

The following data set (Detsum) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Lot). The variable WeightM contains the subgroup medians, the variable WeightR contains the subgroup ranges, and the variable WeightN contains the subgroup sample sizes (these are all five).

```
data Detsum;
   input Lot WeightM WeightR;
   WeightN = 5;
   datalines;
 1  22.56  9.54
 2  23.54  3.07
 3  24.35  4.65
 4  25.52  5.05
 5  23.25  8.04
 6  23.01  6.33
 7  24.19  2.19
 8  26.27  2.18
 9  22.31  6.08
10  22.66  8.21
11  26.38  4.97
12  23.01  5.05
13  23.15  5.37
14  24.73  4.95
15  25.86  5.58
16  23.99  4.35
17  24.30  5.24
18  24.88  4.87
19  25.92  7.22
20  25.63  6.30
21  25.17  6.52
22  26.72  5.18
23  23.67  9.17
24  24.92  4.62
25  24.51  5.40
26  24.75  3.41
27  25.10  4.76
28  25.64  3.47
;
```

A partial listing of Detsum is shown in Figure 19.36.

Figure 19.36 The Summary Data Set Detsum

Summary Data Set for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

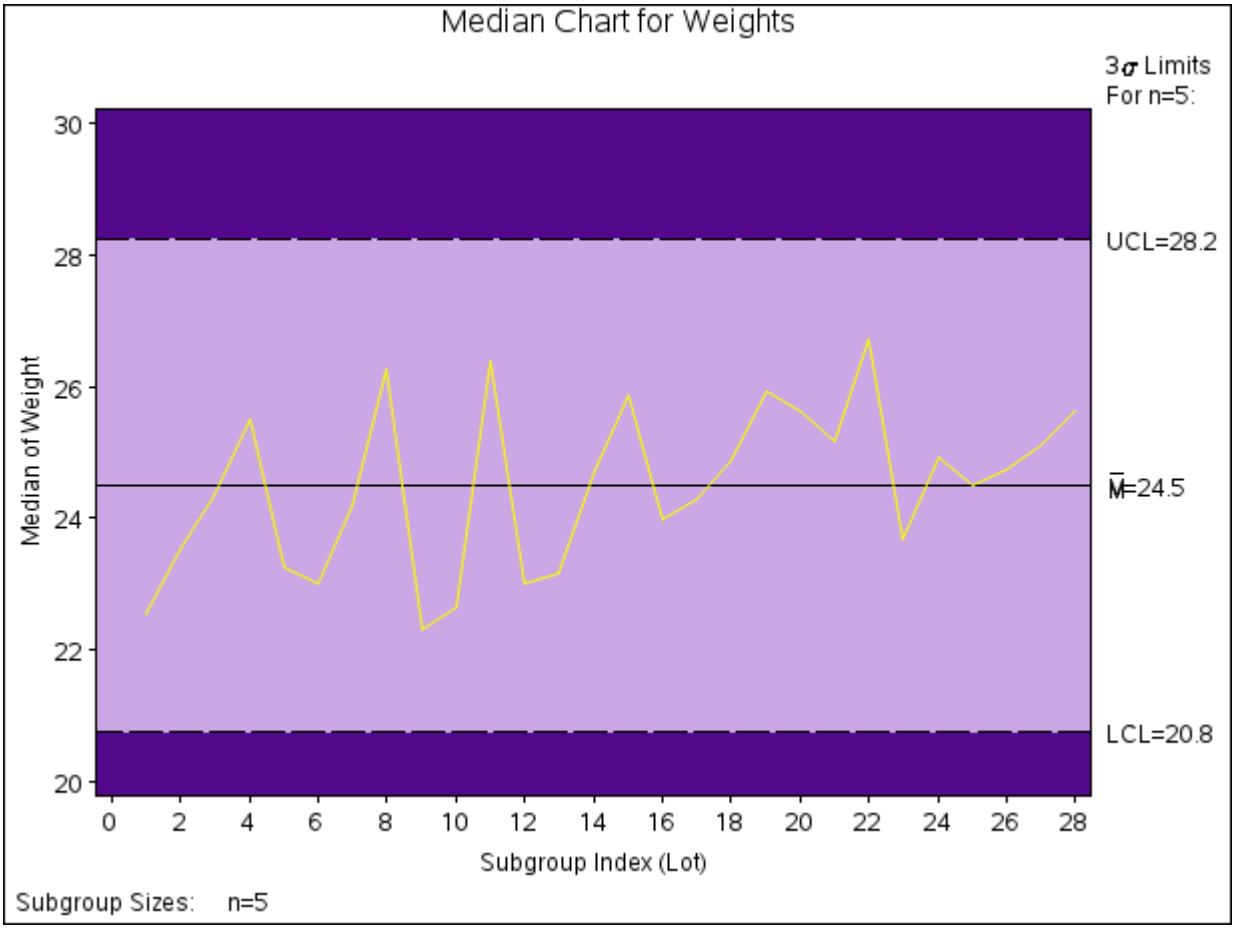
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
title 'Median Chart for Weights';
proc shewhart history=Detsum;
   mchart Weight*Lot / cframe  = viv
                       cinfill = vpav
                       cconnect = yellow;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the MCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting median chart is shown in Figure 19.37.

Note that Weight is *not* the name of a SAS variable in the data set Detsum but is, instead, the common prefix for the names of the three SAS variables WeightM, WeightR, and WeightN. The suffix characters *M*, *R*, and *N* indicate *median*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (Weight), which is referred to as the *process*. The name Lot specified after the asterisk is the name of the *subgroup-variable*.

Figure 19.37 Median Chart from Summary Data Set Detsum (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the MCHART statement must contain the following variables:

- subgroup variable
- subgroup median variable
- either a subgroup range variable or a subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup median, range (or standard deviation), and sample size variables must begin with the *process* name specified in the MCHART statement and end with the special suffix characters *M*, *R* (or *S*), and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Medians and Ranges from Summary Data" on page 1606).

If you specify the STDDEVIATIONS option in the MCHART statement, the HISTORY= data set must contain a subgroup standard deviation variable; otherwise, the HISTORY= data set must contain a subgroup range variable. The STDDEVIATIONS option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations rather than subgroup ranges. For example, in the following statements, the data set Detsum2 must contain a subgroup standard deviation variable named WeightS:

```
title 'Median Chart for Weights';
symbol v=dot;
proc shewhart history=Detsum2;
   mchart Weight*Lot / stddeviations;
run;
```

Options such as STDDEVIATIONS are specified after the slash (/) in the MCHART statement. A complete list of options is presented in the section "Syntax: MCHART Statement" on page 1573.

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1592.

Saving Summary Statistics

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

In this example, the MCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Detergent and create a summary data set named Dethist:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outhistory = Dethist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.35. Figure 19.38 contains a partial listing of Dethist.

Figure 19.38 The Summary Data Set Dethist

Summary Data Set DETHIST for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

There are four variables in the data set Dethist.

- Lot contains the subgroup index.
- WeightM contains the subgroup medians.
- WeightR contains the subgroup ranges.
- WeightN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *M*, *R*, and *N* to the *process* Weight specified in the MCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the STDDEVIATIONS option, the OUTHISTORY= data set includes a subgroup standard deviation variable instead of a subgroup range variable, as demonstrated by the following statements:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outhistory = Dethist2
                       stddeviations
                       nochart;
run;
```

Figure 19.39 contains a partial listing of Dethist2.

Figure 19.39 The Summary Data Set Dethist2

Summary Data Set DETHIST for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

The variable WeightS, which contains the subgroup standard deviations, is named by adding the suffix character *S* to the *process* Weight.

For more information, see "OUTHISTORY= Data Set" on page 1589.

Saving Control Limits

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a median chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1571) or modify the limits with a DATA step program.

The following statements read measurements from the data set Detergent (see "Creating Charts for Medians from Raw Data" on page 1560) and save the control limits displayed in Figure 19.35 in a data set named Detlim:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outlimits=Detlim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Detlim is listed in Figure 19.40.

Figure 19.40 The Data Set Detlim Containing Control Limit Information

Control Limits for Detergent Box Weights

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_
Weight	Lot	ESTIMATE	5	.002909021	3	20.7554	24.4996	28.2439

LCLR	_R_	_UCLR_	_STDDEV_
0	5.42036	11.4613	2.33041

The data set Detlim contains one observation with the limits for the *process* Weight. The variables _LCLM_ and _UCLM_ contain the lower and upper control limits for the medians, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLR_, _R_, and _UCLR_ are not used to create median charts, but they are included so the data set Detlim can be used to create an *R* chart; see "MRCHART Statement: SHEWHART Procedure" on page 1603 and "RCHART Statement: SHEWHART Procedure" on page 1729. If you specify the STDDEVIATIONS option in the MCHART statement, the variables _LCLS_, _S_, and _UCLS_ are included in the OUTLIMITS= data set. These variables can be used to create an *s* chart; see "SCHART Statement: SHEWHART Procedure" on page 1767. For more information, see "OUTLIMITS= Data Set" on page 1587.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outtable=Dtable
                       nochart;
run;
```

The data set Dtable is listed in Figure 19.41.

Figure 19.41 The Data Set Dtable

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_STDDEV_	_EXLIM_
Weight	1	3	5	5	20.7554	22.56	24.4996	28.2439	2.33041	
Weight	2	3	5	5	20.7554	23.54	24.4996	28.2439	2.33041	
Weight	3	3	5	5	20.7554	24.35	24.4996	28.2439	2.33041	
Weight	4	3	5	5	20.7554	25.52	24.4996	28.2439	2.33041	
Weight	5	3	5	5	20.7554	23.25	24.4996	28.2439	2.33041	
Weight	6	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041	
Weight	7	3	5	5	20.7554	24.19	24.4996	28.2439	2.33041	
Weight	8	3	5	5	20.7554	26.27	24.4996	28.2439	2.33041	
Weight	9	3	5	5	20.7554	22.31	24.4996	28.2439	2.33041	
Weight	10	3	5	5	20.7554	22.66	24.4996	28.2439	2.33041	
Weight	11	3	5	5	20.7554	26.38	24.4996	28.2439	2.33041	
Weight	12	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041	
Weight	13	3	5	5	20.7554	23.15	24.4996	28.2439	2.33041	
Weight	14	3	5	5	20.7554	24.73	24.4996	28.2439	2.33041	
Weight	15	3	5	5	20.7554	25.86	24.4996	28.2439	2.33041	
Weight	16	3	5	5	20.7554	23.99	24.4996	28.2439	2.33041	
Weight	17	3	5	5	20.7554	24.30	24.4996	28.2439	2.33041	
Weight	18	3	5	5	20.7554	24.88	24.4996	28.2439	2.33041	
Weight	19	3	5	5	20.7554	25.92	24.4996	28.2439	2.33041	
Weight	20	3	5	5	20.7554	25.63	24.4996	28.2439	2.33041	
Weight	21	3	5	5	20.7554	25.17	24.4996	28.2439	2.33041	
Weight	22	3	5	5	20.7554	26.72	24.4996	28.2439	2.33041	
Weight	23	3	5	5	20.7554	23.67	24.4996	28.2439	2.33041	
Weight	24	3	5	5	20.7554	24.92	24.4996	28.2439	2.33041	
Weight	25	3	5	5	20.7554	24.51	24.4996	28.2439	2.33041	
Weight	26	3	5	5	20.7554	24.75	24.4996	28.2439	2.33041	
Weight	27	3	5	5	20.7554	25.10	24.4996	28.2439	2.33041	
Weight	28	3	5	5	20.7554	25.64	24.4996	28.2439	2.33041	

This data set contains one observation for each subgroup sample. The variables _SUBMED_ and _SUBN_ contain the subgroup medians and subgroup sample sizes. The variables _LCLM_ and _UCLM_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Lot contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1589.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Dtable and display a median chart (not shown here) identical to the chart in Figure 19.35:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart table=Dtable;
   mchart Weight*Lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1593.

Reading Preestablished Control Limits

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Detlim saved control limits computed from the measurements in Detergent. This example shows how these limits can be applied to new data provided in the following data set:

```
data Detergent2;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
29 16.66 27.49 18.87 22.53 24.72
30 23.74 23.67 23.64 20.26 22.09
31 24.56 24.82 23.92 26.67 21.38
32 25.89 28.73 29.21 25.38 23.47
33 23.32 21.61 30.75 23.13 23.82
34 23.04 22.65 24.96 19.64 26.84
35 24.01 24.38 24.86 26.50 24.37
36 26.43 27.36 28.74 26.74 26.27
37 21.41 22.24 25.34 20.59 27.51
38 22.62 20.81 22.64 30.15 25.32
39 26.86 28.14 24.06 27.35 22.49
40 23.03 23.83 25.59 19.85 22.33
41 23.19 23.63 23.00 21.46 27.57
42 27.38 23.18 24.99 24.81 28.82
43 26.60 26.58 20.26 26.27 24.96
44 26.22 23.28 24.15 24.06 28.23
45 25.90 22.88 25.55 24.50 19.95
46 16.66 27.49 18.87 22.53 24.72
47 23.74 23.67 23.64 20.26 22.09
48 24.56 24.82 23.92 26.67 21.38
49 25.89 28.73 29.21 25.38 23.47
50 23.32 21.61 30.75 23.13 23.82
;
```

The following statements create a median chart for the data in Detergent2 using the control limits in Detlim:

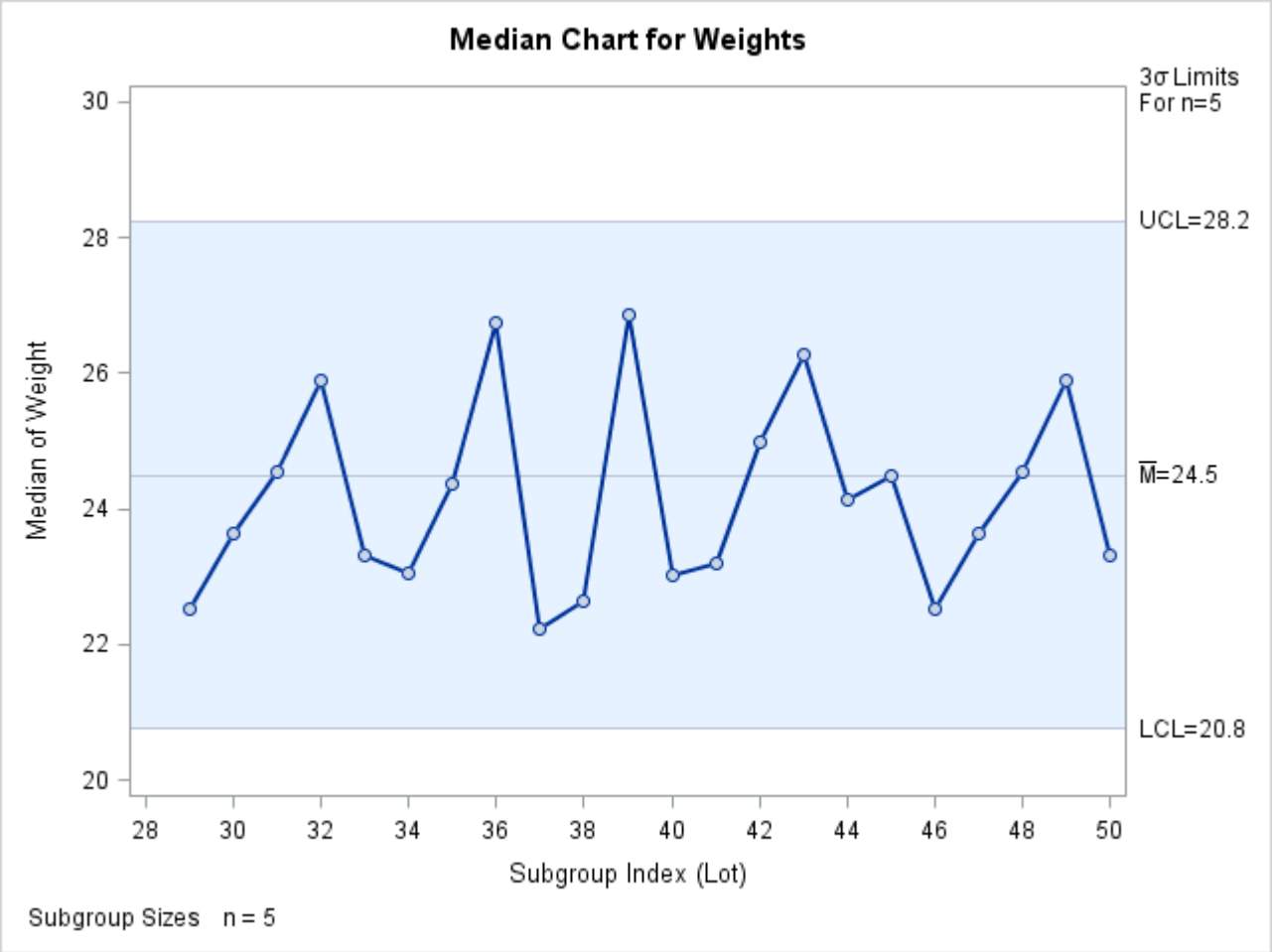
```
ods graphics on;
title 'Median Chart for Weights';
proc shewhart data=Detergent2 limits=Detlim;
   mchart Weight*Lot / odstitle=title markers;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the median chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 19.42.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Weight
- the value of _SUBGRP_ matches the *subgroup-variable* name Lot

Figure 19.42 Median Chart for Second Set of Detergent Box Weight Data (ODS Graphics)



The chart indicates that the process is in control, because all the medians lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1591 for details concerning the variables that you must provide.

Syntax: MCHART Statement

The basic syntax for the MCHART statement is as follows:

MCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

MCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of MCHART statements in the SHEWHART procedure. The components of the MCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Medians from Raw Data" on page 1560.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Medians from Subgroup Summary Data" on page 1563.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1569.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct median charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   mchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding MCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the medians.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create a median chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   mchart Weight*Day='*';
run;
```

options
enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the MCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.25 MCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)

Table 19.25 *continued*

Option	Description
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line
Process Mean and Standard Deviation Options	
MEDCENTRAL=	Specifies method for estimating process mean μ
MU0=	Specifies known value of μ_0 for process mean μ
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
STDDEVIATIONS	Specifies that estimate of process standard deviation σ is to be calculated from subgroup standard deviations
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on median chart
ALLLABEL2=	Labels every point on trend chart
CLABEL=	Specifies color for labels

Table 19.25 *continued*

Option	Description
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
NOTRENDCONNECT	Suppresses line segments that connect points on trend chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines delineating zones A, B, and C

Table 19.25 *continued*

Option	Description
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis

Table 19.25 *continued*

Option	Description
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of median chart
VAXIS2=	Specifies major tick mark values for vertical axis of trend chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel

Table 19.25 *continued*

Option	Description
TOTPANELS=	Specifies number of pages or screens to be used to display chart
TRENDVAR=	Specifies list of trend variables
YPCT1=	Specifies length of vertical axis on median chart as a percentage of sum of lengths of vertical axes for median and trend charts
ZEROSTD	Displays median chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on median chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart

Table 19.25 *continued*

Option	Description
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to median chart
ANNOTATE2=	Specifies annotate data set that adds features to trend chart
DESCRIPTION=	Specifies description of median chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of median chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
WTREND=	Specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors

Table 19.25 *continued*

Option	Description
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart

Table 19.25 *continued*

Option	Description
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend

Table 19.25 *continued*

Option	Description
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on median chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments

Table 19.25 *continued*

Option	Description
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*

Table 19.25 *continued*

Option	Description
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: MCHART Statement

The following sections provide details that are specific to the MCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Median Charts

The following notation is used in this section:

μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	Mean of measurements in ith subgroup
n_i	Sample size of ith subgroup
N	The number of subgroups
x_{ij}	jth measurement in the ith subgroup, $j = 1, 2, 3, \ldots, n_i$
$x_{i(j)}$	jth largest measurement in the ith subgroup. Then $$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$
$\overline{\overline{X}}$	Weighted average of subgroup means
M_i	Median of the measurements in the ith subgroup: $$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$
$\bar{M}$	Average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$

$\tilde{M}$	Median of the subgroup medians. Denote the jth largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. Then $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	Standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	100pth percentile ($0 < p < 1$) of the distribution of the median of n independent observations from a normal population with unit standard deviation
z_p	100pth percentile of the standard normal distribution
$D_p(n)$	100pth percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on a median chart indicates the value of a subgroup median (M_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $M_{10} = 15$.

Central Line

The value of the central line indicates an estimate for μ, which is computed as

- $\bar{M}$ by default
- $\overline{\overline{X}}$ when you specify MEDCENTRAL=AVGMEAN
- $\tilde{M}$ when you specify MEDCENTRAL=MEDMED
- μ_0 when you specify μ_0 with the MU0= option

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard error of M_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that M_i exceeds the limits

The following table provides the formulas for the limits:

Table 19.27 Limits for Median Charts

Control Limits
LCLM = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$ UCLM = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$
Probability Limits
LCLM = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$ UCLM = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$

Note that the limits vary with n_i. In Table 19.27, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN, and replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED. Replace $\bar{M}$ with μ_0 if you specify μ_0 with the MU0= option, and replace $\hat{\sigma}$ with σ_0 if you specify σ_0 with the SIGMA0= option. The formulas assume that the data are normally distributed.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 19.28 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option

Table 19.28 *continued*

Option	Description
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	value of central line on median chart ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of M_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the MCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
UCLS	upper control limit for subgroup standard deviation
USL	upper specification limit
VAR	*process* specified in the MCHART statement

Notes:

1. The variables _LCLS_, _S_, and _UCLS_ are included if you specify the STDDEVIATIONS option; otherwise, the variables _LCLR_, _R_, and _UCLR_ are included. These variables are not used to create median charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XRCHART, XSCHART, and MRCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLM_, _UCLM_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.

3. If the limits are defined in terms of a multiple k of the standard error of M_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _ALPHA_.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = F_{med}^{-1}(1 - \alpha/2, n)$, where $F_{med}^{-1}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MCHART statement. For an example, see "Saving Control Limits" on page 1569.

OUTHISTORY= Data Set

The OUTHISTORY= option saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup median variable named by *process* suffixed with *M*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

A subgroup standard deviation variable is included if you specify the STDDEVIATIONS option; otherwise, a subgroup range variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup summary statistics are created for each *process* specified in the MCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   mchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthM, WidthR, WidthN, DiameterM, DiameterR, and DiameterN. The variables WidthR and DiameterR are included, because the STDDEVIATIONS option is not specified. If you specified the STDDEVIATIONS option, the data set Summary would contain WidthS and DiameterS rather than WidthR and DiameterR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1566.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.29 lists the variables that are saved.

Table 19.29 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on median chart
LCLM	Lower control limit for median
LIMITN	Nominal sample size associated with the control limits
MEAN	Estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBMED	Subgroup median
SUBN	Subgroup sample size
TESTS	Tests for special causes signaled on median chart
UCLM	Upper control limit for median
VAR	*Process* specified in the MCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1569.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the MCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the MCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating Charts for Medians from Raw Data" on page 1560.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   mchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLM_, _MEAN_, and _UCLM_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 19.27

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see the section "Reading Preestablished Control Limits" on page 1571.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the MCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup median variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or a subgroup standard deviation variable for each *process*

The names of the subgroup summary statistics variables must be the *process* name concatenated with the following special suffix characters:

Subgroup Summary Statistic	Suffix Character
Subgroup median	M
Subgroup mean	X
Subgroup sample size	N
Subgroup range	R
Subgroup standard deviation	S

You must provide the subgroup mean variable only if you specify the MEDCENTRAL=AVGMEAN option. If you specify the STDDEVIATIONS option, the subgroup standard deviation variable must be included; otherwise, the subgroup range variable must be included.

For example, consider the following statements:

```
proc shewhart history=Summary;
   mchart (Weight Yieldstrength)*Batch / medcentral=avgmean;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightM, WeightR, WeightN, YieldstrengthX, YieldstrengthM, YieldstrengthR, and YieldstrengthN. If the STDDEVIATIONS option were specified in the preceding MCHART statement, it would be necessary for Summary to include the variables Batch, WeightX, WeightM, WeightS, WeightN, YieldstrengthX, YieldstrengthM, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Medians from Subgroup Summary Data" on page 1563.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.30 lists the variables required in a TABLE= data set used with the MCHART statement.

Table 19.30 Variables Required in a TABLE= Data Set

Variable	Description
LCLM	Lower control limit for median
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
Subgroup-variable	Values of the *subgroup-variable*

Table 19.30 *continued*

Variable	Description
SUBMED	Subgroup median
SUBN	Subgroup sample size
UCLM	Upper control limit for median

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1569.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available with the MCHART statement for estimating the process standard deviation σ. The method used to calculate σ depends on whether you specify the STDDEVIATIONS option in the MCHART statement. If this option is specified, σ is estimated using subgroup standard deviations; otherwise, σ is estimated using subgroup ranges. For further details and formulas, see "Methods for Estimating the Standard Deviation" on page 1871.

Examples: MCHART Statement

This section provides more advanced examples of the MCHART statement.

Example 19.14: Controlling Value of Central Line

You can specify options in the MCHART statement to request one of the following values for the central line on median charts:

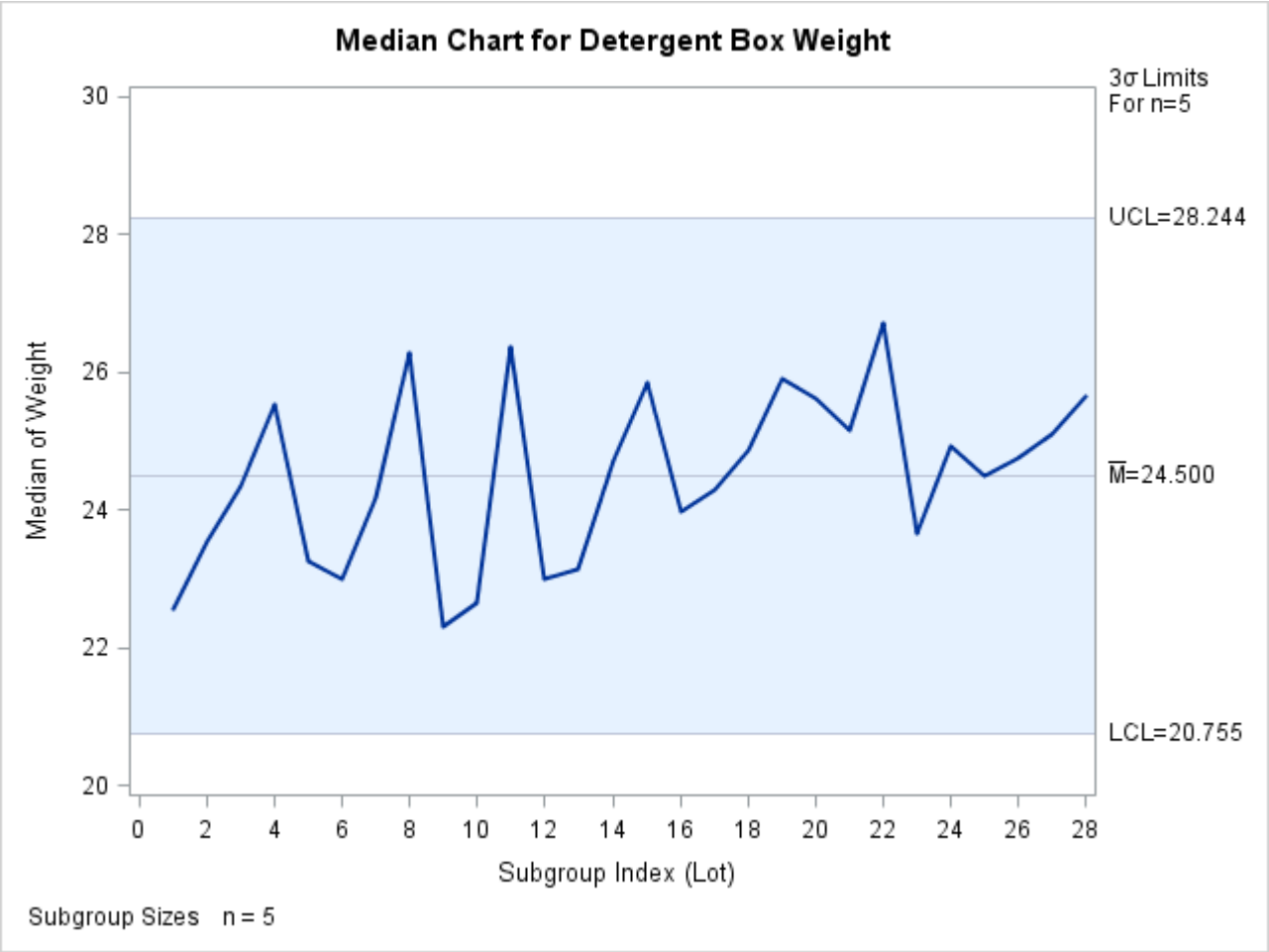
- the average of the subgroup medians
- the average of the subgroup means
- the median of the subgroup medians
- a standard value of the process mean

By default, the value of the central line is the average of the subgroup medians. The following statements create a median chart for the detergent box weights stored in the data set Detergent (see "Creating Charts for Medians from Raw Data" on page 1560) with the average of the subgroup medians as the central line. The resulting chart is shown in Output 19.14.1.

```
ods graphics on;
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal = 3
                       odstitle = title;
run;
```

The NDECIMAL= option specifies the number of decimal digits in the default labels for the control limits and central line.

Output 19.14.1 Central Line is Average of Subgroup Medians

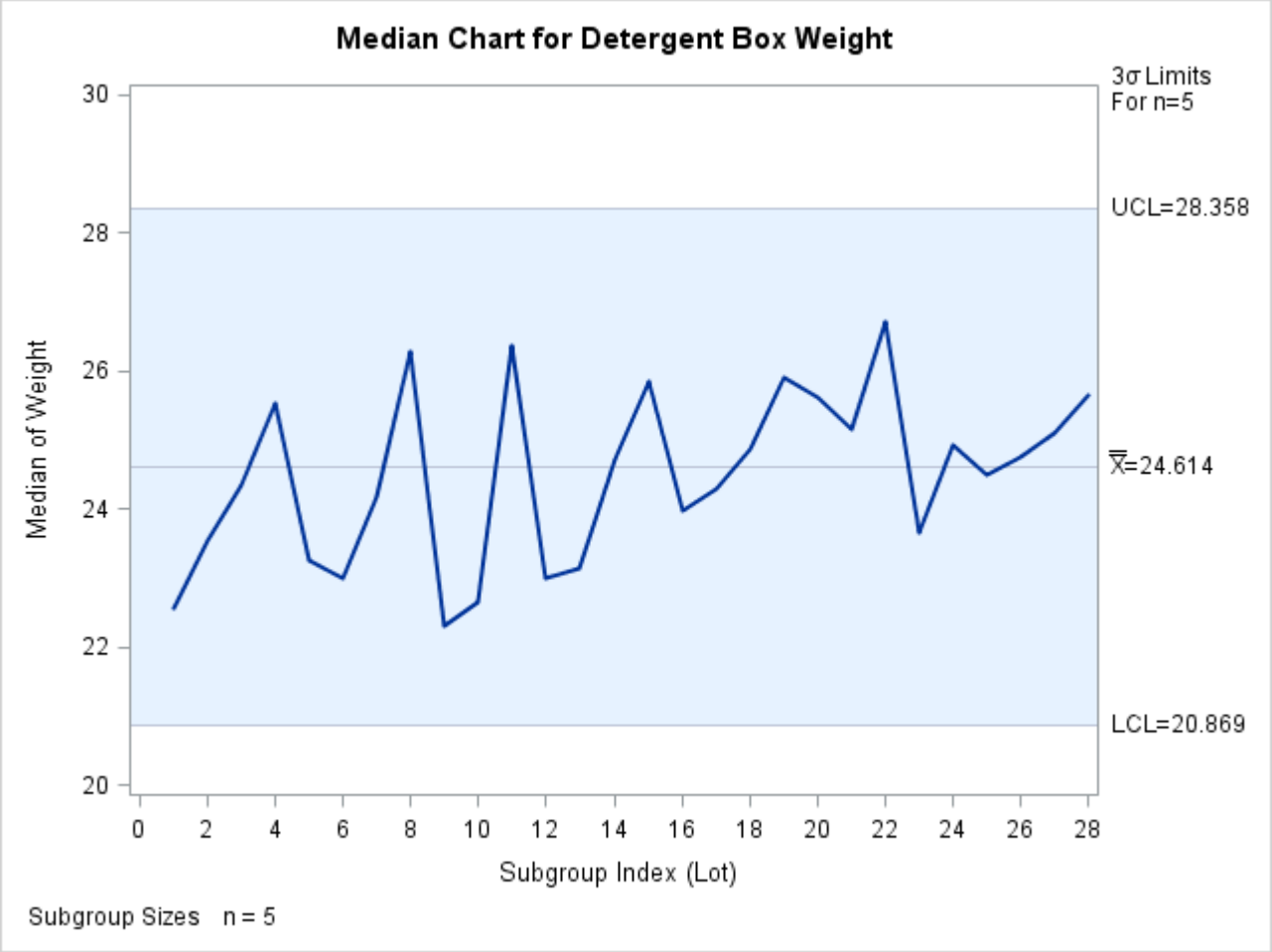


You can also request that the central line indicate the average of the subgroup means. The following statements create a median chart with this value for the central line:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal   = 3
                       odstitle   = title
                       medcentral = avgmean;
run;
```

The MEDCENTRAL= option specifies the value used for the central line. In this case, MEDCENTRAL=AVGMEAN is specified to request a central line indicating the average of the subgroup means. The resulting chart is shown in Output 19.14.2.

Output 19.14.2 Central Line is Average of Subgroup Means

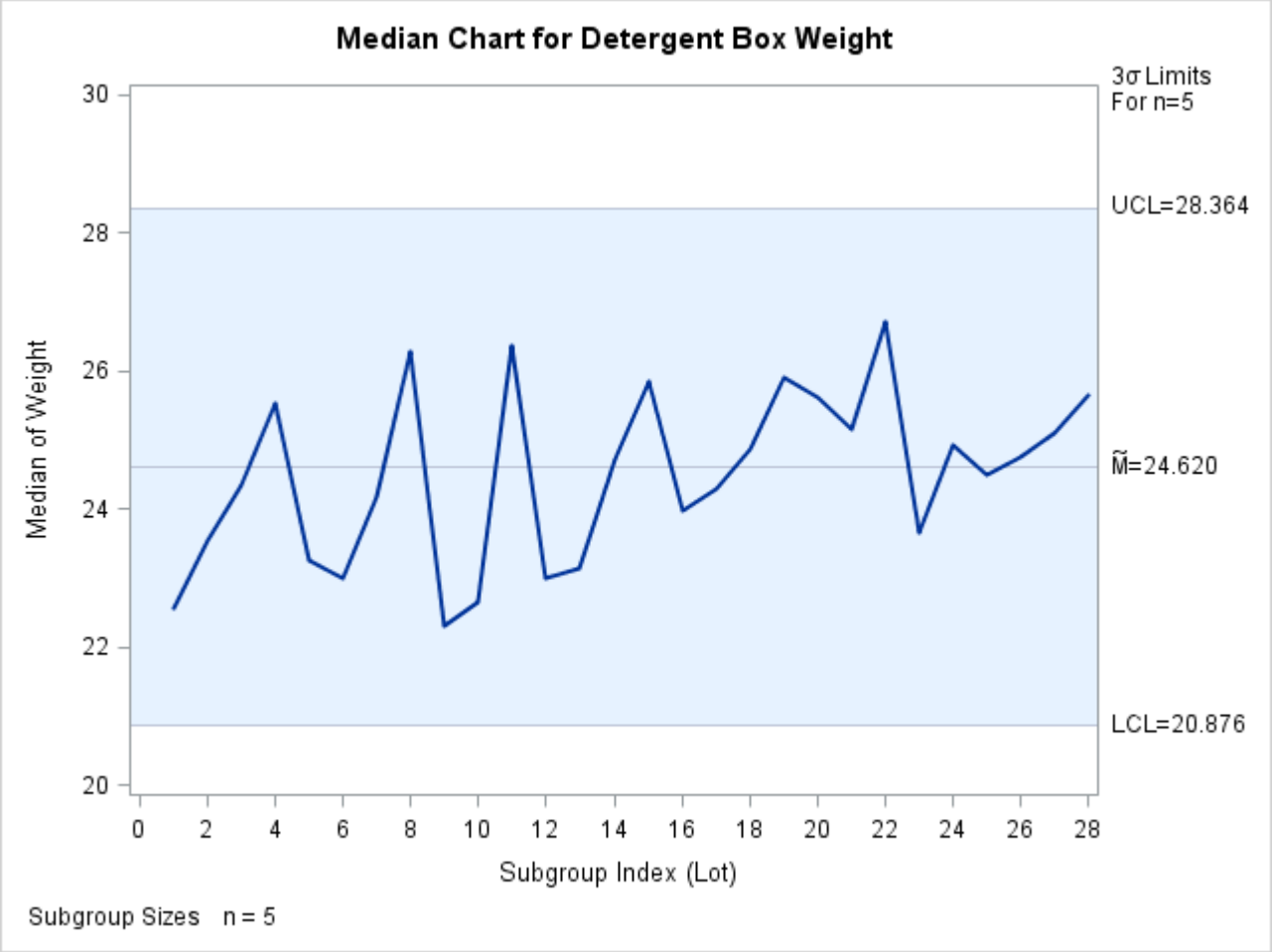


If you specify MEDCENTRAL=MEDMED, the median of the subgroup medians is used for the central line, as demonstrated by the following statements:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal   = 3
                       odstitle   = title
                       medcentral = medmed;
run;
```

The resulting chart is shown in Output 19.14.3.

Output 19.14.3 Central Line is Median of Subgroup Medians

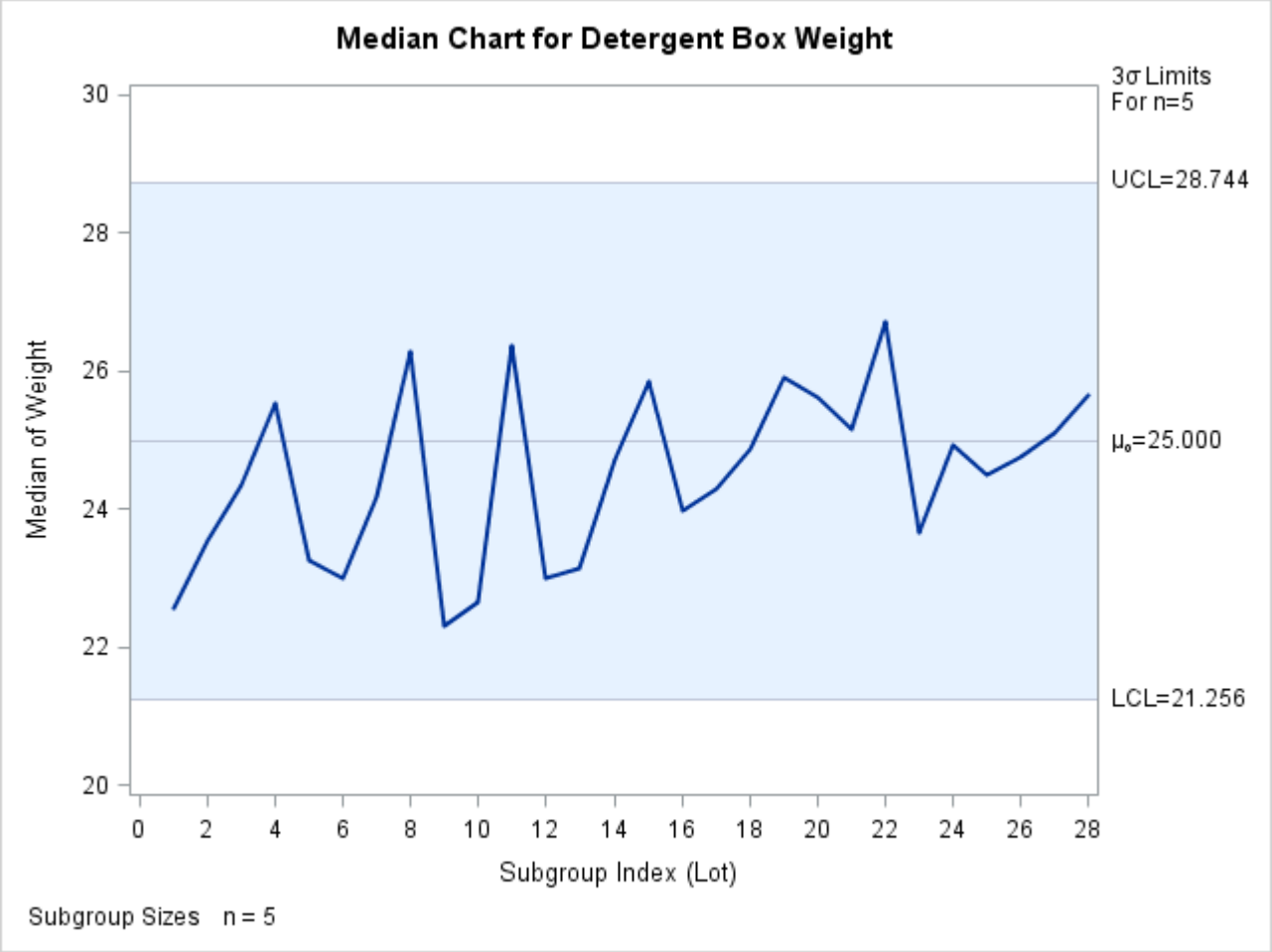


In some situations a standard value for the process mean (μ_0) is available. For instance, extensive startup testing provides an estimate of the process mean. If specified, this value is used for the central line. The following statements create a median chart for the detergent box weights with $\mu_0 = 25$:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal = 3
                       mu0      = 25
                       odstitle = title
                       xsymbol  = mu0;
run;
```

The MU0= option specifies the standard value for the process mean, and the XSYMBOL= option specifies the label for the central line. In this case, XSYMBOL=MU0 is specified to indicate that the central line represents a standard value. The resulting chart is shown in Output 19.14.4.

Output 19.14.4 Median Chart for Detergent Box Weight Data



Note that you can also provide μ_0 with the _MEAN_ variable in a LIMITS= data set. For example, the following DATA step creates a data set (Dlims) which contains the same standard value specified in the preceding MCHART statement:

```
data Dlims;
   _var_    = "Weight  ";
   _subgrp_ = "Lot     ";
   _mean_   = 25;
run;
```

The _VAR_ and _SUBGRP_ variables are required if this data set is to be read as a LIMITS= data set in the PROC SHEWHART statement. These values must match the names of the *process* and *subgroup-variable* specified in the MCHART statement. The following statements specify the data set Dlims as a LIMITS= data set and create a median chart (not shown here) identical to the one in Output 19.14.4:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent limits=Dlims;
   mchart Weight*Lot / xsymbol  = mu0
                       odstitle = title
                       ndecimal = 3;
run;
```

For more information, see "Constructing Median Charts" on page 1585.

Example 19.15: Estimating the Process Standard Deviation

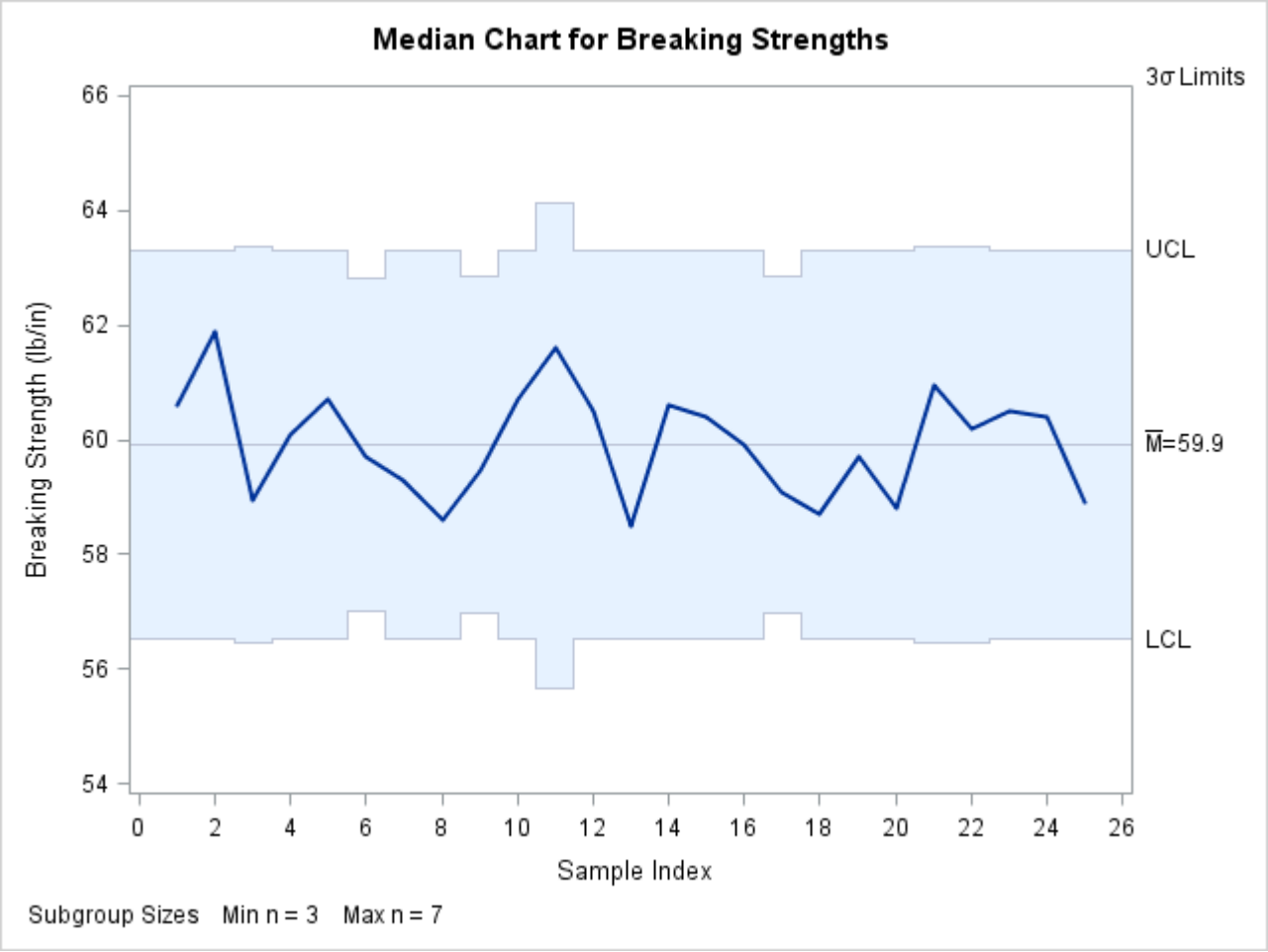
The following data set (Wire) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

```
data Wire;
   input Sample Size @;
   do i=1 to Size;
      input Breakstrength @@;
      output;
   end;
   drop i Size;
   label Breakstrength  ='Breaking Strength (lb/in)'
         Sample ='Sample Index';
   datalines;
 1  5 60.6 62.3 62.0 60.4 59.9
 2  5 61.9 62.1 60.6 58.9 65.3
 3  4 57.8 60.5 60.1 57.7
 4  5 56.8 62.5 60.1 62.9 58.9
 5  5 63.0 60.7 57.2 61.0 53.5
 6  7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
 7  5 59.3 61.7 59.1 58.1 60.3
 8  5 61.3 58.5 57.8 61.0 58.6
 9  6 59.5 58.3 57.5 59.4 61.5 59.6
10  5 61.7 60.7 57.2 56.5 61.5
11  3 63.9 61.6 60.9
12  5 58.7 61.4 62.4 57.3 60.5
13  5 56.8 58.5 55.7 63.0 62.7
14  5 62.1 60.6 62.1 58.7 58.3
15  5 59.1 60.4 60.4 59.0 64.1
16  5 59.9 58.8 59.2 63.0 64.9
17  6 58.8 62.4 59.4 57.1 61.2 58.6
18  5 60.3 58.7 60.5 58.6 56.2
19  5 59.2 59.8 59.7 59.3 60.0
20  5 62.3 56.0 57.0 61.8 58.8
21  4 60.5 62.0 61.4 57.7
22  4 59.3 62.4 60.4 60.0
23  5 62.4 61.3 60.5 57.7 60.2
24  5 61.2 55.5 60.2 60.4 62.4
25  5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request a median chart, shown in Output 19.15.1, for the wire breaking strength measurements:

```
title 'Median Chart for Breaking Strengths';
ods graphics on;
proc shewhart data=Wire;
   mchart Breakstrength*Sample / odstitle=title;
run;
```

Output 19.15.1 Median Chart with Varying Sample Sizes



Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

By default, the control limits shown in Output 19.15.1 are 3σ limits estimated from the data. You can use the STDDEVIATIONS option and the SMETHOD= option in the MCHART statement to control how the estimate of the process standard deviation σ is calculated. The STDDEVIATIONS option specifies that the estimate of σ is to be calculated from subgroup standard deviations rather than subgroup ranges, the default. The SMETHOD= option specifies the method for estimating σ. You can specify the following methods:

- NOWEIGHT

- MVLUE
- RMSDF

The NOWEIGHT method, which is the default, requests an unweighted average of subgroup estimates, the MVLUE method requests a minimum variance linear unbiased estimate, and the RMSDF method requests a weighted root-mean-square estimate. Note that the RMSDF method is only available if, in addition, you specify the STDDEVIATIONS option. For details, see "Methods for Estimating the Standard Deviation" on page 1594.

The following statements contain five MCHART statements, which calculate five different estimates for σ by specifying different combinations of options:

```
title 'Estimates of the Process Standard Deviation';
proc shewhart data=Wire;
   mchart Breakstrength*Sample / outlimits=Wirelim1
                         nochart outindex ='NOWEIGHT-Ranges';
   mchart Breakstrength*Sample / outlimits=Wirelim2
                         stddeviations
                         nochart outindex ='NOWEIGHT-Stds';
   mchart Breakstrength*Sample / outlimits=Wirelim3
                         smethod  =mvlue
                         nochart outindex ='MVLUE   -Ranges';
   mchart Breakstrength*Sample / outlimits=Wirelim4
                         stddeviations
                         smethod  =mvlue
                         nochart outindex ='MVLUE   -Stds';
   mchart Breakstrength*Sample / outlimits=Wirelim5
                         stddeviations
                         smethod  =rmsdf
                         nochart outindex ='RMSDF   -Stds';
run;
```

The OUTLIMITS= option names the data set containing the control limit information. The _STDDEV_ variable in the OUTLIMITS= data set contains the estimate of the process standard deviation. The OUTINDEX= option specifies the value of the _INDEX_ variable in the OUTLIMITS= data set and is used in this example to identify the estimation method. The following statements create a data set named Wlimits, which contains the five different estimates. This data set is listed in Output 19.15.2.

```
data Wlimits;
   set Wirelim1 Wirelim2 Wirelim3 Wirelim4 Wirelim5;
   keep _index_ _stddev_;
run;
```

Output 19.15.2 The Data Set Wlimits

The Wlimits Data Set

INDEX	_STDDEV_
NOWEIGHT-Ranges	2.11146
NOWEIGHT-Stds	2.15453
MVLUE -Ranges	2.11240
MVLUE -Stds	2.14790
RMSDF -Stds	2.17479

The median chart shown in Output 19.14.1 uses the estimate listed first in Output 19.15.2 ($\sigma = 2.11146$), because the MCHART statement used to create this chart omitted the STDDEVIATIONS option and the SMETHOD= option.

MRCHART Statement: SHEWHART Procedure

Overview: MRCHART Statement

The MRCHART statement creates charts for subgroup medians and ranges, which are used to analyze the central tendency and variability of a process.

You can use options in the MRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted medians and ranges or as probability limits
- tabulate subgroup sample sizes, subgroup medians, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup medians, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify the method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing charts of medians and ranges with the MRCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.

- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: MRCHART Statement

This section introduces the MRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the MRCHART statement is presented in the section "Syntax: MRCHART Statement" on page 1615, and advanced examples are given in the section "Examples: MRCHART Statement" on page 1638.

Creating Charts for Medians and Ranges from Raw Data

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

A consumer products company weighs detergent boxes (in pounds) to determine whether the fill process is in control. The following statements create a SAS data set named Detergent, which contains the weights for five boxes in each of 28 lots. A lot is considered a rational subgroup.

```
data Detergent;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
 1 17.39 26.93 19.34 22.56 24.49
 2 23.63 23.57 23.54 20.56 22.17
 3 24.35 24.58 23.79 26.20 21.55
 4 25.52 28.02 28.44 25.07 23.39
 5 23.25 21.76 29.80 23.09 23.70
 6 23.01 22.67 24.70 20.02 26.35
 7 23.86 24.19 24.61 26.05 24.18
 8 26.00 26.82 28.03 26.27 25.85
 9 21.58 22.31 25.03 20.86 26.94
10 22.64 21.05 22.66 29.26 25.02
11 26.38 27.50 23.91 26.80 22.53
12 23.01 23.71 25.26 20.21 22.38
13 23.15 23.53 22.98 21.62 26.99
14 26.83 23.14 24.73 24.57 28.09
15 26.15 26.13 20.57 25.86 24.70
16 25.81 23.22 23.99 23.91 27.57
17 25.53 22.87 25.22 24.30 20.29
18 24.88 24.15 25.29 29.02 24.46
19 22.32 25.96 29.54 25.92 23.44
20 25.63 26.83 20.95 24.80 27.25
21 21.68 21.11 26.07 25.17 27.63
```

```
22 26.72 27.05 24.90 30.08 25.22
23 31.58 22.41 23.67 23.47 24.90
24 28.06 23.44 24.92 24.64 27.42
25 21.10 22.34 24.96 26.50 24.51
26 23.80 24.03 24.75 24.82 27.21
27 25.10 26.09 27.21 24.28 22.45
28 25.53 22.79 26.26 25.85 25.64
;
```

A partial listing of Detergent is shown in Figure 19.43.

Figure 19.43 Partial Listing of the Data Set Detergent

The Data Set DETERGENT

Lot	Weight
1	17.39
1	26.93
1	19.34
1	22.56
1	24.49
2	23.63
2	23.57
2	23.54
2	20.56
2	22.17
3	24.35
3	24.58
3	23.79
3	26.20
3	21.55
4	25.52

The data set Detergent is said to be in "strung-out" form, because each observation contains the lot number and weight of a single box. The first five observations contain the weights for the first lot, the second five observations contain the weights for the second lot, and so on. Because the variable Lot classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Weight contains the weights and is referred to as the *process variable* (or *process* for short).

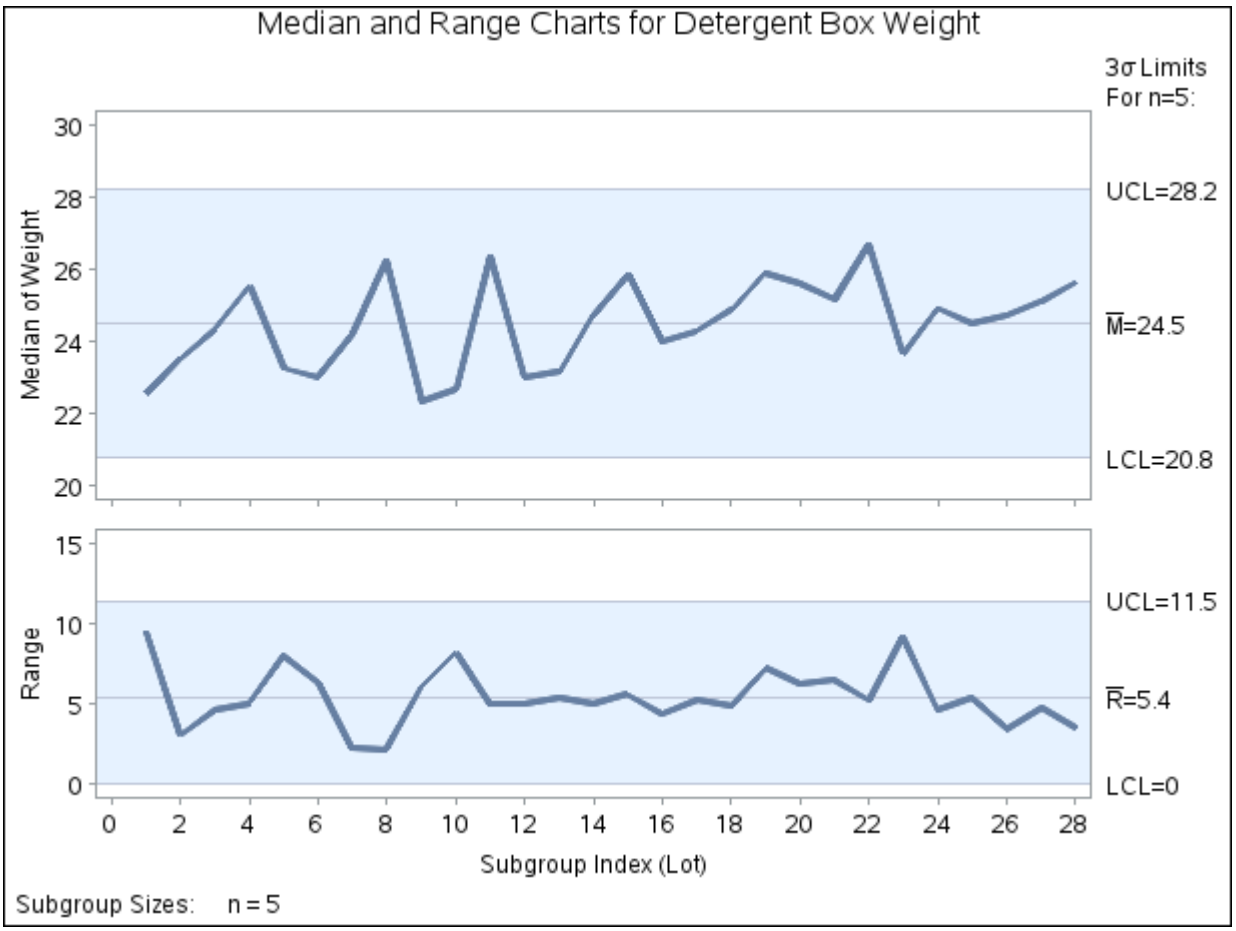
You can use median and range charts to determine whether the fill process is in control. The following statements create the charts shown in Figure 19.44:

```
ods graphics off;
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=Detergent;
   mrchart Weight*Lot ;
run;
```

This example illustrates the basic form of the MRCHART statement. After the keyword MRCHART, you specify the *process* to analyze (in this case, Weight) followed by an asterisk and the *subgroup-variable* (Lot).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 19.44 Median and Range Charts (Traditional Graphics)



Each point on the median chart represents the median of the measurements for a particular lot. For instance, the weights for the first lot are 17.39, 19.34, 22.56, 24.49, and 26.93, and consequently, the median plotted for this lot is 22.56. Each point on the range chart represents the range of the measurements for a particular batch. For instance, the range plotted for the first lot is 26.93 – 17.39 = 9.54. Because all of the points lie within the control limits, you can conclude that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.33. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1613.

For computational details, see "Constructing Charts for Medians and Ranges" on page 1628. For more details on reading raw data, see "DATA= Data Set" on page 1634.

Creating Charts for Medians and Ranges from Summary Data

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create median and range charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the MRCHART statement with data of this type.

The following data set (Detsum) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Lot). The variable WeightM contains the subgroup medians, the variable WeightR contains the subgroup ranges, and the variable WeightN contains the subgroup sample sizes (these are all five).

```
data Detsum;
   input Lot WeightM WeightR;
   WeightN = 5;
   datalines;
 1  22.56  9.54
 2  23.54  3.07
 3  24.35  4.65
 4  25.52  5.05
 5  23.25  8.04
 6  23.01  6.33
 7  24.19  2.19
 8  26.27  2.18
 9  22.31  6.08
10  22.66  8.21
11  26.38  4.97
12  23.01  5.05
13  23.15  5.37
14  24.73  4.95
15  25.86  5.58
16  23.99  4.35
17  24.30  5.24
18  24.88  4.87
19  25.92  7.22
20  25.63  6.30
21  25.17  6.52
22  26.72  5.18
23  23.67  9.17
24  24.92  4.62
25  24.51  5.40
26  24.75  3.41
27  25.10  4.76
28  25.64  3.47
;
```

A partial listing of Detsum is shown in Figure 19.45.

Figure 19.45 The Summary Data Set Detsum

Summary Data for Detergent Box Weights

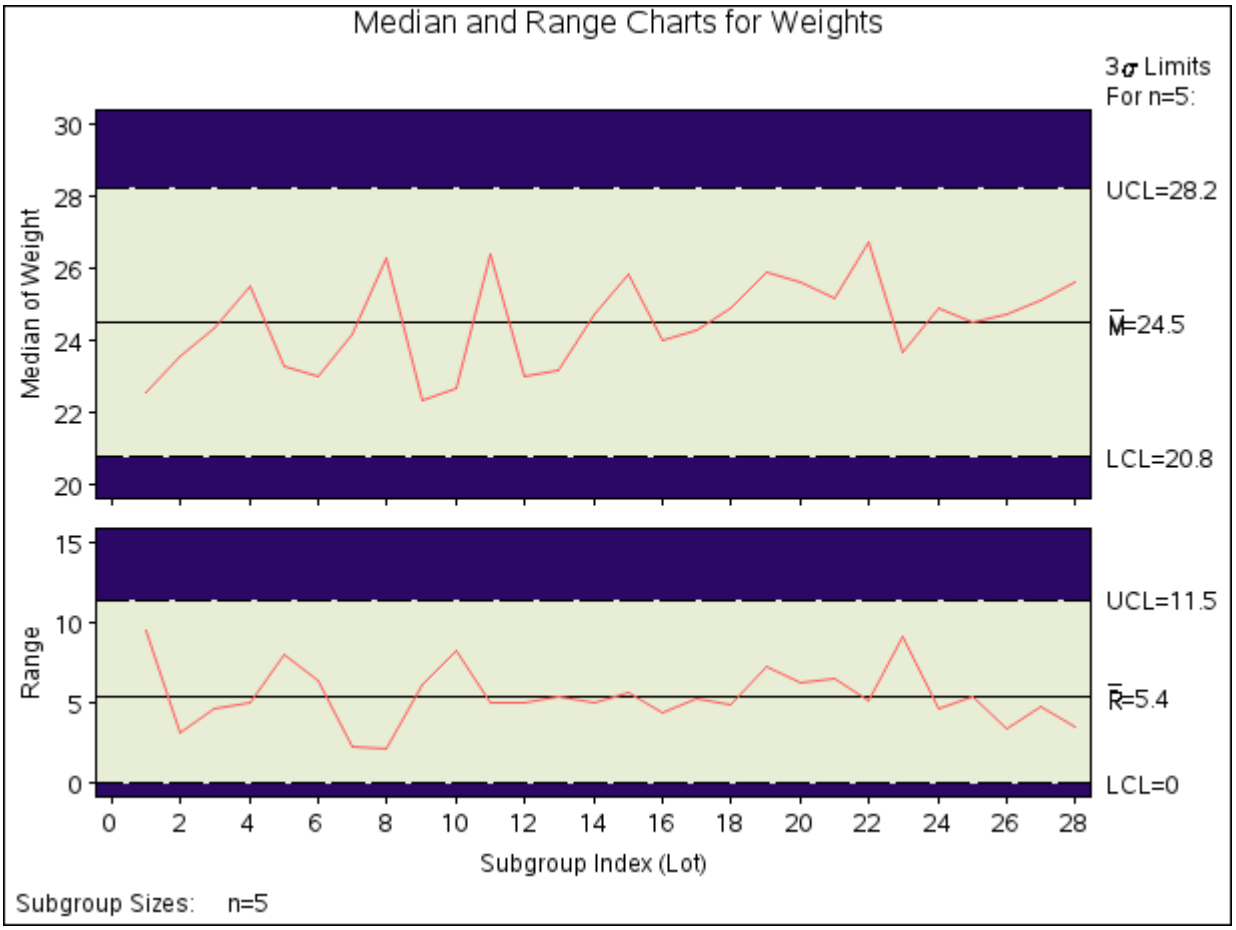
Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
symbol color = rose h = .8;
title 'Median and Range Charts for Weights';
proc shewhart history=Detsum;
   mrchart Weight*Lot / cframe    = vipb
                        cinfill   = ywh
                        cconnect  = rose
                        coutfill  = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and MRCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The charts are shown in Figure 19.46.

Figure 19.46 Median and Range Charts from Summary Data Set Detsum (Traditional Graphics with NOGSTYLE)



Note that Weight is *not* the name of a SAS variable in the data set Detsum but is, instead, the common prefix for the names of the three SAS variables WeightM, WeightR, and WeightN. The suffix characters *M*, *R*, and *N* indicate *median*, *range*, and *sample size*, respectively. This naming convention enables you to specify three subgroup summary variables in the HISTORY= data set with a single name (Weight), referred to as the *process*. The name Lot specified after the asterisk is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the MRCHART statement must contain the following variables:

- subgroup variable
- subgroup median variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup median, range, and sample size variables must begin with the prefix *process* specified in the MRCHART statement and end with the special suffix characters *M*, *R*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables WeightM, WeightR, and WeightN, the data set Detsum contained summary variables named medians, ranges, and sizes. The following statements would temporarily rename medians, ranges, and sizes to WeightM, WeightR, and WeightN, respectively:

```
proc shewhart
   history=Detsum (rename=(medians = WeightM
                           ranges  = WeightR
                           sizes   = WeightN ));
   mrchart Weight*Lot;
run;
```

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1635.

Saving Summary Statistics

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

In this example, the MRCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Detergent and create a summary data set named Dethist:

```
proc shewhart data=Detergent;
   mrchart Weight*Lot / outhistory = Dethist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to the charts in Figure 19.44. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the MRCHART statement. A complete list of options is presented in the section "Syntax: MRCHART Statement" on page 1615.

Figure 19.47 contains a partial listing of Dethist.

Figure 19.47 The Summary Data Set Dethist

Summary Data Set DETHIST for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5

There are four variables in the data set Dethist.

- Lot contains the subgroup index.
- WeightM contains the subgroup medians.
- WeightR contains the subgroup ranges.
- WeightN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *M*, *R*, and *N* to the *process* Weight specified in the MRCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1632.

Saving Control Limits

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

You can save the control limits for median and range charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1613) or modify the limits with a DATA step program.

The following statements read measurements from the data set Detergent (see "Creating Charts for Medians and Ranges from Raw Data" on page 1604) and save the control limits displayed in Figure 19.44 in a data set named Detlim:

```
proc shewhart data=Detergent;
   mrchart Weight*Lot / outlimits=Detlim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Detlim is listed in Figure 19.48.

Figure 19.48 The Data Set Detlim Containing Control Limit Information

Control Limits for Detergent Box Weights

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_
Weight	Lot	ESTIMATE	5	.002909021	3	20.7554	24.4996	28.2439

LCLR	_R_	_UCLR_	_STDDEV_
0	5.42036	11.4613	2.33041

The data set Detlim contains one observation with the limits for *process* Weight. The variables _LCLM_ and _UCLM_ contain the control limits for the medians, and the variable _MEAN_ contains the central line. The variables _LCLR_ and _UCLR_ contain the control limits for the ranges, and the variable _R_ contains the central line. The values of _MEAN_ and _STDDEV_ are estimates of the process mean and process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1631.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Detergent;
   mrchart Weight*Lot / outtable=Dtable
                        nochart;
run;
```

This data set contains one observation for each subgroup sample. The variables _SUBMED_, _SUBR_, and _SUBN_ contain the subgroup medians, subgroup ranges, and subgroup sample sizes. The variables _LCLM_ and _UCLM_ contain the control limits for the median chart, and the variables _LCLR_ and _UCLR_ contain the control limits for the range chart. The variable _MEAN_ contains the central line for the median chart, and the variable _R_ contains the central line for the range chart. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1633.

The data set Dtable is listed in Figure 19.49.

Figure 19.49 The Data Set Dtable

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_STDDEV_
Weight	1	3	5	5	20.7554	22.56	24.4996	28.2439	2.33041
Weight	2	3	5	5	20.7554	23.54	24.4996	28.2439	2.33041
Weight	3	3	5	5	20.7554	24.35	24.4996	28.2439	2.33041
Weight	4	3	5	5	20.7554	25.52	24.4996	28.2439	2.33041
Weight	5	3	5	5	20.7554	23.25	24.4996	28.2439	2.33041
Weight	6	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041
Weight	7	3	5	5	20.7554	24.19	24.4996	28.2439	2.33041
Weight	8	3	5	5	20.7554	26.27	24.4996	28.2439	2.33041
Weight	9	3	5	5	20.7554	22.31	24.4996	28.2439	2.33041
Weight	10	3	5	5	20.7554	22.66	24.4996	28.2439	2.33041
Weight	11	3	5	5	20.7554	26.38	24.4996	28.2439	2.33041
Weight	12	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041
Weight	13	3	5	5	20.7554	23.15	24.4996	28.2439	2.33041
Weight	14	3	5	5	20.7554	24.73	24.4996	28.2439	2.33041
Weight	15	3	5	5	20.7554	25.86	24.4996	28.2439	2.33041
Weight	16	3	5	5	20.7554	23.99	24.4996	28.2439	2.33041
Weight	17	3	5	5	20.7554	24.30	24.4996	28.2439	2.33041
Weight	18	3	5	5	20.7554	24.88	24.4996	28.2439	2.33041
Weight	19	3	5	5	20.7554	25.92	24.4996	28.2439	2.33041
Weight	20	3	5	5	20.7554	25.63	24.4996	28.2439	2.33041
Weight	21	3	5	5	20.7554	25.17	24.4996	28.2439	2.33041
Weight	22	3	5	5	20.7554	26.72	24.4996	28.2439	2.33041

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	9.54	5.42036	11.4613	
	0	3.07	5.42036	11.4613	
	0	4.65	5.42036	11.4613	
	0	5.05	5.42036	11.4613	
	0	8.04	5.42036	11.4613	
	0	6.33	5.42036	11.4613	
	0	2.19	5.42036	11.4613	
	0	2.18	5.42036	11.4613	
	0	6.08	5.42036	11.4613	
	0	8.21	5.42036	11.4613	
	0	4.97	5.42036	11.4613	
	0	5.05	5.42036	11.4613	
	0	5.37	5.42036	11.4613	
	0	4.95	5.42036	11.4613	
	0	5.58	5.42036	11.4613	
	0	4.35	5.42036	11.4613	
	0	5.24	5.42036	11.4613	
	0	4.87	5.42036	11.4613	
	0	7.22	5.42036	11.4613	
	0	6.30	5.42036	11.4613	
	0	6.52	5.42036	11.4613	
	0	5.18	5.42036	11.4613	

Figure 19.49 *continued*

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_STDDEV_
Weight	23	3	5	5	20.7554	23.67	24.4996	28.2439	2.33041
Weight	24	3	5	5	20.7554	24.92	24.4996	28.2439	2.33041
Weight	25	3	5	5	20.7554	24.51	24.4996	28.2439	2.33041
Weight	26	3	5	5	20.7554	24.75	24.4996	28.2439	2.33041
Weight	27	3	5	5	20.7554	25.10	24.4996	28.2439	2.33041
Weight	28	3	5	5	20.7554	25.64	24.4996	28.2439	2.33041

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	9.17	5.42036	11.4613	
	0	4.62	5.42036	11.4613	
	0	5.40	5.42036	11.4613	
	0	3.41	5.42036	11.4613	
	0	4.76	5.42036	11.4613	
	0	3.47	5.42036	11.4613	

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Dtable and display charts (not shown here) identical to those in Figure 19.44:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart table=Dtable;
   mrchart Weight*Lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1636.

Reading Preestablished Control Limits

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Detlim saved control limits computed from the measurements in Detergent. This example shows how these limits can be applied to new data provided in the following data set:

```
data Detergent2;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
29 16.66 27.49 18.87 22.53 24.72
30 23.74 23.67 23.64 20.26 22.09
31 24.56 24.82 23.92 26.67 21.38
32 25.89 28.73 29.21 25.38 23.47
33 23.32 21.61 30.75 23.13 23.82
```

```
34 23.04 22.65 24.96 19.64 26.84
35 24.01 24.38 24.86 26.50 24.37
36 26.43 27.36 28.74 26.74 26.27
37 21.41 22.24 25.34 20.59 27.51
38 22.62 20.81 22.64 30.15 25.32
39 26.86 28.14 24.06 27.35 22.49
40 23.03 23.83 25.59 19.85 22.33
41 23.19 23.63 23.00 21.46 27.57
42 27.38 23.18 24.99 24.81 28.82
43 26.60 26.58 20.26 26.27 24.96
44 26.22 23.28 24.15 24.06 28.23
45 25.90 22.88 25.55 24.50 19.95
46 16.66 27.49 18.87 22.53 24.72
47 23.74 23.67 23.64 20.26 22.09
48 24.56 24.82 23.92 26.67 21.38
49 25.89 28.73 29.21 25.38 23.47
50 23.32 21.61 30.75 23.13 23.82
;
```

The following statements create median and range charts for the data in Detergent2 using the control limits in Detlim:

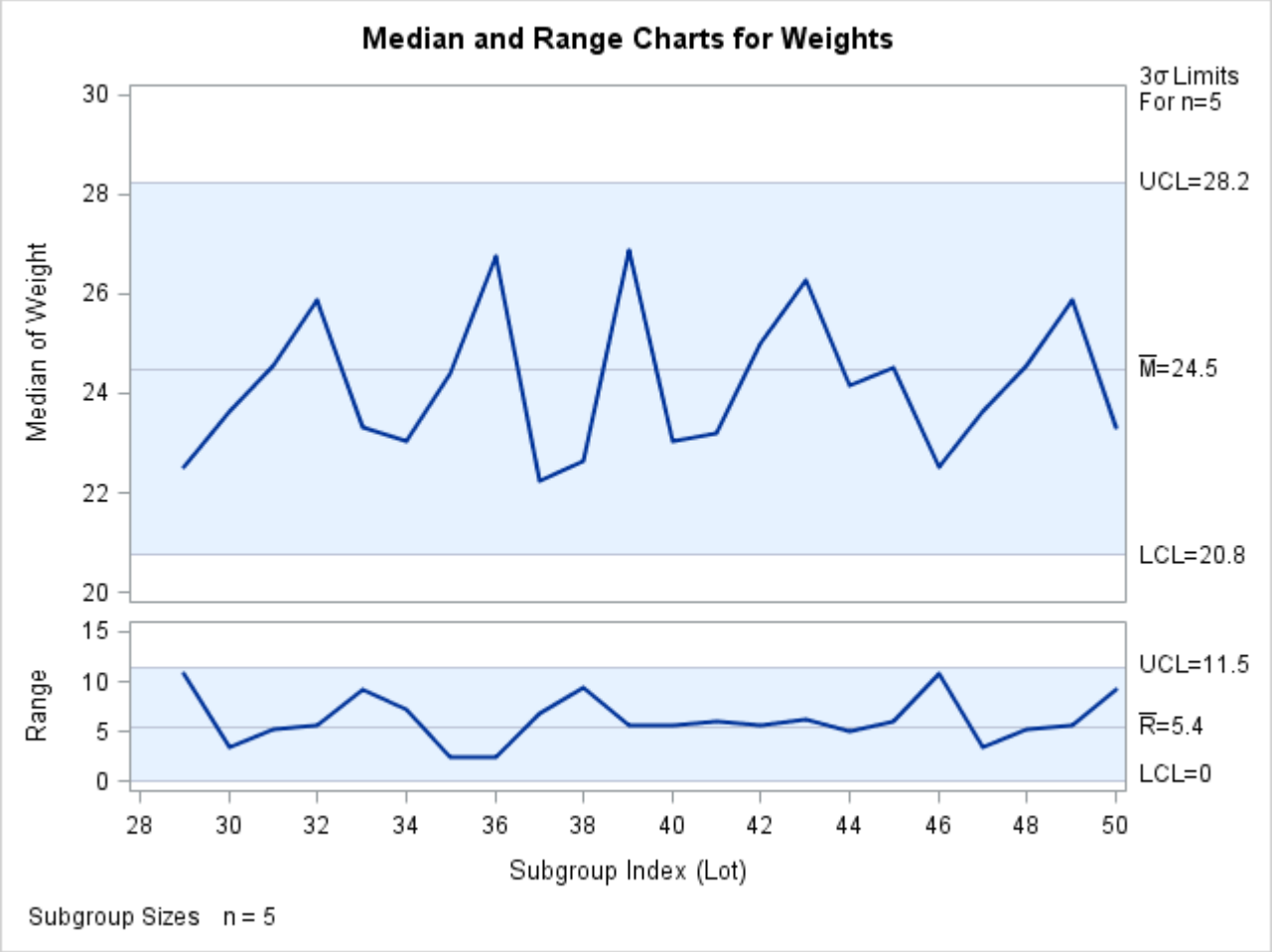
```
ods graphics on;
title 'Median and Range Charts for Weights';
proc shewhart data=Detergent2 limits=Detlim;
   mrchart Weight*Lot / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the median and range charts are created using ODS Graphics instead of traditional graphics. The charts are shown in Figure 19.50.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Weight
- the value of _SUBGRP_ matches the *subgroup-variable* name Lot

Figure 19.50 Median and Range Charts for Second Set of Detergent Box Weights (ODS Graphics)



The charts indicate that the process is in control, because all the medians and ranges lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1634 for details concerning the variables that you must provide.

Syntax: MRCHART Statement

The basic syntax for the MRCHART statement is as follows:

MRCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

MRCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of MRCHART statements in the SHEWHART procedure. The components of the MRCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Medians and Ranges from Raw Data" on page 1604.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Medians and Ranges from Summary Data" on page 1606.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1610.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct median and range charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   mrchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding MRCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the medians and ranges.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create median and range charts using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   mrchart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the MRCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.31 MRCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit on median chart
LCLLABEL2=	Specifies label for lower control limit on *R* chart
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits

Table 19.31 *continued*

Option	Description
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on median chart
NDECIMAL2=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on *R* chart
NOCTL	Suppresses display of central line on median chart
NOCTL2	Suppresses display of central line on *R* chart
NOLCL	Suppresses display of lower control limit on median chart
NOLCL2	Suppresses display of lower control limit on *R* chart
NOLIMIT0	Suppresses display of zero lower control limit on *R* chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit on median chart
NOUCL2	Suppresses display of upper control limit on *R* chart
RSYMBOL=	Specifies label for central line on *R* chart
UCLLABEL=	Specifies label for upper control limit on median chart
UCLLABEL2=	Specifies label for upper control limit on *R* chart
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line on median chart
Process Mean and Standard Deviation Options	
MEDCENTRAL=	Specifies method for estimating process mean μ
MU0=	Specifies known value of μ_0 for process mean μ
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on median chart
ALLLABEL2=	Labels every point on *R* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points

Table 19.31 *continued*

Option	Description
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits on median chart
OUTLABEL2=	Labels points outside control limits on *R* chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes for the median chart
TESTS2=	Specifies tests for special causes for the *R* chart
TEST2RESET=	Enables tests for special causes to be reset for the *R* chart
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied to the median chart
ZONELABELS	Adds labels A, B, and C to zone lines for median chart
ZONE2LABELS	Adds labels A, B, and C to zone lines for *R* chart
ZONES	Adds lines to median chart delineating zones A, B, and C

Table 19.31 *continued*

Option	Description
ZONES2	Adds lines to *R* chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels median chart zone lines with their values
ZONE2VALUES	Labels *R* zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis

Table 19.31 *continued*

Option	Description
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default to *R* chart
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of median chart
VAXIS2=	Specifies major tick mark values for vertical axis of *R* chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of charts
NOCHART2	Suppresses creation of *R* chart

Table 19.31 *continued*

Option	Description
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	Displays median and *R* charts on separate screens or pages
TOTPANELS=	Specifies number of pages or screens to be used to display chart
YPCT1=	Specifies length of vertical axis on median chart as a percentage of sum of lengths of vertical axes for median and *R* charts
ZEROSTD	Displays median chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on *R* chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on *R* chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on median chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on *R* chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels

Table 19.31 *continued*

Option	Description
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to median chart
ANNOTATE2=	Specifies annotate data set that adds features to *R* chart
DESCRIPTION=	Specifies description of median chart's GRSEG catalog entry
DESCRIPTION2=	Specifies description of *R* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of median chart's GRSEG catalog entry
NAME2=	Specifies name of *R* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color

Table 19.31 *continued*

Option	Description
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers in the median chart
MARKERDISPLAY2=	Specifies a subset of subgroups to be plotted with markers in the *R* chart
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers in the median chart
MARKERLABEL2=	Specifies labels for subgroups that are plotted with markers in the *R* chart
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases

Table 19.31 *continued*

Option	Description
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits

Table 19.31 *continued*

Option	Description
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star

Table 19.31 *continued*

Option	Description
STARVERTICES=	superimposes star at each point on median chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Table 19.31 *continued*

Option	Description
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: MRCHART Statement

The following sections provide details that are specific to the MRCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Medians and Ranges

The following notation is used in this section:

μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	Mean of measurements in ith subgroup
R_i	Range of measurements in ith subgroup
n_i	Sample size of ith subgroup
N	The number of subgroups
x_{ij}	jth measurement in the ith subgroup, $j = 1, 2, 3, \ldots, n_i$
$x_{i(j)}$	jth largest measurement in the ith subgroup. Then

$$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$

$\overline{\overline{X}}$	Weighted average of subgroup means
M_i	Median of the measurements in the ith subgroup:

$$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$

$\bar{M}$	Average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$
$\tilde{M}$	Median of the subgroup medians. Denote the jth largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	Standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	100pth percentile ($0 < p < 1$) of the distribution of the median of n independent observations from a normal population with unit standard deviation
$d_2(n)$	Expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	Standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	100pth percentile of the standard normal distribution
$D_p(n)$	100pth percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on a median chart indicates the value of a subgroup median (M_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $M_{10} = 15$. Each point on a range chart indicates the value of a subgroup range (R_i). For example, the value plotted for the tenth subgroup is $R_{10} = 19 - 12 = 7$.

Central Lines

On a median chart, the value of the central line indicates an estimate for μ, which is computed as

- $\bar{M}$ by default
- $\overline{\overline{X}}$ when you specify MEDCENTRAL=AVGMEAN
- $\tilde{M}$ when you specify MEDCENTRAL=MEDMED
- μ_0 when you specify μ_0 with the MU0= option

On the range chart, by default, the central line for the ith subgroup indicates an estimate for the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. The central line on the range chart varies with n_i.

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard errors of M_i and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that M_i or R_i exceeds its limits

The following table provides the formulas for the limits:

Table 19.33 Limits for Median and Range Charts

Control Limits	
Median Chart	LCL = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$
	UCL = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$
Range Chart	LCL = lower control limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$
	UCL = upper control limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$
Probability Limits	
Median Chart	LCL = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$
	UCL = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$
Range Chart	LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$
	UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

In Table 19.33, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN, and replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED. Replace $\bar{M}$ with μ_0 if you specify μ_0 with the MU0= option, and replace $\hat{\sigma}$ with σ_0 if you specify σ_0 with the SIGMA0= option.

The formulas assume that the data are normally distributed. Note that the limits for both charts vary with n_i and that the probability limits for R_i are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 19.34 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	value of central line on range chart
SIGMAS	multiple (k) of standard error of M_i or R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the MRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
USL	upper specification limit
VAR	*process* specified in the MRCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLM_, _UCLM_, _LCLR_, _R_, and _UCLR_.

2. If the limits are defined in terms of a multiple k of the standard errors of M_i and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _ALPHA_.

3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = F^{-1}_{med}(1 - \alpha/2, n)$, where $F^{-1}_{med}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.

4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MRCHART statement. For an example of an OUTLIMITS= data set, see "Saving Control Limits" on page 1610.

OUTHISTORY= Data Set

The OUTHISTORY= option saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup median variable named by *process* suffixed with *M*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup medians, ranges, and sample sizes are created for each *process* specified in the MRCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   mrchart (Width Diameter)*lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthM, WidthR, WidthN, DiameterM, DiameterR, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1609.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.35 lists the variables that are saved.

Table 19.35 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on median chart
EXLIMR	Control limit exceeded on range chart
LCLM	Lower control limit for median
LCLR	Lower control limit for range
LIMITN	Nominal sample size associated with the control limits
MEAN	Estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	Average range
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBM	Subgroup median
SUBN	Subgroup sample size
SUBR	Subgroup range
TESTS	Tests for special causes signaled on median chart
TESTS2	Tests for special causes signaled on range chart
UCLM	Upper control limit for mean
UCLR	Upper control limit for range
VAR	*Process* specified in the MRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variable _TESTS2_ is saved if you specify the TESTS2= option. The *k*th character of a value of _TESTS2_ is *k* if Test *k* is positive at that subgroup.

4. The variables _EXLIM_, _EXLIMR_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limits" on page 1610.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the MRCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the MRCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating Charts for Medians and Ranges from Raw Data" on page 1604.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   mrchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLM_, _MEAN_, _UCLM_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_ , which are used to calculate the control limits according to the equations in Table 19.33

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1613.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedures or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the MRCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup median variable for each *process*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup median, subgroup range, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *X*, *M*, *R*, and *N*, respectively. You must provide the subgroup mean variable only if you specify the MEDCENTRAL=AVGMEAN option.

For example, consider the following statements:

```
proc shewhart history=Summary;
   mrchart (Weight Yieldstrength)*Batch / medcentral=avgmean;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightM, WeightR, WeightN, YieldstrengthX, YieldstrengthM, YieldstrengthR, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Medians and Ranges from Summary Data" on page 1606.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to read data sets created by other SAS procedures. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.36 lists the variables required in a TABLE= data set used with the MRCHART statement.

Table 19.36 Variables Required in a TABLE= Data Set

Variable	Description
LCLM	Lower control limit for median
LCLR	Lower control limit for range
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
_R_	Average range
Subgroup-variable	Values of the *subgroup-variable*
SUBM	Subgroup median
SUBN	Subgroup sample size
SUBR	Subgroup range
UCLM	Upper control limit for median
UCLR	Upper control limit for range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup medians and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup ranges and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1610.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, two methods are available for estimating the process standard deviation σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is equal to $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

See Example 19.16 for illustrations of the default and MVLUE methods.

Examples: MRCHART Statement

This section provides advanced examples of the MRCHART statement.

Example 19.16: Working with Unequal Subgroup Sample Sizes

NOTE: See *Median and Range Charts-Unequal Subgroup Sizes* in the SAS/QC Sample Library.

A brewery monitors its bottling process to ensure that each bottle is filled with the proper amount of beer. The following data set contains the amount of beer recorded in fluid ounces for 23 batches:

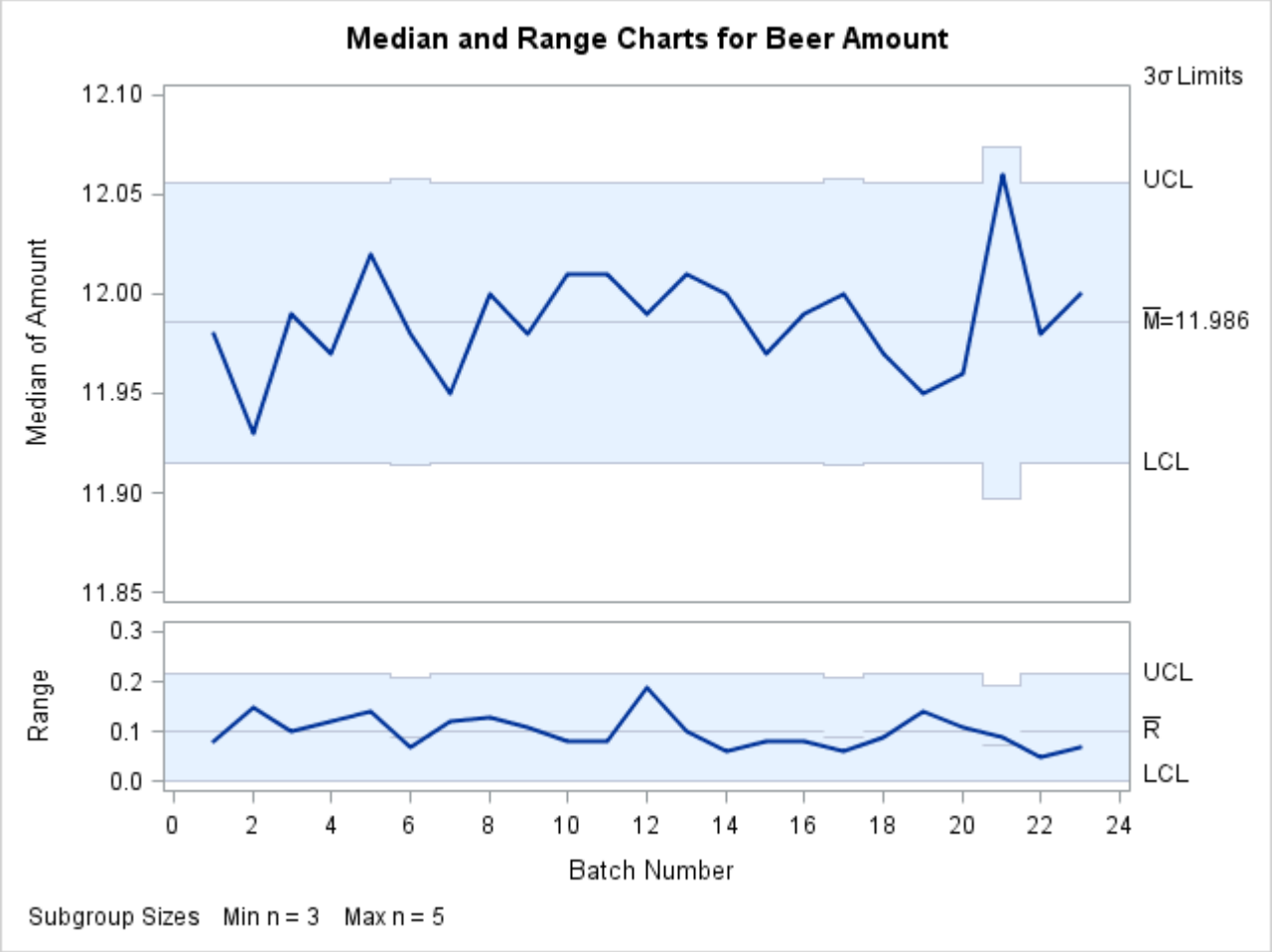
```
data Beer;
   input Batch size @;
   do i=1 to size;
      input Amount @@;
      output;
   end;
   drop i size;
```

```
   label Batch  ='Batch Number';
   datalines;
 1  5  12.01 11.97 11.93 11.98 12.00
 2  5  11.88 11.98 11.93 12.03 11.92
 3  5  11.93 11.99 12.00 12.03 11.95
 4  5  11.98 11.94 12.02 11.90 11.97
 5  5  12.02 12.02 11.98 12.04 11.90
 6  4  11.98 11.98 12.00 11.93
 7  5  11.93 11.95 12.02 11.91 12.03
 8  5  12.00 11.98 12.02 11.89 12.01
 9  5  11.98 11.93 11.99 12.02 11.91
10  5  11.97 12.02 12.05 12.01 11.97
11  5  12.02 12.01 11.97 12.02 11.94
12  5  11.93 11.83 11.99 12.02 12.01
13  5  12.01 11.98 11.94 12.04 12.01
14  5  11.98 11.96 12.02 12.00 12.00
15  5  11.97 11.99 12.03 11.95 11.96
16  5  11.99 11.95 11.96 12.03 12.01
17  4  11.99 11.97 12.03 12.01
18  5  11.94 11.96 11.98 12.03 11.97
19  5  11.97 11.87 11.90 12.01 11.95
20  5  11.96 11.94 11.96 11.98 12.05
21  3  12.06 12.07 11.98
22  5  12.01 11.98 11.96 11.97 12.00
23  5  12.00 12.02 12.03 11.99 11.96
;
```

A batch is regarded as a rational subgroup. Five bottles of beer are supposed to be tested in each batch. However, in batch 6 and batch 17 only four bottles are tested, and in batch 21 only three bottles are tested. The following statements request median and range charts, shown in Output 19.16.1, for the beer amounts:

```
ods graphics on;
title 'Median and Range Charts for Beer Amount';
proc shewhart data=Beer;
   mrchart Amount*Batch / odstitle=title;
run;
```

Output 19.16.1 Median and Range Charts with Varying Sample Sizes



Because none of the subgroup medians or subgroup ranges fall outside their respective control limits, you can conclude that the process is in control.

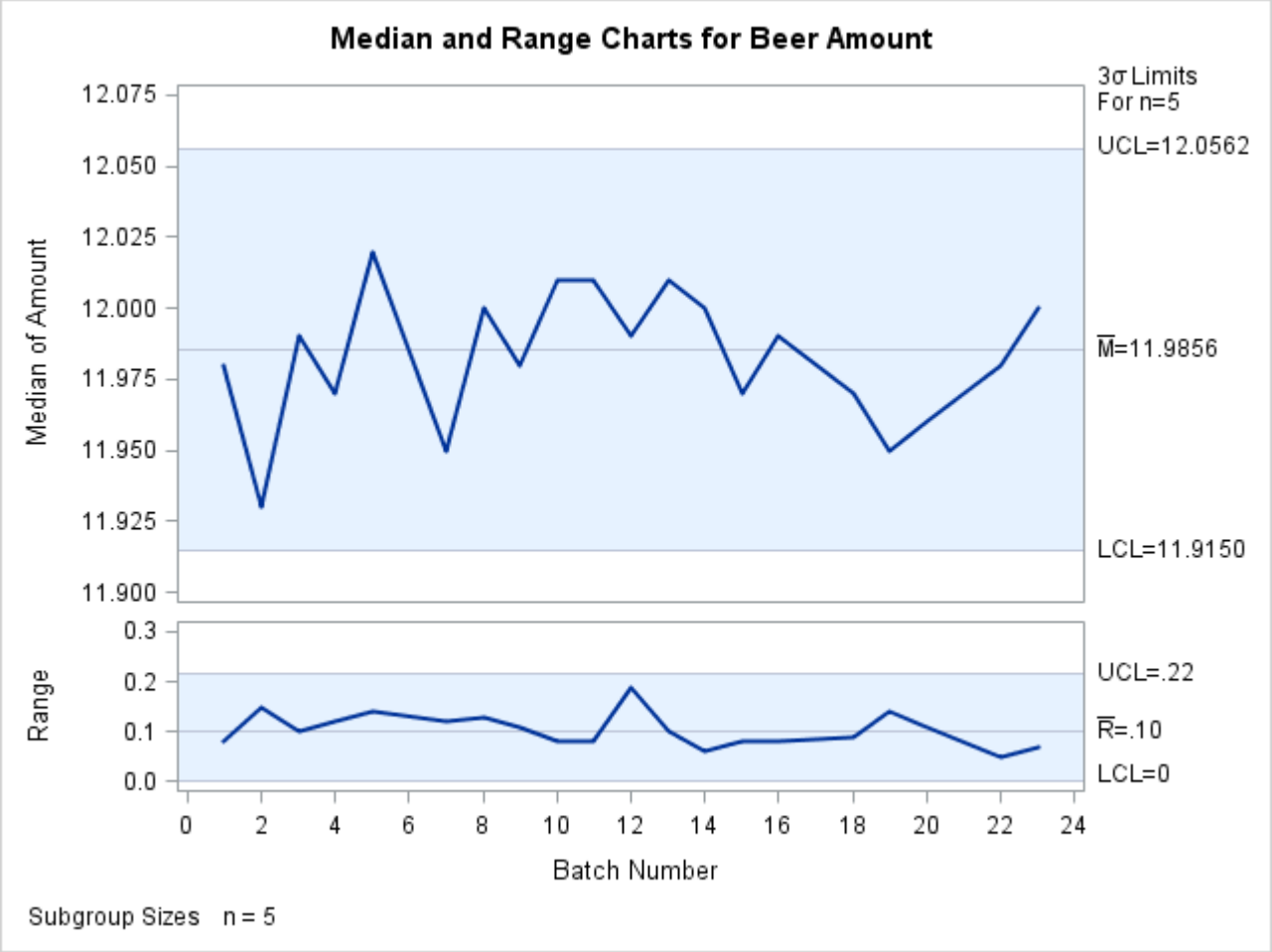
Note that the central line on the range chart and the control limits on both charts vary with the subgroup sample size. The subgroup sample size legend displays the minimum and maximum subgroup sample sizes.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for the control limits, as illustrated by the following statements:

```
title 'Median and Range Charts for Beer Amount';
proc shewhart data=Beer;
   mrchart Amount*Batch / limitn=5 odstitle=title;
run;
```

The resulting charts are shown in Output 19.16.2.

Output 19.16.2 Control Limits Based on Fixed Sample Size

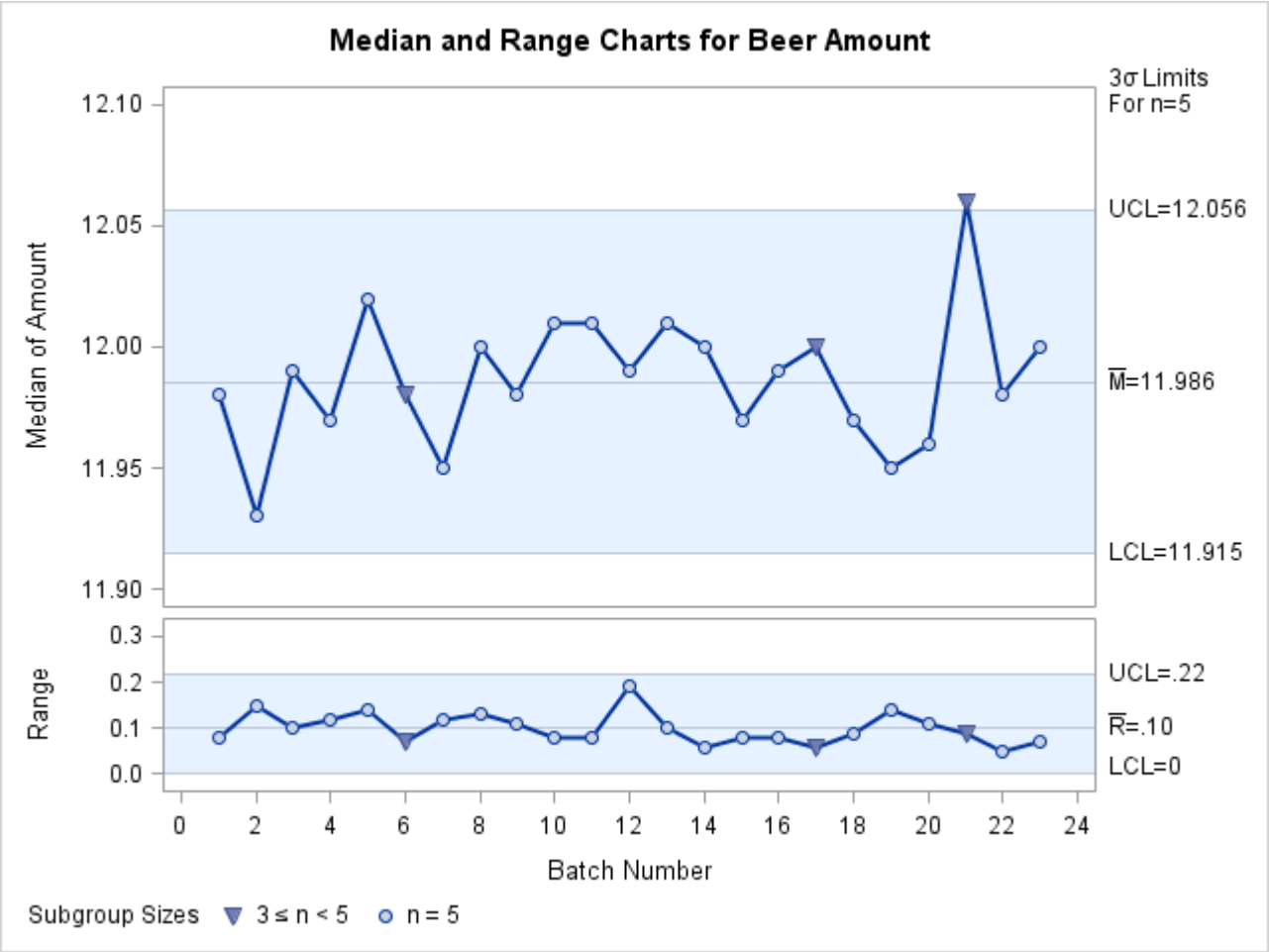


Note that the points displayed on the chart are those corresponding to subgroups whose sample size matches the nominal sample size (five) specified with the LIMITN= option. Points are not plotted for batches 6, 17, and 21. To display points for all subgroups (regardless of subgroup sample size), specify the ALLN option. The following statements produce the charts shown in Output 19.16.3:

```
title 'Median and Range Charts for Beer Amount';
proc shewhart data=Beer;
   mrchart Amount*Batch / limitn   = 5
                          odstitle = title
                          alln
                          nmarkers;
run;
```

The NMARKERS option requests special symbols that identify points for which the subgroup sample size differs from the nominal sample size. In Output 19.16.3, the median amount for batch 21 exceeds the upper control limits, indicating that the process is not in control. This illustrates the approximate nature of fixed control limits used with subgroup samples of varying sizes.

Output 19.16.3 Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes σ as an unweighted average of subgroup estimates of σ. The MVLUE method assigns greater weight to estimates of σ from subgroups with larger sample sizes. If the subgroup sample sizes are constant, the MVLUE method reduces to the NOWEIGHT method.

For details, see "Methods for Estimating the Standard Deviation" on page 1637. The following statements estimate σ using both methods:

```
proc shewhart data=Beer;
   mrchart Amount*Batch / outindex  = 'Default'
                          outlimits = Blim1
                          nochart;
   mrchart Amount*Batch / smethod   = mvlue
                          outindex  = 'MVLUE'
                          outlimits = Blim2
                          nochart;
run;

data Blimits;
```

```
   set Blim1 Blim2;
run;
```

The estimates are saved as values of the variable _STDDEV_ in the data set Blimits, which is listed in Output 19.16.4. The bookkeeping variable _INDEX_ identifies the estimate.

Output 19.16.4 The Data Set Blimits

The Data Set Blimits

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_
Amount	Batch	Default	ESTIMATE	V	V	3	V	11.9856
Amount	Batch	MVLUE	ESTIMATE	V	V	3	V	11.9856

UCLM	_LCLR_	_R_	_UCLR_	_STDDEV_
V	V	V	V	0.043938
V	V	V	V	0.044004

In the data set Blimits, the variables _LIMITN_, _ALPHA_, _LCLM_, _UCLM_, _LCLR_, _R_, and _UCLR_ have been assigned the special missing value *V*. This indicates that the quantities represented by these variables vary with the subgroup sample size.

Example 19.17: Specifying Axis Labels

NOTE: See *Median and Range Charts-Specifying Axis Labels* in the SAS/QC Sample Library.

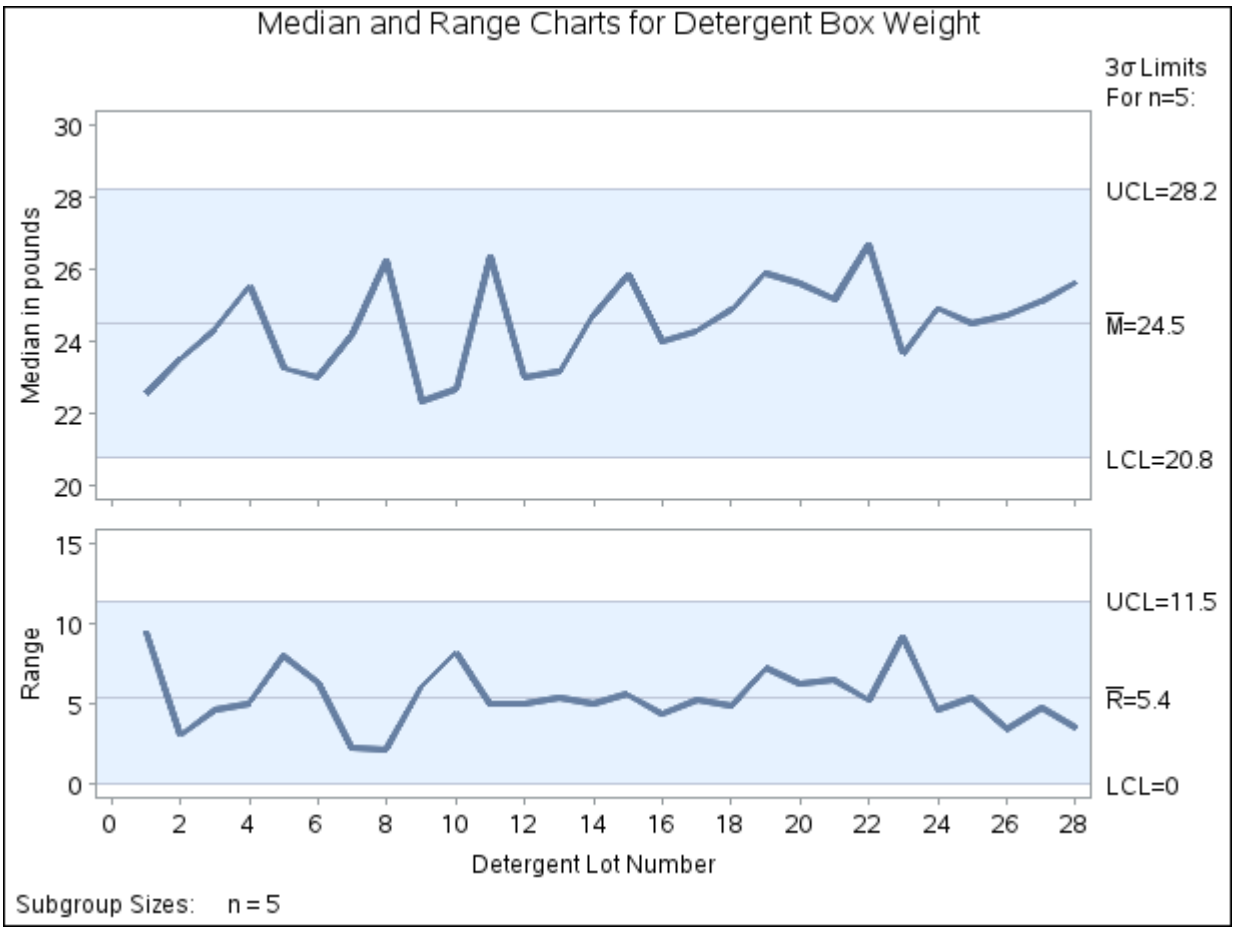
This example illustrates various methods for specifying axis labels and other axis features for median and range charts. For further details, see "Labeling Axes" on page 2109.

The charts in Figure 19.44, which are based on the data set Detergent introduced in the section "Getting Started: MRCHART Statement" on page 1604, display default labels for the horizontal and vertical axes. You can specify axis labels by associating labels with the *process* and *subgroup* variables as illustrated by the following statements:

```
ods graphics off;
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=Detergent;
   mrchart Weight*Lot / split   = '/';
   label Lot    = 'Detergent Lot Number'
         Weight = 'Median in pounds/Range';
run;
```

The charts are shown in Output 19.17.1. The horizontal axis label is the label associated with the *subgroup-variable* Lot. The vertical axis label for the median chart, referred to as the primary vertical axis label, is the first portion of the label associated with the *process* variable Weight, up to but not including the split character, which is specified with the SPLIT= option. The vertical axis label for the range chart, referred to as the secondary vertical axis label, is the second portion of the label associated with Weight.

Output 19.17.1 Customized Axis Labels Using Variable Labels



When the input data set is a HISTORY= data set, the vertical axis labels are determined by the label associated with the subgroup median variable. This is illustrated by the following statements, which use the data set Detsum introduced in "Creating Charts for Medians and Ranges from Summary Data" on page 1606:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart history=Detsum;
   mrchart Weight*Lot / split = '/';
   label Lot     = 'Detergent Lot Number'
         WeightM = 'Median (pounds)/Range';
run;
```

The charts are identical to those in Output 19.17.1.

When the input data set is a TABLE= data set, the vertical axis labels are determined by the label associated with the subgroup median variable _SUBMED_. This is illustrated by the following statements, which use the data set Dtable introduced in Figure 19.49:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart table=Dtable;
   mrchart Weight*Lot / split = '/';
   label Lot      = 'Detergent Lot Number'
         _submed_ = 'Median (pounds)/Range';
run;
```

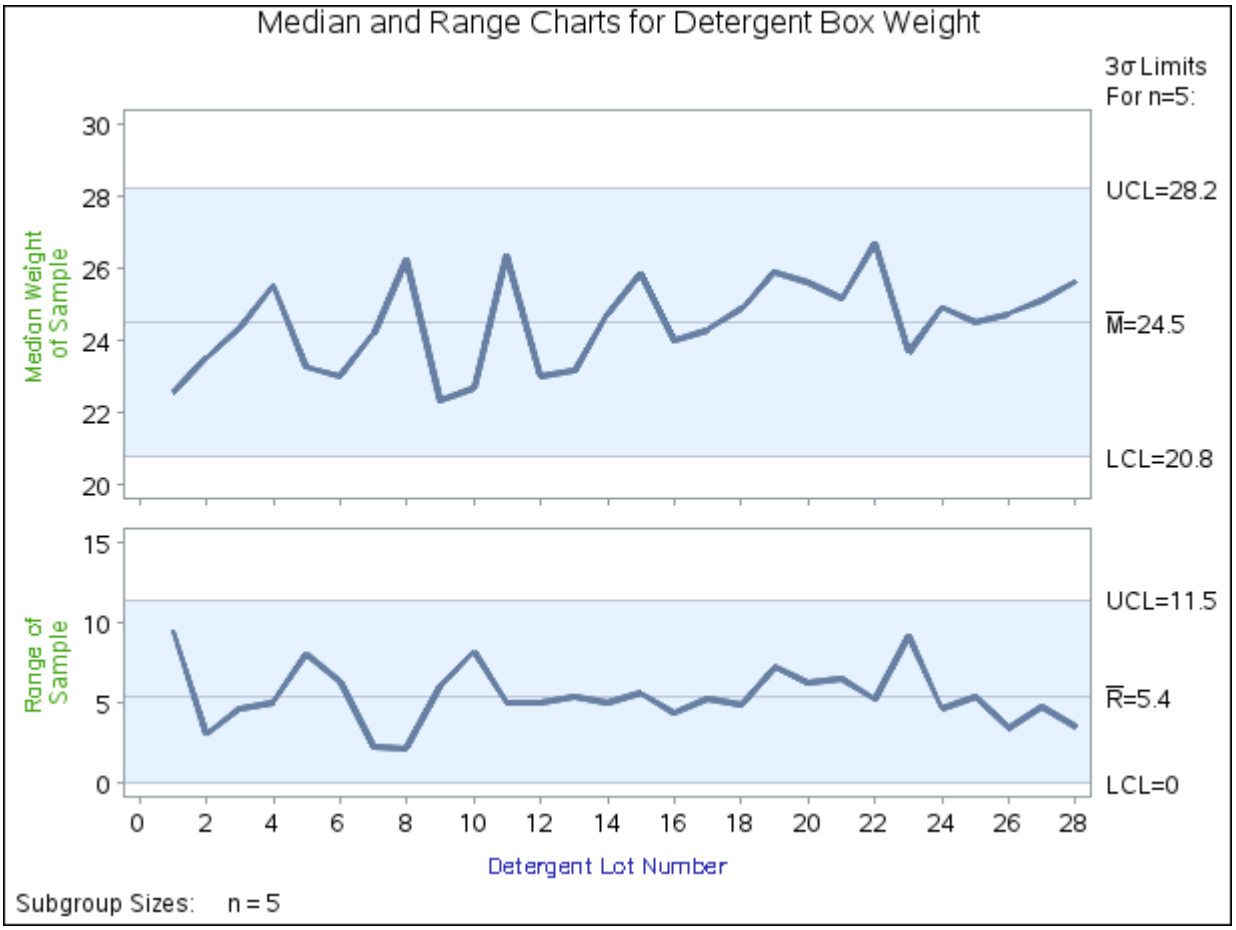
The charts are identical to those in Output 19.17.1.

When you are creating traditional graphics, you can use AXIS statements to enhance the appearance of the axes. This method is illustrated by the following statements:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=Detergent;
   mrchart Weight*Lot / haxis   = axis1
                        vaxis   = axis2
                        vaxis2  = axis3;
axis1 label=(c=bib  f=simplex 'Detergent Lot Number'          );
axis2 label=(c=vilg f=simplex 'Median Weight' j=c 'of Sample' );
axis3 label=(c=vilg f=simplex 'Range of'      j=c 'Sample'    );
run;
```

The charts are shown in Output 19.17.2.

Output 19.17.2 Customized Axis Labels Using AXIS Statements



You can use AXIS statements to customize a variety of axis features. For details, see *SAS/GRAPH: Reference*.

NPCHART Statement: SHEWHART Procedure

Overview: NPCHART Statement

The NPCHART statement creates *np* charts for the numbers of nonconforming (defective) items in subgroup samples.

You can use options in the NPCHART statement to

- compute control limits from the data based on a multiple of the standard error of the numbers of nonconforming items or as probability limits
- tabulate subgroup sample sizes, numbers of nonconforming items, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and proportions of nonconforming items in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) proportion of nonconforming items for computing control limits
- specify the data as counts, proportions, or percentages of nonconforming items
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *np* charts with the NPCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: NPCHART Statement

This section introduces the NPCHART statement with simple examples that illustrate commonly used options. Complete syntax for the NPCHART statement is presented in the section "Syntax: NPCHART Statement" on page 1656, and advanced examples are given in the section "Examples: NPCHART Statement" on page 1676.

Creating np Charts from Count Data

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

An electronics company manufactures circuits in batches of 500 and uses an *np* chart to monitor the number of failing circuits. Thirty batches are examined, and the failures in each batch are counted. The following statements create a SAS data set named Circuits,[5], which contains the failure counts:

```
data Circuits;
   input Batch Fail @@;
   datalines;
1    5   2   6   3  11   4   6    5   4
6    9   7  17   8  10   9  12   10   9
11   8  12   7  13   7  14  15   15   8
16  18  17  12  18  16  19   4   20   7
21  17  22  12  23   8  24   7   25  15
26   6  27   8  28  12  29   7   30   9
;
```

A partial listing of Circuits is shown in Figure 19.51.

Figure 19.51 The Data Set Circuits

Number of Failing Circuits

Batch	Fail
1	5
2	6
3	11
4	6
5	4

There is a single observation for each batch. The variable Batch identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Fail contains the number of nonconforming items in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *np* chart shown in Figure 19.52:

```
ods graphics off;
title 'np Chart for the Number of Failing Circuits';
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn = 500;
run;
```

[5]This data set is also used in the "Getting Started" section of "PCHART Statement: SHEWHART Procedure" on page 1686

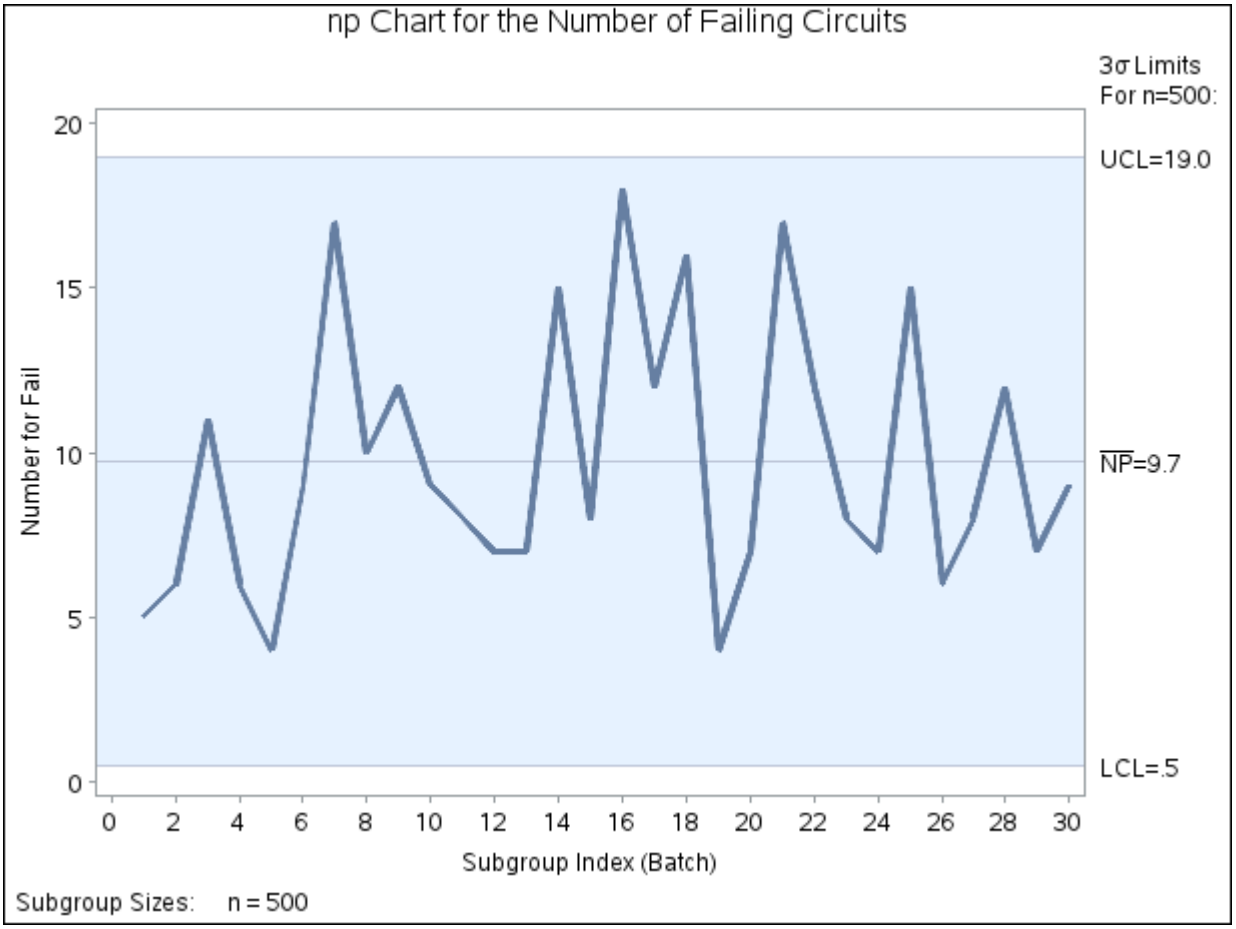
This example illustrates the basic form of the NPCHART statement. After the keyword NPCHART, you specify the *process* to analyze (in this case, Fail), followed by an asterisk and the *subgroup-variable* (Batch).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. The SUBGROUPN= option specifies the number of items in each subgroup sample and is required with a DATA= input data set. The SUBGROUPN= option specifies one of the following:

- a constant subgroup sample size (in this case)
- a variable in the input data set whose values provide the subgroup sample sizes (see the next example)

Options such as SUBGROUPN= are specified after the slash (/) in the NPCHART statement. A complete list of options is presented in the section "Syntax: NPCHART Statement" on page 1656.

Figure 19.52 *np* Chart for Circuit Failures (Traditional Graphics)



Each point on the *np* chart represents the number of nonconforming items for a particular subgroup. For instance, the value plotted for the first batch is 5.

Because all the points fall within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1668. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1655. For computational details, see "Constructing Charts for Number Nonconforming (np Charts)" on page 1667. For more details on reading raw data, see "DATA= Data Set" on page 1672.

Creating np Charts from Summary Data

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create *np* charts using raw data (counts of nonconforming items). However, in many applications, the data are provided in summarized form as proportions or percentages of nonconforming items. This example illustrates how you can use the NPCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data Cirprop;
   input Batch pFailed @@;
   sizes=500;
   datalines;
 1  0.010   2  0.012   3  0.022   4  0.012   5  0.008
 6  0.018   7  0.034   8  0.020   9  0.024  10  0.018
11  0.016  12  0.014  13  0.014  14  0.030  15  0.016
16  0.036  17  0.024  18  0.032  19  0.008  20  0.014
21  0.034  22  0.024  23  0.016  24  0.014  25  0.030
26  0.012  27  0.016  28  0.024  29  0.014  30  0.018
;
```

A partial listing of Cirprop is shown in Figure 19.53. The subgroups are still indexed by Batch. The variable pFailed contains the proportions of nonconforming items, and the variable Sampsize contains the subgroup sample sizes.

Figure 19.53 The Data Set Cirprop

Subgroup Proportions of Nonconforming Items

Batch	pFailed	sizes
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500

The following statements create an *np* chart identical to the one in Figure 19.52:

```
title 'np Chart for the Number of Failing Circuits';
proc shewhart data=Cirprop;
   npchart pFailed*Batch / subgroupn=Sampsize
                           dataunit =proportion;
label pFailed = 'Number of FAIL';
run;
```

The DATAUNIT= option specifies that the values of the *process* (pFailed) are proportions of nonconforming items. By default, the values of the *process* are assumed to be counts of nonconforming items (see the previous example).

Alternatively, you can read the data set Cirprop by specifying it as a HISTORY= data set in the PROC SHEWHART statement. A HISTORY= data set used with the NPCHART statement must contain the following variables:

- subgroup variable
- subgroup proportion of nonconforming items variable
- subgroup sample size variable

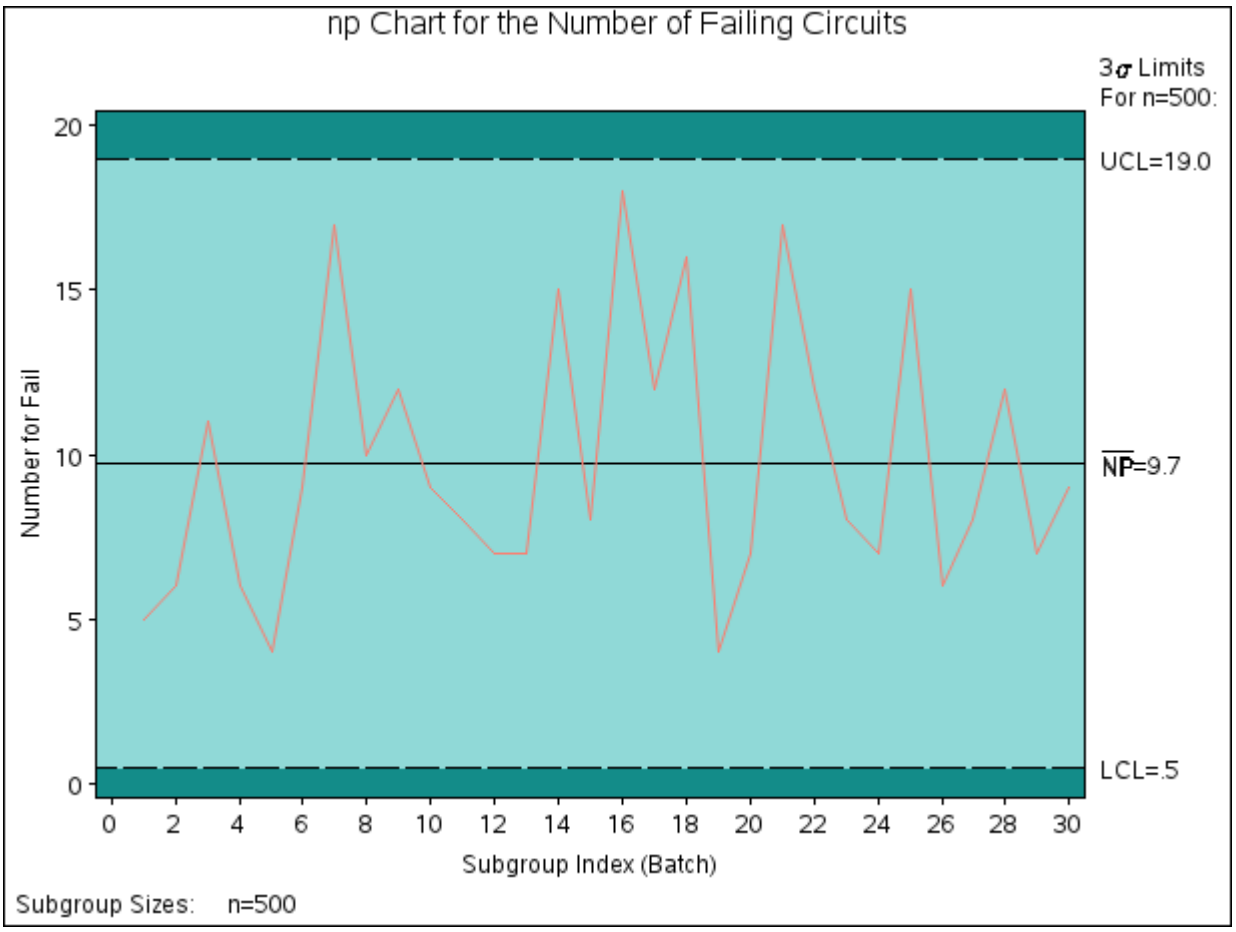
Furthermore, the names of the subgroup proportion and sample size variables must begin with the *process* name specified in the NPCHART statement and end with the special suffix characters *P* and *N*, respectively.

To specify Cirprop as a HISTORY= data set and Fail as the *process*, you must rename the variables pFailed and Sampsize to FailP and FailN, respectively. The following statements temporarily rename pFailed and Sampsize for the duration of the procedure step:

```
options nogstyle;
goptions ftext='albany amt';
title 'np Chart for the Number of Failing Circuits';
proc shewhart history=Cirprop(rename=(pFailed =FailP
                                 sizes=FailN ));
   npchart Fail*Batch / cframe    = vibg
                        cinfill   = vlibg
                        coutfill  = salmon
                        cconnect  = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the NPCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *np* chart is shown in Figure 19.54.

Figure 19.54 *np* Chart for Circuit Failures (Traditional Graphics with NOGSTYLE)



In this example, it is more convenient to use Cirprop as a DATA= data set than as a HISTORY= data set. As illustrated in the next example, it is generally more convenient to use the HISTORY= option for input data sets that have been created previously by the SHEWHART procedure as OUTHISTORY= data sets.

For more information, see "HISTORY= Data Set" on page 1674.

Saving Proportions of Nonconforming Items

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

In this example, the NPCHART statement is used to create a data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read the number of nonconforming items from the data set Circuits (see "Creating np Charts from Count Data" on page 1647) and create a summary data set named Cirhist:

```
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn  = 500
                        outhistory = Cirhist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.52. Figure 19.55 contains a partial listing of Cirhist.

Figure 19.55 The Data Set Cirhist

Subgroup Proportions of Failing Circuits

Batch	FailP	FailN
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500

There are three variables in the data set Cirhist.

- Batch contains the subgroup index.
- FailP contains the subgroup proportion of nonconforming items.
- FailN contains the subgroup sample size.

Note that the variables containing the subgroup proportions of nonconforming items and subgroup sample sizes are named by adding the suffix characters *P* and *N* to the *process* Fail specified in the NPCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1670.

Saving Control Limits

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for an *np* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1655) or modify the limits with a DATA step program.

The following statements read the number of nonconforming items per subgroup from the data set Circuits (see "Creating np Charts from Count Data" on page 1647) and save the control limits displayed in Figure 19.52 in a data set named Cirlim:

```
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn=500
                        outlimits=Cirlim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Cirlim is listed in Figure 19.56.

Figure 19.56 The Data Set Cirlim Containing Control Limit Information

Control Limits for the Number of Failing Circuits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_P_	_LCLNP_	_NP_	_UCLNP_
Fail	Batch	ESTIMATE	500	.002320877	3	0.019467	0.46539	9.73333	19.0013

The data set Cirlim contains one observation with the limits for *process* Fail. The variables _LCLNP_ and _UCLNP_ contain the lower and upper control limits, and the variable _NP_ contains the central line. The variable _P_ contains the average proportion of nonconforming items. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or a standard value.

For more information, see "OUTLIMITS= Data Set" on page 1669.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn=500
                        outtable=Cirtable
                        nochart;
run;
```

The Cirtable data set contains one observation for each subgroup sample. The variables _SUBNP_ and _SUBN_ contain the subgroup numbers of nonconforming items and subgroup sample sizes, respectively. The variables _LCLNP_ and _UCLNP_ contain the lower and upper control limits, and the variable _NP_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1671.

The data set Cirtable is listed in Figure 19.57.

Figure 19.57 The Data Set Cirtable

Number Nonconforming and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLNP_	_SUBNP_	_NP_	_UCLNP_	_EXLIM_
Fail	1	3	500	500	0.46539	5	9.73333	19.0013	
Fail	2	3	500	500	0.46539	6	9.73333	19.0013	
Fail	3	3	500	500	0.46539	11	9.73333	19.0013	
Fail	4	3	500	500	0.46539	6	9.73333	19.0013	
Fail	5	3	500	500	0.46539	4	9.73333	19.0013	
Fail	6	3	500	500	0.46539	9	9.73333	19.0013	
Fail	7	3	500	500	0.46539	17	9.73333	19.0013	
Fail	8	3	500	500	0.46539	10	9.73333	19.0013	
Fail	9	3	500	500	0.46539	12	9.73333	19.0013	
Fail	10	3	500	500	0.46539	9	9.73333	19.0013	
Fail	11	3	500	500	0.46539	8	9.73333	19.0013	
Fail	12	3	500	500	0.46539	7	9.73333	19.0013	
Fail	13	3	500	500	0.46539	7	9.73333	19.0013	
Fail	14	3	500	500	0.46539	15	9.73333	19.0013	
Fail	15	3	500	500	0.46539	8	9.73333	19.0013	
Fail	16	3	500	500	0.46539	18	9.73333	19.0013	
Fail	17	3	500	500	0.46539	12	9.73333	19.0013	
Fail	18	3	500	500	0.46539	16	9.73333	19.0013	
Fail	19	3	500	500	0.46539	4	9.73333	19.0013	
Fail	20	3	500	500	0.46539	7	9.73333	19.0013	
Fail	21	3	500	500	0.46539	17	9.73333	19.0013	
Fail	22	3	500	500	0.46539	12	9.73333	19.0013	
Fail	23	3	500	500	0.46539	8	9.73333	19.0013	
Fail	24	3	500	500	0.46539	7	9.73333	19.0013	
Fail	25	3	500	500	0.46539	15	9.73333	19.0013	
Fail	26	3	500	500	0.46539	6	9.73333	19.0013	
Fail	27	3	500	500	0.46539	8	9.73333	19.0013	
Fail	28	3	500	500	0.46539	12	9.73333	19.0013	
Fail	29	3	500	500	0.46539	7	9.73333	19.0013	
Fail	30	3	500	500	0.46539	9	9.73333	19.0013	

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Cirtable and display an *np* chart (not shown here) identical to the chart in Figure 19.52:

```
title 'np Chart for the Number of Failing Circuits';
proc shewhart table=Cirtable;
   npchart Fail*Batch;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1675.

Reading Preestablished Control Limits

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Cirlim saved control limits computed from the data in Circuits. This example shows how these limits can be applied to new data provided in the following data set:

```
data Circuit2;
   input Batch Fail;
   datalines;
31  12   32   9   33  16   34   9
35   3   36   8   37  20   38   4
39   8   40   6   41  12   42  16
43   9   44   2   45  10   46   8
47  14   48  10   49  11   50   9
;
```

The following statements create an *np* chart for the data in Circuit2 using the control limits in Cirlim:

```
ods graphics on;
title 'np Chart for the Proportion of Failing Circuits';
proc shewhart data=Circuit2 limits=Cirlim;
   npchart Fail*Batch / subgroupn = 500
                        odstitle  = title;
run;
```

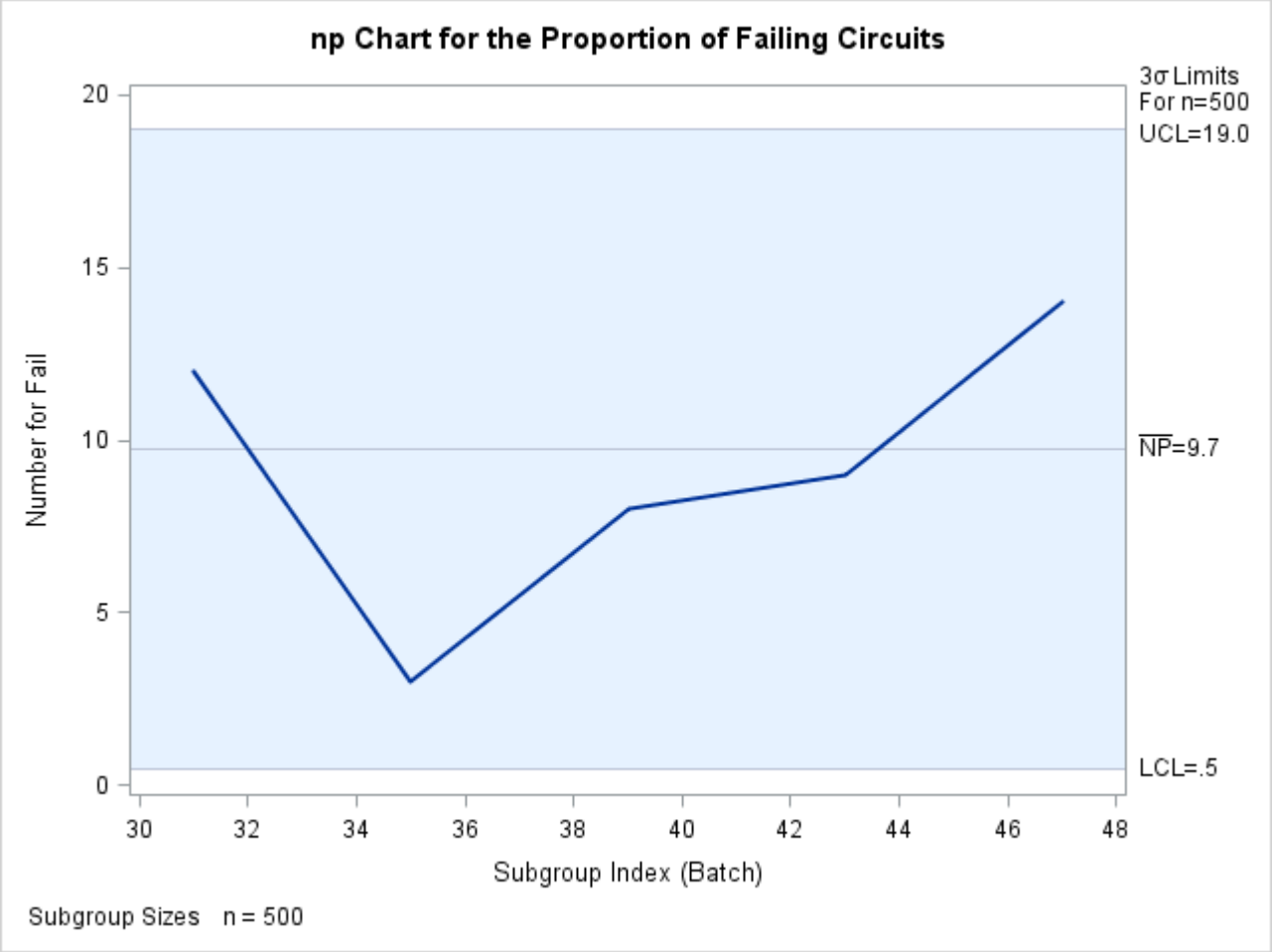
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *np* chart is created using ODS Graphics instead of traditional graphics.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Fail
- the value of _SUBGRP_ matches the *subgroup-variable* name Batch

The resulting *np* chart is shown in Figure 19.58.

Figure 19.58 np Chart for Second Set of Circuit Failures (ODS Graphics)



The number of nonconforming items in the 37th batch exceeds the upper control limit, signaling that the process is out of control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step; see Example 19.21 for an example. See "LIMITS= Data Set" on page 1673 for details concerning the variables that you must provide.

Syntax: NPCHART Statement

The basic syntax for the NPCHART statement is as follows:

NPCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

NPCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of NPCHART statements in the SHEWHART procedure. The components of the NPCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconforming items are read from a DATA= data set, *process* must be the name of the variable containing the numbers. For an example, see "Creating np Charts from Count Data" on page 1647.
- If proportions of nonconforming items are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating np Charts from Summary Data" on page 1649.
- If numbers of nonconforming items and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1652.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *np* charts for Rejects and Reworks:

```
proc shewhart data=Measures;
   npchart (Rejects Reworks)*Sample / subgroupn=100;
run;
```

Note that when data are read from a DATA= data set, the SUBGROUPN= option, which specifies subgroup sample sizes, is required.

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding NPCHART statement, Sample is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot numbers of nonconforming items.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create an *np* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   npchart Rejects*Day='*' / subgroupn=100;
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" on page 1658 lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the NPCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.37 NPCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	Requests probability limits at discrete values
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	Displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit

Table 19.37 *continued*

Option	Description
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMIT0	Suppresses display of lower control limit if it is 0
NOLIMIT1	Suppresses display of upper control limit if it is 1 (100%)
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
NPSYMBOL=	Specifies label for central line
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
Standard Value Options	
P0=	Specifies known (standard) value p_0 for proportion of nonconforming items
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on np chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits

Table 19.37 *continued*

Option	Description
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive

Table 19.37 *continued*

Option	Description
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages

Table 19.37 *continued*

Option	Description
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays np chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= options
CVREF=	Specifies color for lines requested by VREF= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart

Table 19.37 *continued*

Option	Description
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features chart
DESCRIPTION=	Specifies description of *np* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of *np* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency

Table 19.37 *continued*

Option	Description
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
DATAUNIT	Specifies that input values are proportions or percentages (rather than counts) of nonconforming items
MISSBREAK	Specifies that observations with missing values are not to be processed
SUBGROUPN	Specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set

Table 19.37 *continued*

Option	Description
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend

Table 19.37 *continued*

Option	Description
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments

Table 19.37 *continued*

Option	Description
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: NPCHART Statement

The following sections provide details that are specific to the NPCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Number Nonconforming (np Charts)

The following notation is used in this section:

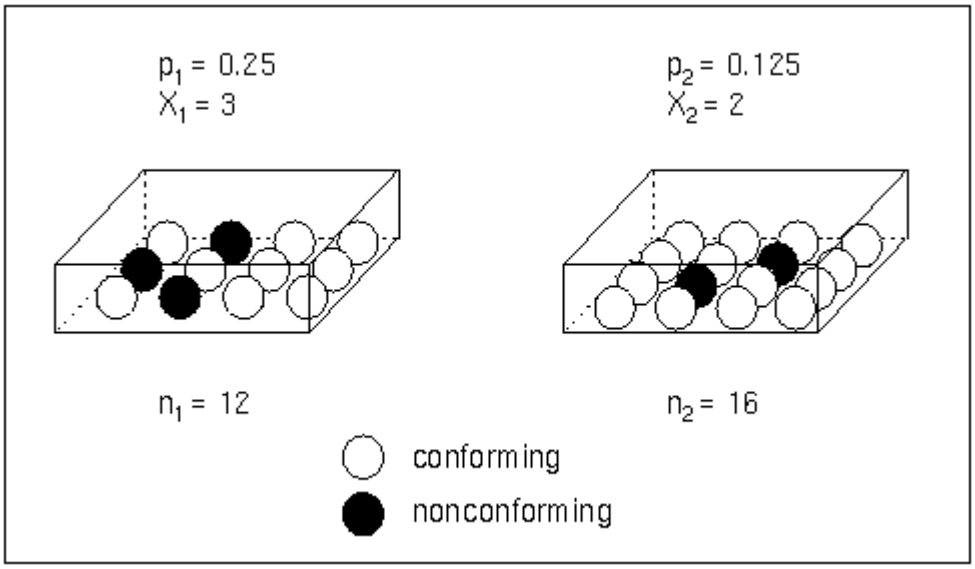
p	Expected proportion of nonconforming items produced by the process
p_i	Proportion of nonconforming items in the ith subgroup
X_i	Number of nonconforming items in the ith subgroup
n_i	Number of items in the ith subgroup
$\bar{p}$	Average proportion of nonconforming items taken across subgroups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_N p_N}{n_1 + \cdots + n_N} = \frac{X_1 + \cdots + X_N}{n_1 + \cdots + n_N}$$
N	Number of subgroups

$I_T(\alpha, \beta)$	Incomplete beta function: $$I_T(\alpha,\beta) = (\Gamma(\alpha+\beta)/\Gamma(\alpha)\Gamma(\beta)) \int_0^T t^{\alpha-1}(1-t)^{\beta-1}dt$$ for $0 < T < 1$, $\alpha > 0$, and $\beta > 0$, where $\Gamma(\cdot)$ is the gamma function

Plotted Points

Each point on an np chart represents the observed number (X_i) of nonconforming items in a subgroup. For example, suppose the first subgroup (see Figure 19.59) contains 12 items, of which three are nonconforming. The point plotted for the first subgroup is $X_1 = 3$.

Figure 19.59 Proportions Versus Counts



Note that a p chart displays the proportion of nonconforming items p_i. You can use the PCHART statement to create p charts; see "PCHART Statement: SHEWHART Procedure" on page 1686.

Central Line

By default, the central line on an np chart indicates an estimate for $n_i\,p$, which is computed as $n_i\,\bar{p}$. If you specify a known value (p_0) for p, the central line indicates the value of $n_i\,p_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of X_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that X_i exceeds the limits

The lower and upper control limits, LCL and UCL respectively, are computed as

$$\text{LCL} = \max\left(n_i\,\bar{p} - k\sqrt{n_i\,\bar{p}(1-\bar{p})}\,, 0\right)$$
$$\text{UCL} = \min\left(n_i\,\bar{p} + k\sqrt{n_i\,\bar{p}(1-\bar{p})}\,, n_i\right)$$

A lower probability limit for X_i can be determined using the fact that

$$\begin{aligned} P\{X_i < \text{LCL}\} &= 1 - P\{X_i \geq \text{LCL}\} \\ &= 1 - I_{\bar{p}}(\text{LCL}, n_i + 1 - \text{LCL}) \\ &= I_{1-\bar{p}}(n_i + 1 - \text{LCL}, \text{LCL}) \end{aligned}$$

Refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that X_i is binomially distributed. The lower probability limit LCL is then calculated by setting

$$I_{1-\bar{p}}(n_i + 1 - \text{LCL}, \text{LCL}) = \alpha/2$$

and solving for LCL. Similarly, the upper probability limit for X_i can be determined using the fact that

$$\begin{aligned} P\{X_i > \text{UCL}\} &= P\{X_i > \text{UCL}\} \\ &= I_{\bar{p}}(\text{UCL} + 1, n_i - \text{UCL}) \end{aligned}$$

The upper probability limit UCL is then calculated by setting

$$I_{\bar{p}}(\text{UCL} + 1, n_i - \text{UCL}) = \alpha/2$$

and solving for UCL. The probability limits are asymmetric about the central line. Note that both the control limits and probability limits vary with n_i.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify p_0 with the P0= option or with the variable _P_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.39 lists the variables that can be saved.

Table 19.39 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLNP	Lower control limit for number of nonconforming items
LIMITN	Sample size associated with the control limits
_NP_	Average number of nonconforming items ($n_i\,\bar{p}$ or $n_i\,p_0$)
_P_	Average proportion of nonconforming items ($\bar{p}$ or p_0)
SIGMAS	Multiple (k) of standard error of X_i
SUBGRP	*Subgroup-variable* specified in the NPCHART statement
TYPE	Type (standard or estimate) of _NP_
UCLNP	Upper control limit for number of nonconforming items
VAR	*Process* specified in the NPCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLNP_, _UCLNP_, _NP_, and _SIGMAS_.
2. If the limits are defined in terms of a multiple k of the standard error of X_i, the value of _ALPHA_ is computed as $\alpha = P\{X_i < \text{\_LCLNP\_}\} + P\{X_i > \text{\_UCLNP\_}\}$, using the incomplete beta function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = (\text{\_UCLNP\_} - \text{\_NP\_})/\sqrt{\text{\_NP\_}(1 - \text{\_NP\_})/\text{\_LIMITN\_}}$. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the NPCHART statement. For an example, see "Saving Control Limits" on page 1652.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- the subgroup proportion of nonconforming items variable named by the *process* suffixed with *P*
- a subgroup sample size variable named by the *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the NPCHART statement.

For example, consider the following statements:

```
proc shewhart data=Input;
   npchart (Rework Rejected)*Batch / outhistory=Summary
                                     subgroupn =30;
run;
```

The data set Summary contains variables named Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Proportions of Nonconforming Items" on page 1651.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.40 lists the variables that are saved.

Table 19.40 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on np chart
LCLNP	Lower control limit for number of nonconforming items
LIMITN	Nominal sample size associated with the control limits
_NP_	Average number of nonconforming items
SIGMAS	Multiple (k) of the standard error of X_i associated with the control limits
Subgroup	Values of the subgroup variable
SUBNP	Subgroup number of nonconforming items
SUBN	Subgroup sample size
TESTS	Tests for special causes signaled on np chart
UCLNP	Upper control limit for number of nonconforming items
VAR	*Process* specified in the NPCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*

- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the tests appropriate for np charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1652.

Input Data Sets

DATA= Data Set

You can read raw data (counts of nonconforming items) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the NPCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the NPCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *process* and a value for the *subgroup-variable*. The data set must contain one observation for each subgroup. Note that you can specify the DATAUNIT= option in the NPCHART statement to read proportions or percentages of nonconforming items instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the NPCHART statement, the SUBGROUPN= option (which specifies the subgroup sample size) is required. By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating np Charts from Count Data" on page 1647.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   npchart Rejects*Batch  / subgroupn=100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLNP_, _NP_, and _UCLNP_, which specify the control limits directly
- the variable _P_, which is used to calculate the control limits according to the equations in the section "Control Limits" on page 1668

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1655.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set.

A HISTORY= data set used with the NPCHART statement must contain

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable for each *process*
- a subgroup sample size variable for each *process*

The names of the proportion sample size variables must be the *process* name concatenated with the special suffix characters *P* and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   npchart ( Rework Rejected)*Batch / subgroupn=50;
run;
```

The data set Summary must include the variables Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating np Charts from Summary Data" on page 1649.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.41 lists the variables required in a TABLE= data set used with the NPCHART statement.

Table 19.41 Variables Required in a TABLE= Data Set

Variable	Description
LCLNP	Lower control limit for number of nonconforming items
LIMITN	Nominal sample size associated with the control limits
_NP_	Average number of nonconforming items
Subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBNP	Subgroup number of nonconforming items
UCLNP	Upper control limit for number of nonconforming items

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1652.

Examples: NPCHART Statement

This section provides advanced examples of the NPCHART statement.

Example 19.18: Applying Tests for Special Causes

NOTE: See *np Charts-Tests for Special Causes* in the SAS/QC Sample Library.

This example shows how you can apply tests for special causes to make *np* charts more sensitive to special causes of variation. The following statements create a SAS data set named Circuit3, which contains the number of failing circuits for 20 batches from the circuit manufacturing process introduced in the section "Creating np Charts from Count Data" on page 1647:

```
data Circuit3;
   input Batch Fail @@;
   datalines;
 1 12    2 21    3 16    4  9
 5  3    6  4    7  6    8  9
 9 11   10 13   11 12   12  7
13  2   14 14   15  9   16  8
17 14   18 10   19 11   20  9
;
```

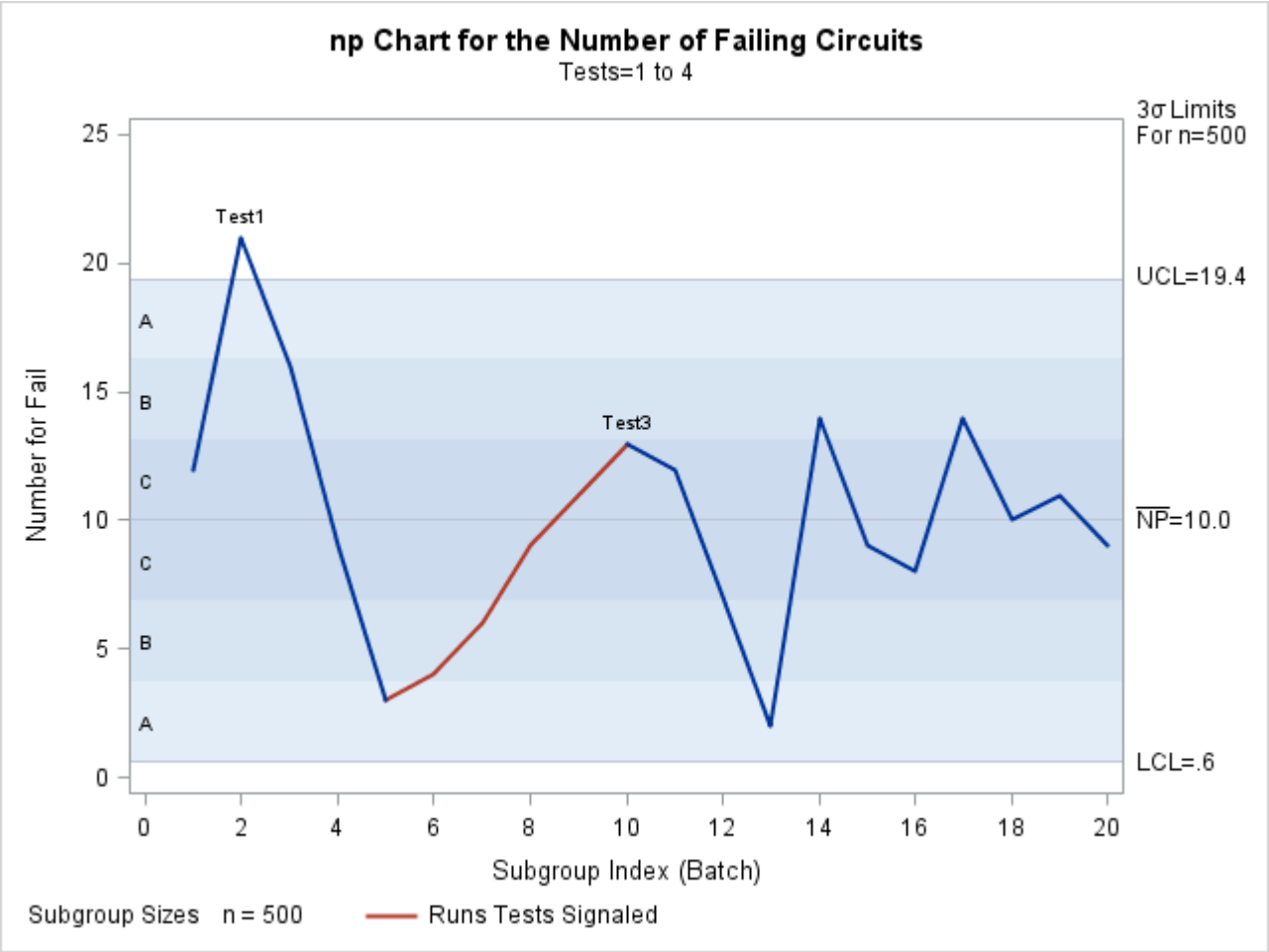
The following statements create the *np* chart, apply several tests to the chart, and tabulate the results:

```
ods graphics on;
title1'np Chart for the Number of Failing Circuits';
title2 'Tests=1 to 4';
proc shewhart data=Circuit3;
   npchart Fail*Batch / subgroupn = 500
                        tests=1 to 4
                        table
                        tabletest
                        tablelegend
                        zones
                        zonelabels
                        odstitle  = title
                        odstitle2 = title2;
run;
```

The chart is shown in Output 19.18.1, and the printed output is shown in Output 19.18.2. The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. The TABLETESTS option requests a table of counts of nonconforming items and control limits, with a column indicating which subgroups tested positive for special causes. The TABLELEGEND option adds a legend describing the tests. The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests.

Output 19.18.1 and Output 19.18.2 indicate that Test 1 is positive at batch 2 and Test 3 is positive at batch 10.

Output 19.18.1 Tests for Special Causes Displayed on np Chart



Output 19.18.2 Tabular Form of np Chart

np Chart for the Number of Failing Circuits Tests=1 to 4

The SHEWHART Procedure

np Chart Summary for Fail

Batch	Subgroup Sample Size	3 Sigma Limits with n=500 for Number Lower Limit	Subgroup Number	Upper Limit	Special Tests Signaled
1	500	0.60851449	12.000000	19.391486	
2	500	0.60851449	21.000000	19.391486	1
3	500	0.60851449	16.000000	19.391486	
4	500	0.60851449	9.000000	19.391486	
5	500	0.60851449	3.000000	19.391486	
6	500	0.60851449	4.000000	19.391486	
7	500	0.60851449	6.000000	19.391486	
8	500	0.60851449	9.000000	19.391486	
9	500	0.60851449	11.000000	19.391486	
10	500	0.60851449	13.000000	19.391486	3
11	500	0.60851449	12.000000	19.391486	
12	500	0.60851449	7.000000	19.391486	
13	500	0.60851449	2.000000	19.391486	
14	500	0.60851449	14.000000	19.391486	
15	500	0.60851449	9.000000	19.391486	
16	500	0.60851449	8.000000	19.391486	
17	500	0.60851449	14.000000	19.391486	
18	500	0.60851449	10.000000	19.391486	
19	500	0.60851449	11.000000	19.391486	
20	500	0.60851449	9.000000	19.391486	

Test Descriptions	
Test 1	One point beyond Zone A (outside control limits)
Test 3	Six points in a row steadily increasing or decreasing

Example 19.19: Specifying Standard Average Proportion

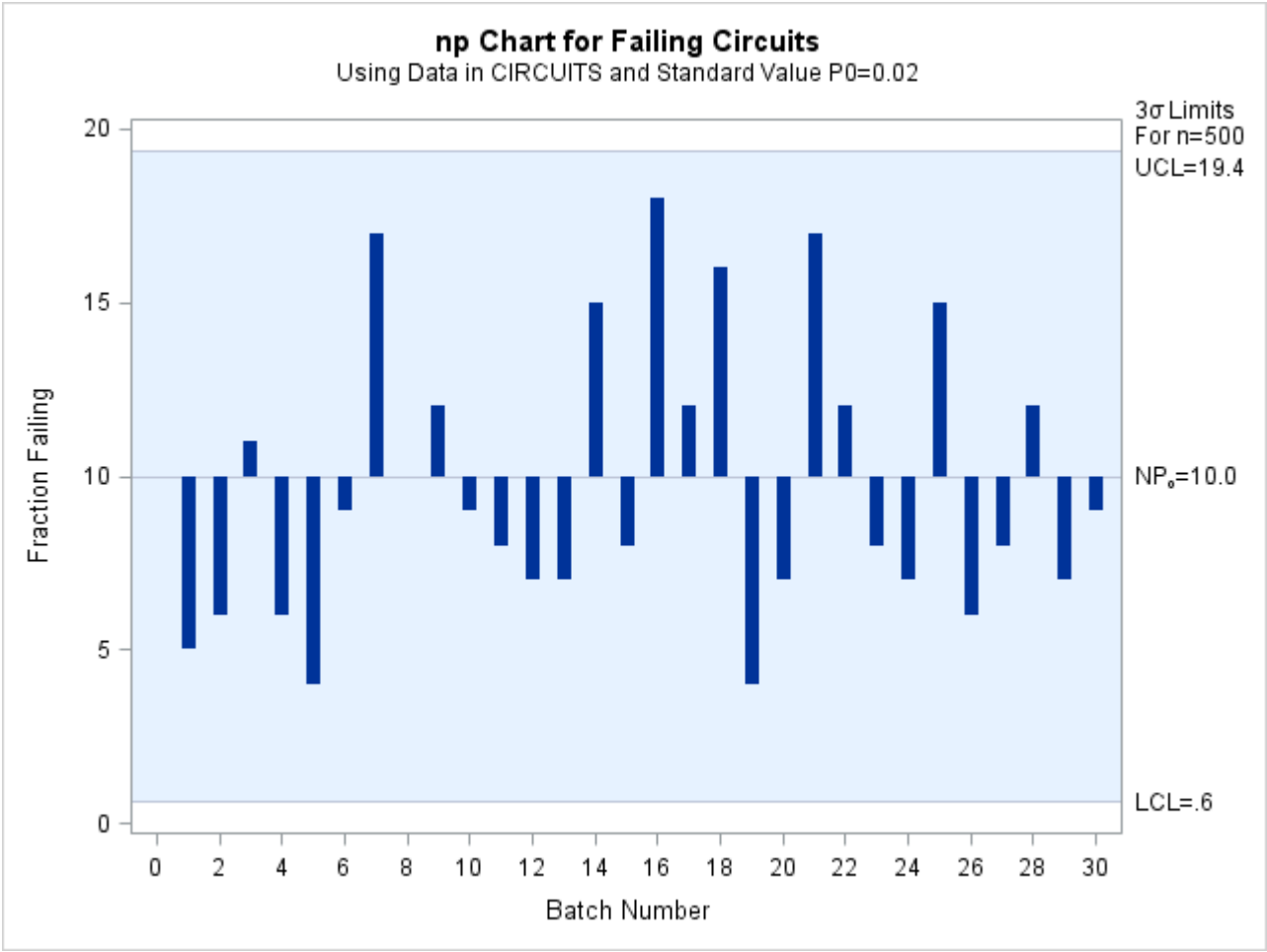
NOTE: See *Specifying a Known Proportion for np Charts* in the SAS/QC Sample Library.

In some situations, a standard (known) value (p_0) is available for the expected proportion of nonconforming items, based on extensive testing or previous sampling. This example illustrates how you can specify p_0 to create an np chart.

An np chart is used to monitor the number of failing circuits in the data set Circuits, which is introduced in "Creating np Charts from Count Data" on page 1647. The expected proportion of failing circuits is known to be $p_0 = 0.02$. The following statements create an np chart, shown in Output 19.19.1, using p_0 to compute the control limits:

```
ods graphics on;
title1 'np Chart for Failing Circuits';
title2 'Using Data in CIRCUITS and Standard Value P0=0.02';
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn = 500
                        p0        = 0.02
                        npsymbol  = np0
                        nolegend
                        needles
                        odstitle  = title
                        odstitle2 = title2;
   label Batch ='Batch Number'
         Fail  ='Fraction Failing';
run;
```

Output 19.19.1 An np Chart with Standard Value of p_0



The chart indicates that the process is in control. The P0= option specifies p_0. The NPSYMBOL= option specifies a label for the central line indicating that the line represents a standard value. The NEEDLES option connects points to the central line with vertical needles. The NOLEGEND option suppresses the default

legend for subgroup sample sizes. Labels for the vertical and horizontal axes are provided with the LABEL statement. For details concerning axis labeling, see "Axis Labels" on page 1973.

Alternatively, you can specify p_0 using the variable _P_ in a LIMITS= data set, as follows:

```
data Climits;
   length _var_ _subgrp_ _type_ $8;
   _p_      = 0.02;
   _subgrp_ = 'Batch';
   _var_    = 'Fail';
   _type_   = 'STANDARD';
   _limitn_ = 500;

proc shewhart data=Circuits limits=Climits;
   npchart Fail*Batch / subgroupn = 500
                        npsymbol  = np0
                        nolegend
                        needles;
   label Batch ='Batch Number'
         Fail  ='Fraction Failing';
run;
```

The bookkeeping variable _TYPE_ indicates that _P_ has a standard value. The chart produced by these statements is identical to the chart in Output 19.19.1.

Example 19.20: Working with Unequal Subgroup Sample Sizes

NOTE: See *np Charts with Unequal Subgroup Sample Sizes* in the SAS/QC Sample Library.

The following statements create a SAS data set named Battery, which contains the number of alkaline batteries per lot failing an acceptance test. The number of batteries tested in each lot varies but is approximately 150.

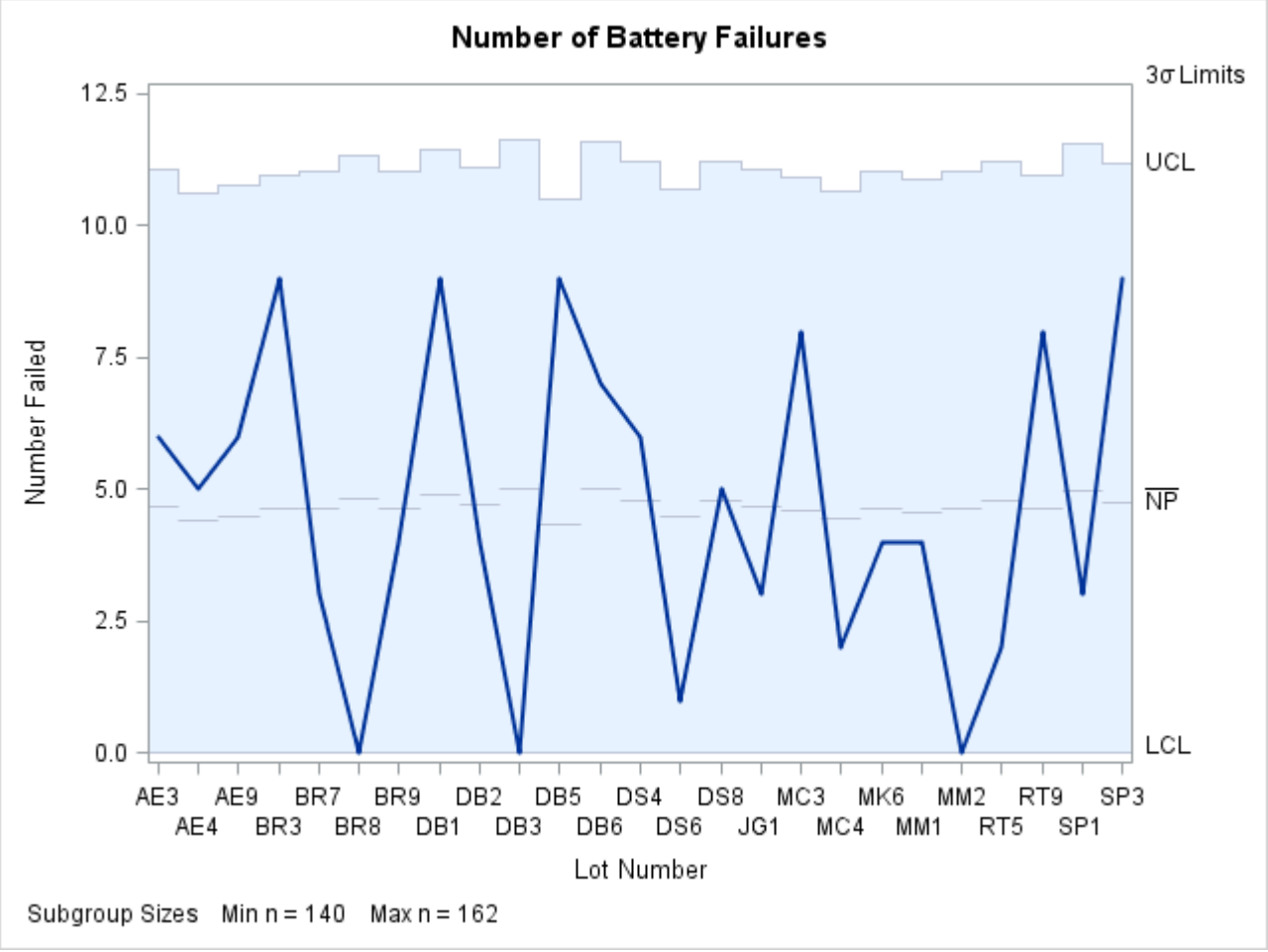
```
data Battery;
   length Lot $3;
   input Lot nFailed Sampsize @@;
   label nFailed ='Number Failed'
         Lot     = 'Lot Number'
         Sampsize='Number Sampled';
   datalines;
AE3  6  151     AE4  5  142     AE9  6  145
BR3  9  149     BR7  3  150     BR8  0  156
BR9  4  150     DB1  9  158     DB2  4  152
DB3  0  162     DB5  9  140     DB6  7  161
DS4  6  154     DS6  1  144     DS8  5  154
JG1  3  151     MC3  8  148     MC4  2  143
MK6  4  150     MM1  4  147     MM2  0  150
RT5  2  154     RT9  8  149     SP1  3  160
SP3  9  153
;
```

The variable nFailed contains the number of battery failures, the variable Lot contains the lot number, and the variable Sampsize contains the lot sample size. The following statements request an np chart for this data:

```
ods graphics on;
title 'Number of Battery Failures';
proc shewhart data=Battery;
   npchart nFailed*Lot / subgroupn = Sampsize
                         outlimits = Batlim
                         odstitle  = title;
   label nFailed='Number Failed';
run;
```

The chart is shown in Output 19.20.1, and the OUTLIMITS= data set Batlim is listed in Output 19.20.2.

Output 19.20.1 An np Chart with Varying Subgroup Sample Sizes



Note that the upper control limit and central line on the np chart vary with the subgroup sample size. The lower control limit is truncated at zero. The sample size legend indicates the minimum and maximum subgroup sample sizes.

Output 19.20.2 The Control Limits Data Set Batlim

Control Limits for Battery Failures

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_P_	_LCLNP_	_NP_	_UCLNP_
nFailed	Lot	ESTIMATE	V	V	3	0.031010	V	V	V

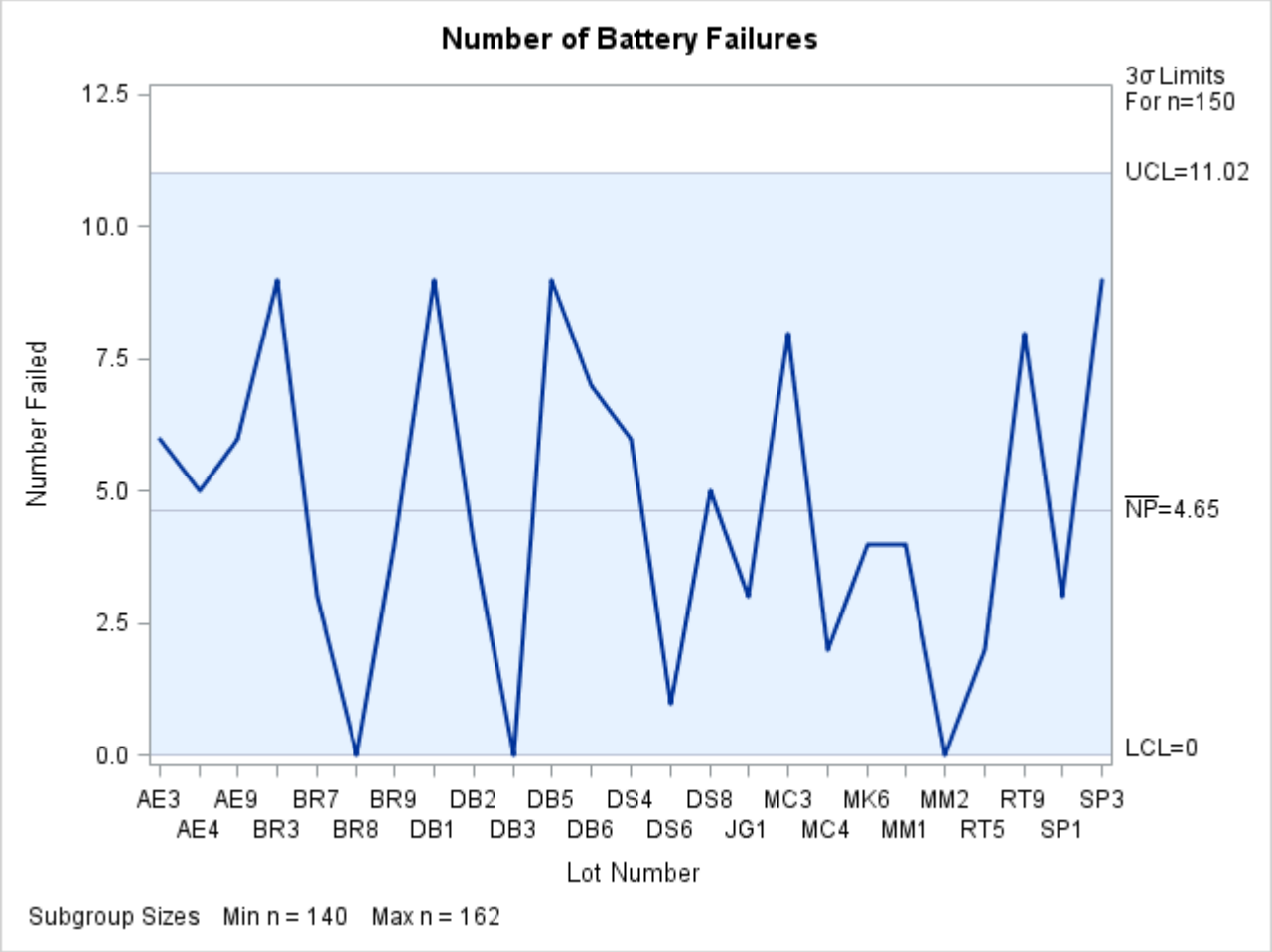
The variables in Batlim whose values vary with subgroup sample size are assigned the special missing value *V*.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing the control limits, as illustrated by the following statements:

```
title 'Number of Battery Failures';
proc shewhart data=Battery;
   npchart nFailed*Lot / subgroupn = Sampsize
                         limitn    = 150
                         odstitle  = title
                         alln;
   label nFailed='Number Failed';
run;
```

The ALLN option specifies that all points (regardless of subgroup sample size) are to be displayed. By default, only points for subgroups whose sample size matches the LIMITN= value are displayed. The chart is shown in Output 19.20.3.

Output 19.20.3 Control Limits Based on Fixed Subgroup Sample Size



All the points are inside the control limits, indicating that the process is in statistical control. Because there is relatively little variation in the sample sizes, the control limits in Output 19.20.3 provide a close approximation to the exact control limits in Output 19.20.1, and the same conclusions can be drawn from both charts. In general, you should be careful when interpreting charts that use a nominal sample size to compute control limits, because these limits are only approximate when the sample sizes vary.

Example 19.21: Specifying Control Limit Information

NOTE: See *np Charts-Specifying Control Limit Info* in the SAS/QC Sample Library.

This example shows how to use the DATA step to create LIMITS= data sets for use with the NPCHART statement. The variables _VAR_ and _SUBGRP_ are required. These variables must be character variables whose lengths are no greater than 32, and their values must match the *process* and *subgroup-variable* specified in the NPCHART statement. In addition, you must provide one of the following:

- the variables _LCLNP_, _NP_, and _UCLNP_
- the variable _P_

The following DATA step creates a data set named Climits1, which provides a complete set of control limits for an *np* chart:

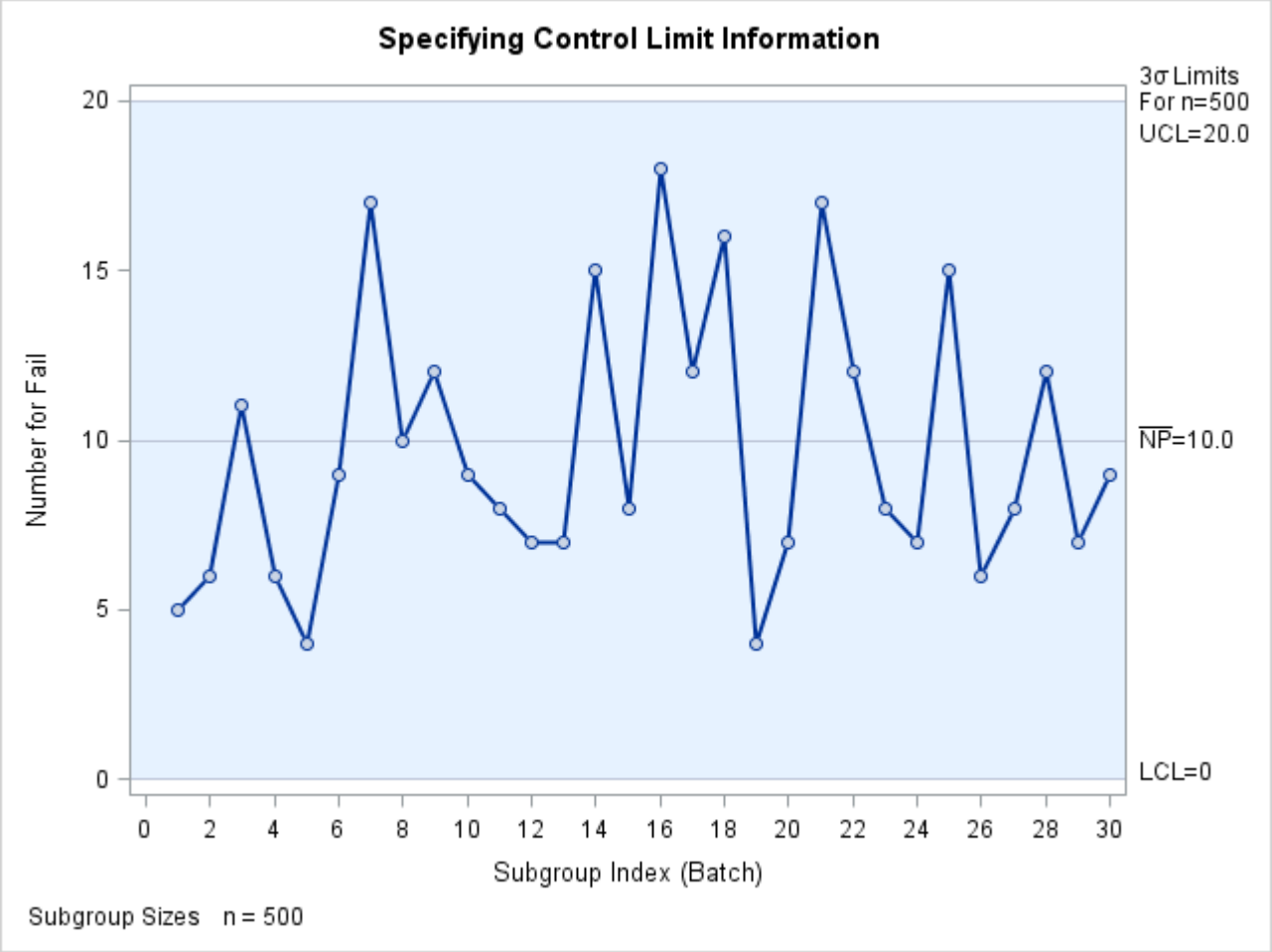
```
data Climits1;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Fail';
   _subgrp_ = 'Batch';
   _limitn_ = 500;
   _type_   = 'STANDARD';
   _lclnp_  =  0;
   _np_     = 10;
   _uclnp_  = 20;
run;
```

The following statements read the control limits from the data set Climits1 and apply them to the count data in the data set Circuits, which is introduced in "Creating np Charts from Count Data" on page 1647:

```
ods graphics on;
title 'Specifying Control Limit Information';
proc shewhart data=Circuits limits=Climits1;
   npchart Fail*Batch / subgroupn = 500
                        odstitle  = title
                        markers;
run;
```

The chart is shown in Output 19.21.1.

Output 19.21.1 Control Limit Information Read from Climits1

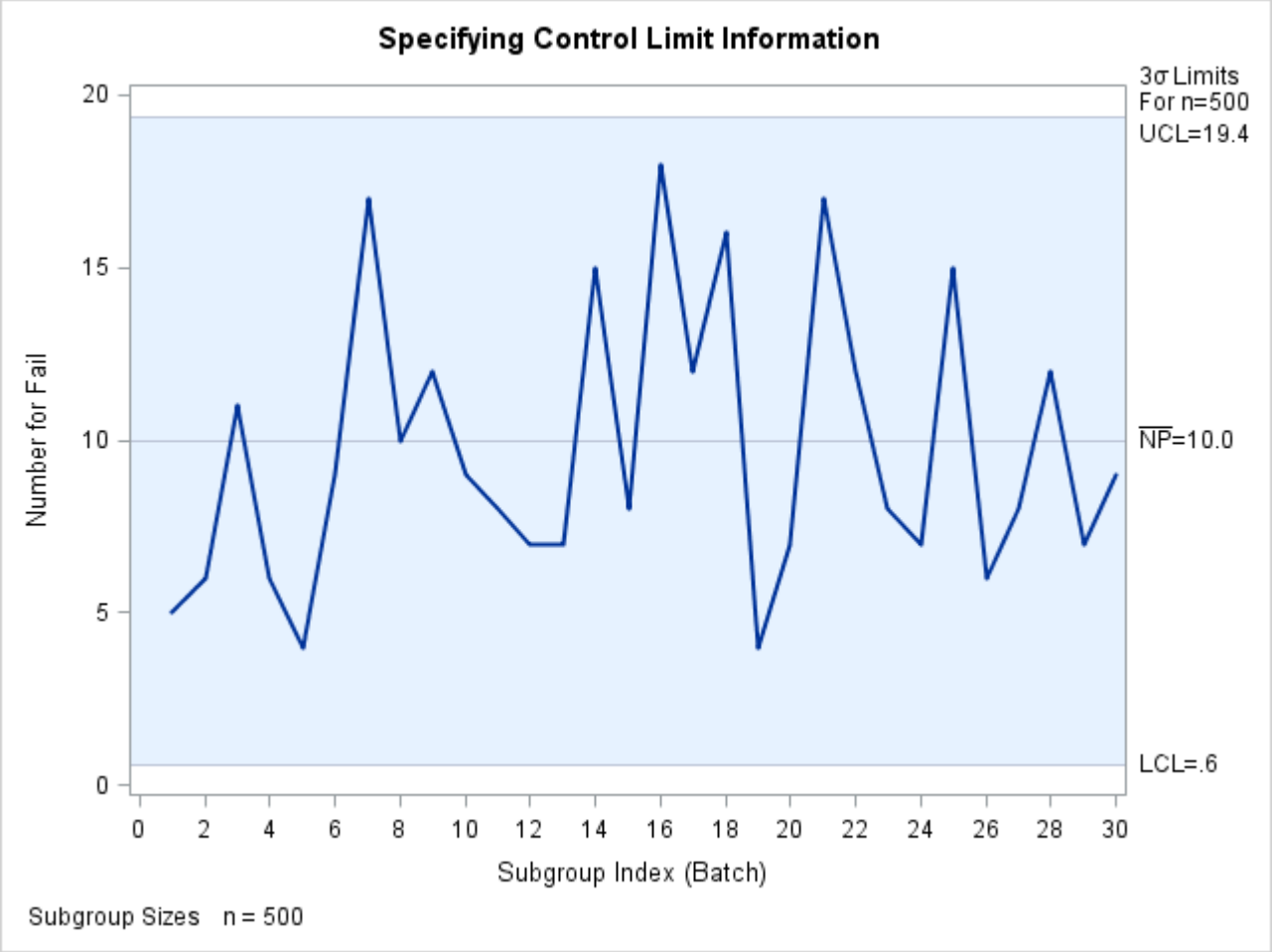


The following DATA step creates a data set named Climits2, which provides a value for the expected proportion of nonconforming items (_P_). This parameter is then used to compute the control limits for the data in Circuits according to the equations in "Control Limits" on page 1668.

```
data Climits2;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Fail';
   _subgrp_ = 'Batch';
   _limitn_ = 500;
   _type_   = 'STANDARD';
   _p_      = .02;
run;

title 'Specifying Control Limit Information';
proc shewhart data=Circuits limits=Climits2;
   npchart Fail*Batch / subgroupn = 500
                        odstitle  = title;
run;
```

The chart is shown in Output 19.21.2. Note that the control limits are not the same as those shown in Output 19.21.1.

Output 19.21.2 Control Limit Information Read from Climits2



PCHART Statement: SHEWHART Procedure

Overview: PCHART Statement

The PCHART statement creates *p* charts for the proportions of nonconforming (defective) items in subgroup samples.

You can use options in the PCHART statement to

- compute control limits from the data based on a multiple of the standard error of the proportions or as probability limits
- tabulate subgroup sample sizes, proportions of nonconforming items, control limits, and other information
- save control limits in an output data set

- save subgroup sample sizes and proportions of nonconforming items in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) proportion of nonconforming items for computing control limits
- specify the data as counts, proportions, or percentages of nonconforming items
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *p* charts with the PCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: PCHART Statement

This section introduces the PCHART statement with simple examples that illustrate commonly used options. Complete syntax for the PCHART statement is presented in the section "Syntax: PCHART Statement" on page 1697, and advanced examples are given in the section "Examples: PCHART Statement" on page 1716.

Creating p Charts from Count Data

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

An electronics company manufactures circuits in batches of 500 and uses a *p* chart to monitor the proportion of failing circuits. Thirty batches are examined, and the failures in each batch are counted. The following statements create a SAS data set named Circuits,[6] which contains the failure counts:

[6] This data set is also used in the "Getting Started" section of "NPCHART Statement: SHEWHART Procedure" on page 1646.

```
data Circuits;
   input Batch Fail @@;
   datalines;
1    5   2   6   3  11   4   6    5   4
6    9   7  17   8  10   9  12   10   9
11   8  12   7  13   7  14  15   15   8
16  18  17  12  18  16  19   4   20   7
21  17  22  12  23   8  24   7   25  15
26   6  27   8  28  12  29   7   30   9
;
```

A partial listing of Circuits is shown in Figure 19.60.

Figure 19.60 The Data Set Circuits

Number of Failing Circuits

Batch	Fail
1	5
2	6
3	11
4	6
5	4

There is a single observation for each batch. The variable Batch identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Fail contains the number of nonconforming items in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *p* chart shown in Figure 19.61:

```
ods graphics off;
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500;
run;
```

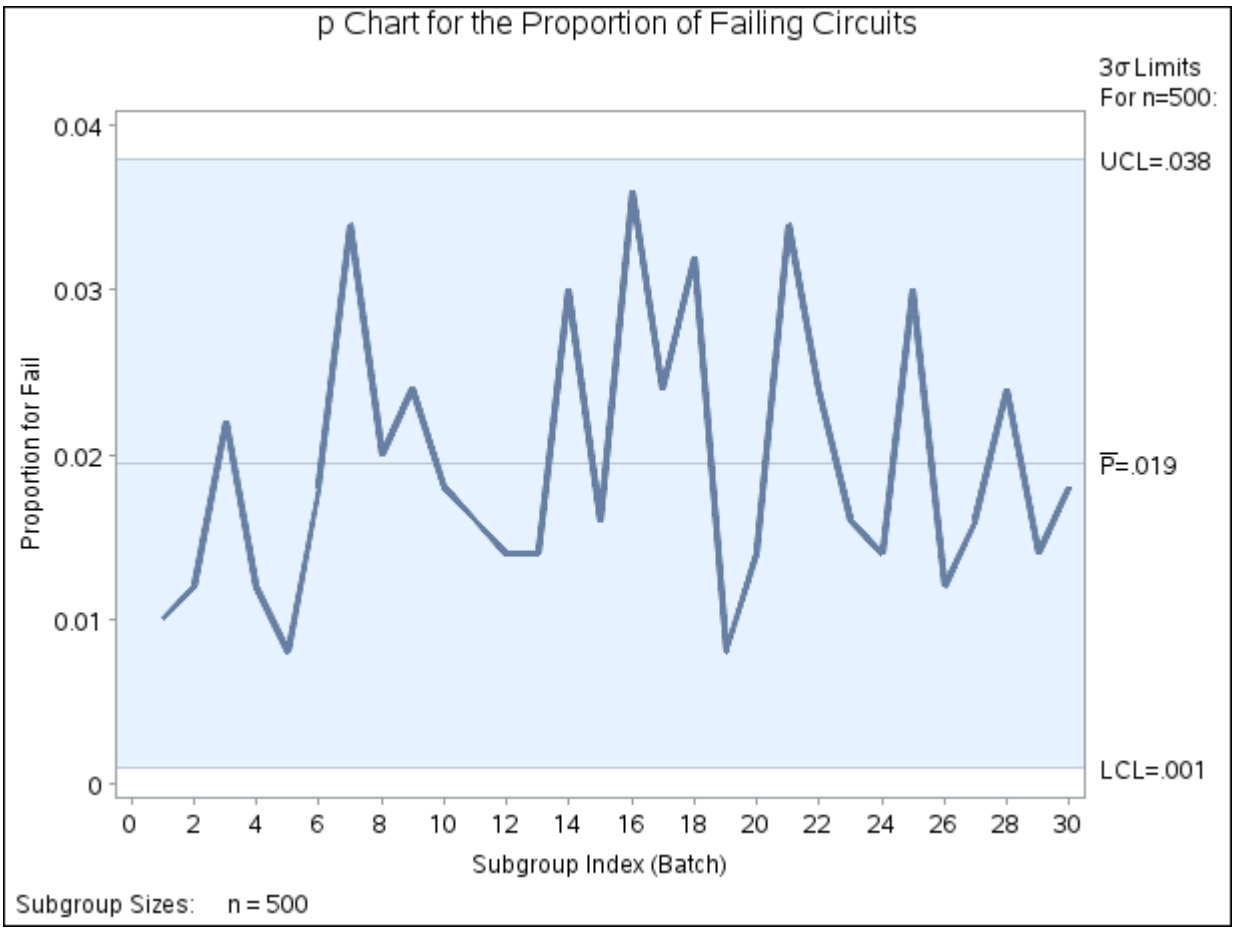
This example illustrates the basic form of the PCHART statement. After the keyword PCHART, you specify the *process* to analyze (in this case, Fail), followed by an asterisk and the *subgroup-variable* (Batch).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. The SUBGROUPN= option specifies the number of items in each subgroup sample and is required with a DATA= input data set. The SUBGROUPN= option specifies one of the following:

- a constant subgroup sample size (as in this case)
- a variable in the input data set whose values provide the subgroup sample sizes (see the next example)

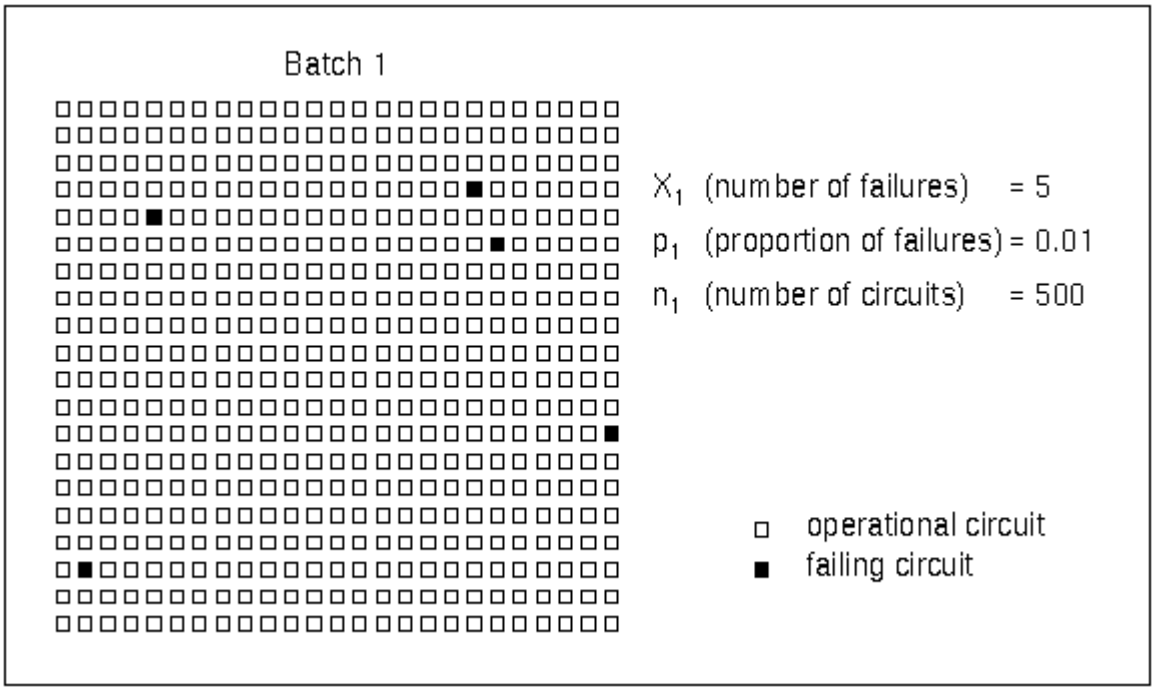
Options such as SUBGROUPN= are specified after the slash (/) in the PCHART statement. A complete list of options is presented in the section "Syntax: PCHART Statement" on page 1697.

Figure 19.61 p Chart for Circuit Failures (Traditional Graphics)



Each point on the p chart represents the proportion of nonconforming items for a particular subgroup. For instance, the value plotted for the first batch is $5/500 = 0.01$, as illustrated in Figure 19.62.

Figure 19.62 Proportions Versus Counts



Because all the points fall within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1710. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1696. For computational details, see "Constructing Charts for Proportion Nonconforming (p Charts)" on page 1708. For more details on reading counts of nonconforming items, see "DATA= Data Set" on page 1713.

Creating p Charts from Summary Data

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create *p* charts using raw data (counts of nonconforming items). However, in many applications, the data are provided in summarized form as proportions or percentages of nonconforming items. This example illustrates how you can use the PCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data Cirprop;
   input Batch pFailed @@;
   Sampsize=500;
   datalines;
 1  0.010   2  0.012   3  0.022   4  0.012   5  0.008
 6  0.018   7  0.034   8  0.020   9  0.024  10  0.018
11  0.016  12  0.014  13  0.014  14  0.030  15  0.016
16  0.036  17  0.024  18  0.032  19  0.008  20  0.014
```

```
21  0.034  22  0.024  23  0.016  24  0.014  25  0.030
26  0.012  27  0.016  28  0.024  29  0.014  30  0.018
;
```

A partial listing of Cirprop is shown in Figure 19.63. The subgroups are still indexed by Batch. The variable pFailed contains the proportions of nonconforming items, and the variable Sampsize contains the subgroup sample sizes.

Figure 19.63 The Data Set Cirprop

Number of Failing Circuits

Batch	Fail
1	5
2	6
3	11
4	6
5	4

The following statements create a *p* chart identical to the one in Figure 19.61:

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=Cirprop;
   pchart pFailed*Batch / subgroupn=Sampsize
                          dataunit =proportion;
label pfailed = 'Proportion for Fail';
run;
```

The DATAUNIT= option specifies that the values of the *process* (pFailed) are proportions of nonconforming items. By default, the values of the *process* are assumed to be counts of nonconforming items (see the previous example).

Alternatively, you can read the data set Cirprop by specifying it as a HISTORY= data set in the PROC SHEWHART statement. A HISTORY= data set used with the PCHART statement must contain the following variables:

- subgroup variable
- subgroup proportion of nonconforming items variable
- subgroup sample size variable

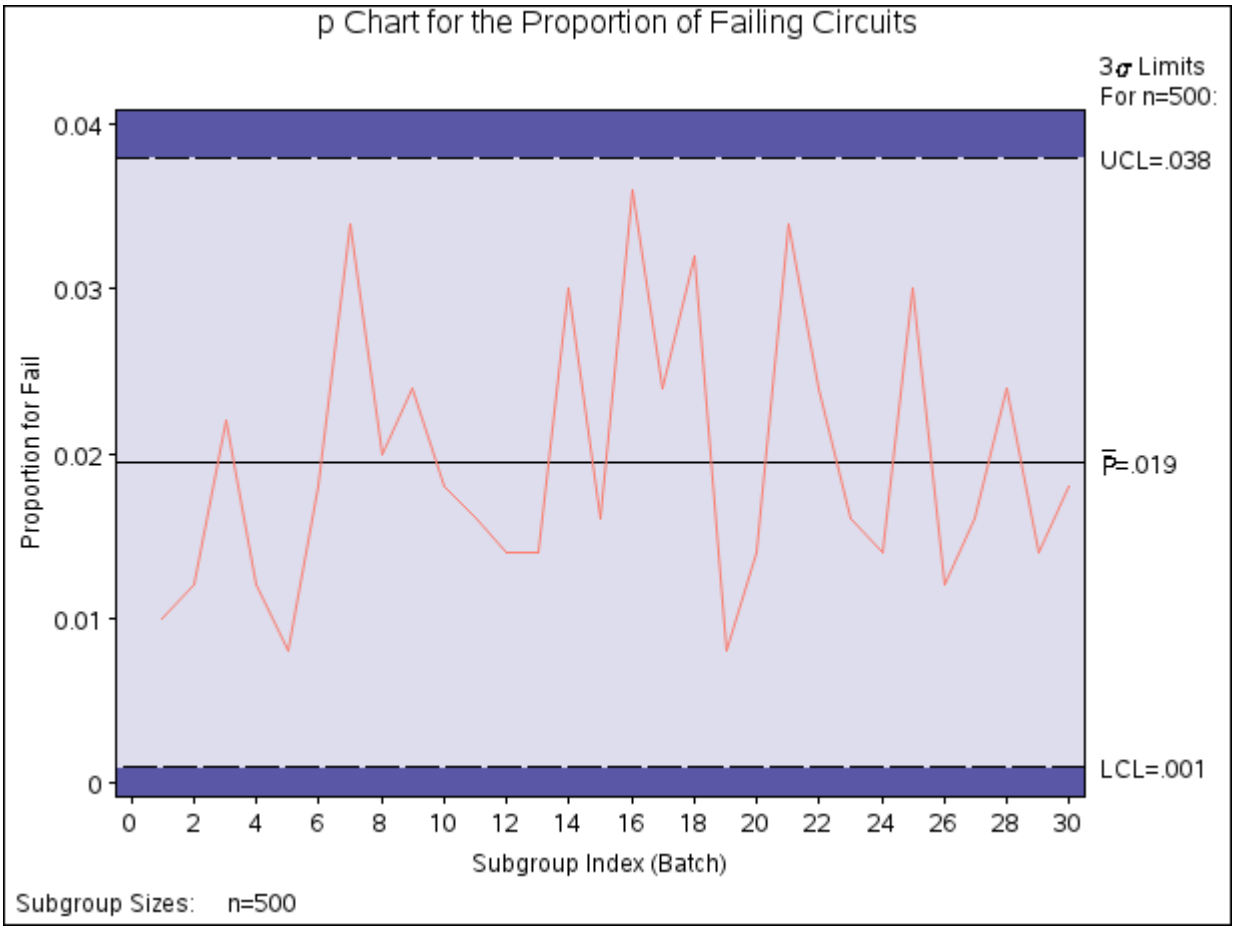
Furthermore, the names of the subgroup proportion and sample size variables must begin with the *process* name specified in the PCHART statement and end with the special suffix characters *P* and *N*, respectively.

To specify Cirprop as a HISTORY= data set and Fail as the *process*, you must rename the variables pFailed and Sampsize to FailP and FailN, respectively. The following statements temporarily rename pFailed and Sampsize for the duration of the procedure step:

```
options nogstyle;
goptions ftext='albany amt';
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart history=Cirprop(rename=(pFailed=FailP
                                      Sampsize=FailN ));
   pchart Fail*Batch / cframe    = lib
                       cinfill   = bwh
                       coutfill  = yellow
                       cconnect  = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the PCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *p* chart is shown in Figure 19.64.

Figure 19.64 *p* Chart from Subgroup Proportions (Traditional Graphics with NOGSTYLE)



In this example, it is more convenient to use Cirprop as a DATA= data set than as a HISTORY= data set. In general, it is more convenient to use the HISTORY= option for input data sets that have been previously created by the SHEWHART procedure as OUTHISTORY= data sets, as illustrated in the next example. For more information, see "HISTORY= Data Set" on page 1715.

Saving Proportions of Nonconforming Items

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

In this example, the PCHART statement is used to create a data set that can later be read by the SHEWHART procedure (as in the preceding example). The following statements read the number of nonconforming items from the data set Circuits (see "Creating p Charts from Count Data" on page 1687) and create a summary data set named Cirhist:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn  = 500
                       outhistory = Cirhist
                       nochart ;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.61. Figure 19.65 contains a partial listing of Cirhist.

Figure 19.65 The Data Set Cirhist

Subgroup Proportions and Control Limit Information

Batch	FailP	FailN
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500

There are three variables in the data set Cirhist.

- Batch contains the subgroup index.
- FailP contains the subgroup proportion of nonconforming items.
- FailN contains the subgroup sample size.

Note that the variables containing the subgroup proportions of nonconforming items and subgroup sample sizes are named by adding the suffix characters *P* and *N* to the *process* Fail specified in the PCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets. For more information, see "OUTHISTORY= Data Set" on page 1711.

Saving Control Limits

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a *p* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1696) or modify the limits with a DATA step program.

The following statements read the number of nonconforming items per subgroup from the data set Circuits (see "Creating p Charts from Count Data" on page 1687) and save the control limits displayed in Figure 19.61 in a data set named Cirlim:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500
                       outlimits = Cirlim
                       nochart ;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Cirlim is listed in Figure 19.66.

Figure 19.66 The Data Set Cirlim Containing Control Limit Information

Control Limits for the Proportion of Failing Circuits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
Fail	Batch	ESTIMATE	500	.002320877	3	.000930786	0.019467	0.038003

The data set Cirlim contains one observation with the limits for *process* Fail. The variables _LCLP_ and _UCLP_ contain the lower and upper control limits, and the variable _P_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or standard value.

For more information, see "OUTLIMITS= Data Set" on page 1711.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500
                       outtable  = Cirtable
                       nochart ;
run;
```

The data set Cirtable is listed in Figure 19.67.

Figure 19.67 The Data Set Cirtable

Subgroup Proportions and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLP_	_SUBP_	_P_	_UCLP_	_EXLIM_
Fail	1	3	500	500	.000930786	0.010	0.019467	0.038003	
Fail	2	3	500	500	.000930786	0.012	0.019467	0.038003	
Fail	3	3	500	500	.000930786	0.022	0.019467	0.038003	
Fail	4	3	500	500	.000930786	0.012	0.019467	0.038003	
Fail	5	3	500	500	.000930786	0.008	0.019467	0.038003	
Fail	6	3	500	500	.000930786	0.018	0.019467	0.038003	
Fail	7	3	500	500	.000930786	0.034	0.019467	0.038003	
Fail	8	3	500	500	.000930786	0.020	0.019467	0.038003	
Fail	9	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	10	3	500	500	.000930786	0.018	0.019467	0.038003	
Fail	11	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	12	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	13	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	14	3	500	500	.000930786	0.030	0.019467	0.038003	
Fail	15	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	16	3	500	500	.000930786	0.036	0.019467	0.038003	
Fail	17	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	18	3	500	500	.000930786	0.032	0.019467	0.038003	
Fail	19	3	500	500	.000930786	0.008	0.019467	0.038003	
Fail	20	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	21	3	500	500	.000930786	0.034	0.019467	0.038003	
Fail	22	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	23	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	24	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	25	3	500	500	.000930786	0.030	0.019467	0.038003	
Fail	26	3	500	500	.000930786	0.012	0.019467	0.038003	
Fail	27	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	28	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	29	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	30	3	500	500	.000930786	0.018	0.019467	0.038003	

This data set contains one observation for each subgroup sample. The variables _SUBP_ and _SUBN_ contain the subgroup proportions of nonconforming items and subgroup sample sizes. The variables _LCLP_ and _UCLP_ contain the lower and upper control limits, and the variable _P_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1712.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read the information in Cirtable and display a *p* chart (not shown here) identical to the chart in Figure 19.61:

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart table=Cirtable;
   pchart Fail*Batch;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1715.

Reading Preestablished Control Limits

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Cirlim saved control limits computed from the data in Circuits. This example shows how these limits can be applied to new data provided in the following data set:

```
data Circuit2;
   input Batch Fail @@;
   datalines;
31  12   32   9   33  16   34   9
35   3   36   8   37  20   38   4
39   8   40   6   41  12   42  16
43   9   44   2   45  10   46   8
47  14   48  10   49  11   50   9
;
```

The following statements create a *p* chart for the data in Circuit2 using the control limits in Cirlim:

```
ods graphics on;
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=Circuit2 limits=Cirlim;
   pchart Fail*Batch / subgroupn = 500
                       odstitle  = title;
run;
```

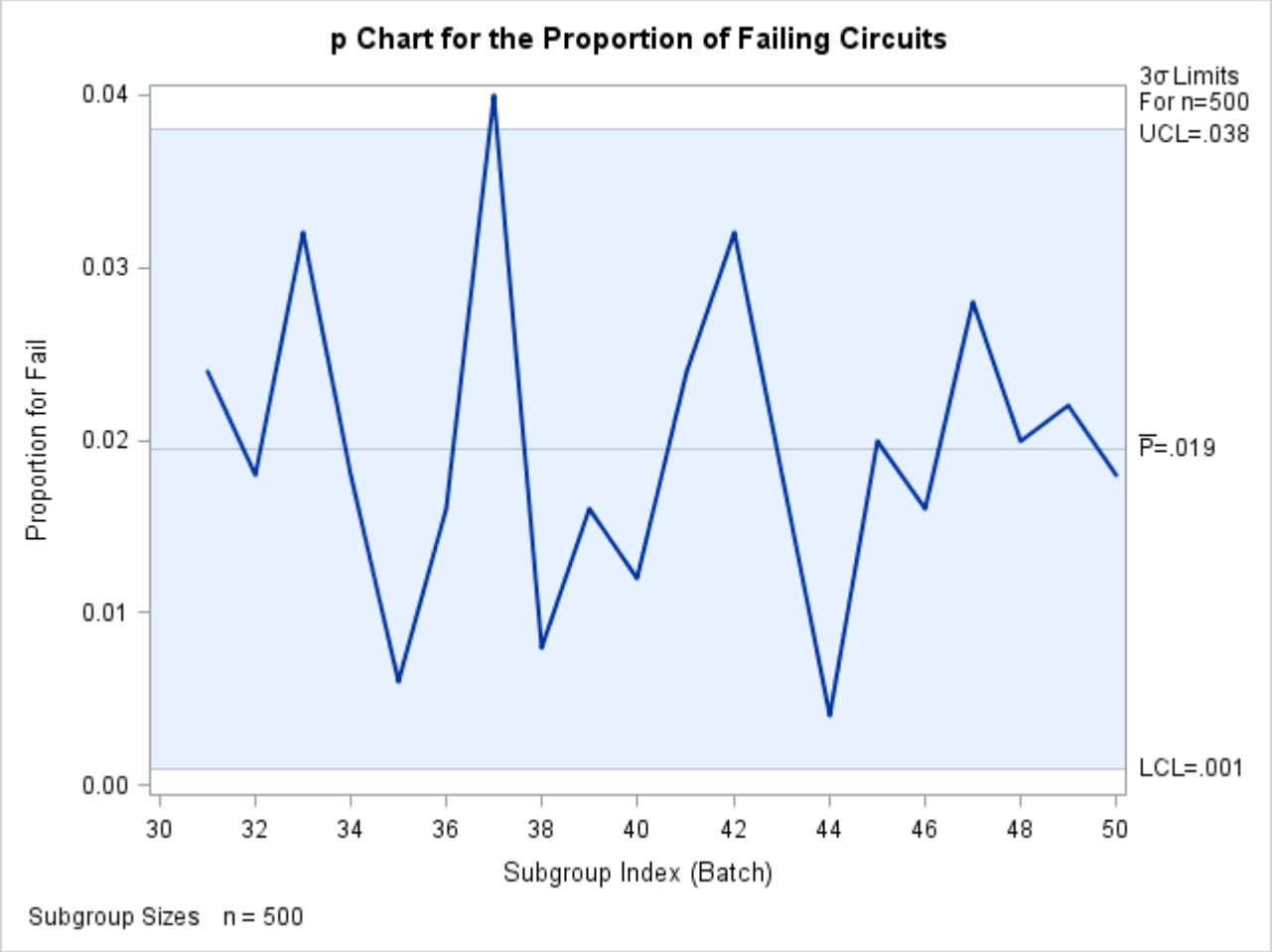
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *p* chart is created by using ODS Graphics instead of traditional graphics.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Fail
- the value of _SUBGRP_ matches the *subgroup-variable* name Batch

The resulting *p* chart is shown in Figure 19.68.

Figure 19.68 p Chart for Second Set of Circuit Failures (ODS Graphics)



The proportion of nonconforming items in the 37th batch exceeds the upper control limit, signaling that the process is out of control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1714 for details concerning the variables that you must provide.

Syntax: PCHART Statement

The basic syntax for the PCHART statement is as follows:

PCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

PCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of PCHART statements in the SHEWHART procedure. The components of the PCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconforming items are read from a DATA= data set, *process* must be the name of the variable containing the numbers. For an example, see "Creating p Charts from Summary Data" on page 1690.
- If proportions of nonconforming items are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating p Charts from Summary Data" on page 1690.
- If proportions of nonconforming items and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1693.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *p* charts for Rejects and Reworks:

```
proc shewhart data=Measures;
   pchart (Rejects Reworks)*Sample / subgroupn=100;
run;
```

Note that when data are read from a DATA= data set, the SUBGROUPN= option, which specifies subgroup sample sizes, is required.

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding PCHART statement, Sample is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot proportions of nonconforming items.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a *p* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   pchart Rejects*Day='*' / subgroupn=100;
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the PCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.42 PCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	Requests probability limits at discrete values
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	Displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit

Table 19.42 *continued*

Option	Description
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMIT0	Suppresses display of lower control limit if it is 0
NOLIMIT1	Suppresses display of upper control limit if it is 1 (100%)
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
PSYMBOL=	Specifies label for central line
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
Standard Value Options	
P0=	Specifies known (standard) value p_0 for proportion of nonconforming items p
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on p chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits

Table 19.42 *continued*

Option	Description
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive

Table 19.42 *continued*

Option	Description
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis
WAXIS=	Specifies width of axis lines
YSCALE=	scales vertical axis in percent units (rather than proportions)

Table 19.42 *continued*

Option	Description
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays p chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= options
CVREF=	Specifies color for lines requested by VREF= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option

Table 19.42 *continued*

Option	Description
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features chart
DESCRIPTION=	Specifies description of *p* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of *p* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors

Table 19.42 *continued*

Option	Description
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
DATAUNIT	Specifies that input values are proportions or percentages (rather than counts) of nonconforming items
MISSBREAK	Specifies that observations with missing values are not to be processed

Table 19.42 *continued*

Option	Description
SUBGROUPN	Specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set

Table 19.42 *continued*

Option	Description
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend

Table 19.42 *continued*

Option	Description
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: PCHART Statement

The following sections provide details that are specific to the PCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Proportion Nonconforming (p Charts)

The following notation is used in this section:

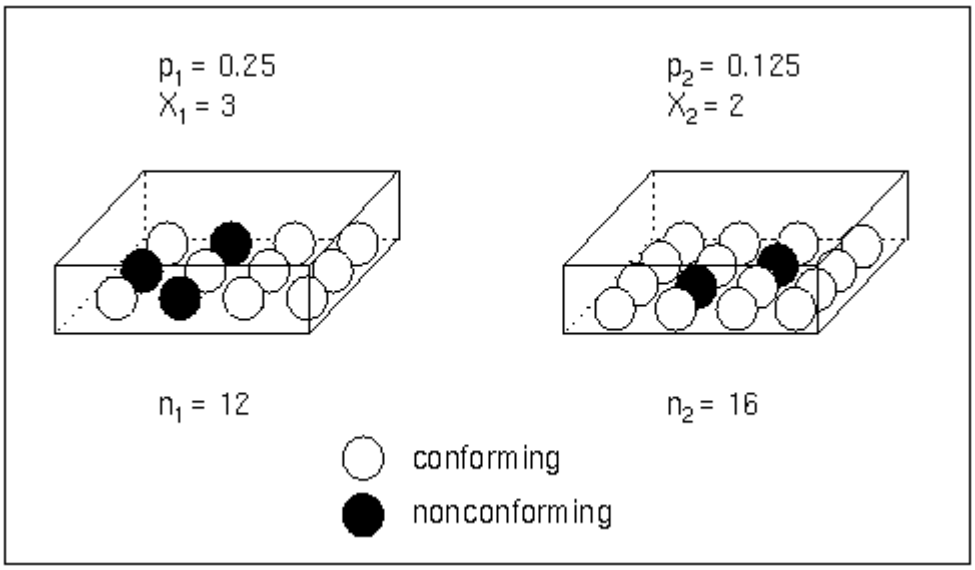
p	Expected proportion of nonconforming items produced by the process
p_i	Proportion of nonconforming items in the ith subgroup
X_i	Number of nonconforming items in the ith subgroup
n_i	Number of items in the ith subgroup

$\bar{p}$	Average proportion of nonconforming items taken across subgroups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_N p_N}{n_1 + \cdots + n_N} = \frac{X_1 + \cdots + X_N}{n_1 + \cdots + n_N}$$
N	Number of subgroups
$I_T(\alpha, \beta)$	Incomplete beta function: $$I_T(\alpha, \beta) = (\Gamma(\alpha + \beta)/\Gamma(\alpha)\Gamma(\beta)) \int_0^T t^{\alpha-1}(1-t)^{\beta-1} dt$$ for $0 < T < 1$, $\alpha > 0$, and $\beta > 0$, where $\Gamma(\cdot)$ is the gamma function

Plotted Points

Each point on a p chart represents the observed proportion ($p_i = X_i/n_i$) of nonconforming items in a subgroup. For example, suppose the second subgroup (see Figure 19.69) contains 16 items, of which two are nonconforming. The point plotted for the second subgroup is $p_2 = 2/16 = 0.125$.

Figure 19.69 Proportions Versus Counts



Note that an np chart displays the number (count) of nonconforming items X_i. You can use the NPCHART statement to create np charts; see "NPCHART Statement: SHEWHART Procedure" on page 1646.

Central Line

By default, the central line on a p chart indicates an estimate of p that is computed as $\bar{p}$. If you specify a known value (p_0) for p, the central line indicates the value of p_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of p_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that p_i exceeds the limits

The lower and upper control limits, LCL and UCL, respectively, are computed as

$$\begin{aligned} \text{LCL} &= \max\left(\bar{p} - k\sqrt{\bar{p}(1-\bar{p})/n_i}\ , 0\right) \\ \text{UCL} &= \min\left(\bar{p} + k\sqrt{\bar{p}(1-\bar{p})/n_i}\ , 1\right) \end{aligned}$$

A lower probability limit for p_i can be determined using the fact that

$$\begin{aligned} P\{p_i < \text{LCL}\} &= 1 - P\{p_i \geq \text{LCL}\} \\ &= 1 - P\{X_i \geq n_i\text{LCL}\} \\ &= 1 - I_{\bar{p}}(n_i\text{LCL}, n_i + 1 - n_i\text{LCL}) \\ &= I_{1-\bar{p}}(n_i + 1 - n_i\text{LCL}, n_i\text{LCL}) \end{aligned}$$

Refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that X_i is binomially distributed. The lower probability limit LCL is then calculated by setting

$$I_{1-\bar{p}}(n_i + 1 - n_i\text{LCL}, n_i\text{LCL}) = \alpha/2$$

and solving for LCL. Similarly, the upper probability limit for p_i can be determined using the fact that

$$\begin{aligned} P\{p_i > \text{UCL}\} &= P\{p_i > \text{UCL}\} \\ &= P\{X_i > n_i\text{UCL}\} \\ &= I_{\bar{p}}(n_i\text{UCL} + 1, n_i - n_i\text{UCL}) \end{aligned}$$

The upper probability limit UCL is then calculated by setting

$$I_{\bar{p}}(n_i\text{UCL} + 1, n_i - n_i\text{UCL}) = \alpha/2$$

and solving for UCL. The probability limits are asymmetric around the central line. Note that both the control limits and probability limits vary with n_i.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify p_0 with the P0= option or with the variable _P_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.44 lists the variables that can be saved.

Table 19.44 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLP	Lower control limit for proportion of nonconforming items
LIMITN	Nominal sample size associated with the control limits
_P_	Average proportion of nonconforming items ($\bar{p}$ or p_0)
SIGMAS	Multiple (k) of standard error of p_i
SUBGRP	*Subgroup-variable* specified in the PCHART statement
TYPE	Type (standard or estimate) of _P_
UCLP	Upper control limit for proportion of nonconforming items
VAR	*Process* specified in the PCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLP_, _UCLP_, and _SIGMAS_.
2. If the limits are defined in terms of a multiple k of the standard error of p_i, the value of _ALPHA_ is computed as $\alpha = P\{p_i < \text{\_LCLP\_}\} + P\{p_i > \text{\_UCLP\_}\}$, using the incomplete beta function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = (\text{\_UCLP\_} - \text{\_P\_})/\sqrt{\text{\_P\_}(1 - \text{\_P\_})/\text{\_LIMITN\_}}$. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the PCHART statement. For an example, see "Saving Control Limits" on page 1693.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable named by *process* suffixed with *P*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the PCHART statement. For example, consider the following statements:

```
proc shewhart data=Input;
   pchart (Rework Rejected)*Batch / outhistory=Summary
                                    subgroupn =30;
run;
```

The data set Summary contains variables named Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Proportions of Nonconforming Items" on page 1693.

Note that an OUTHISTORY= data set created with the PCHART statement can be reused as a HISTORY= data set by either the PCHART statement or the NPCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.45 lists the variables that are saved.

Table 19.45 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on p chart
LCLP	Lower control limit for proportion of nonconforming items
LIMITN	Nominal sample size associated with the control limits
_P_	Average proportion of nonconforming items
SIGMAS	Multiple (k) of the standard error of p_i associated with the control limits
Subgroup	Values of the subgroup variable
SUBP	Subgroup proportion of nonconforming items
SUBN	Subgroup sample size
TESTS	Tests for special causes signaled on p chart
UCLP	Upper control limit for proportion of nonconforming items
VAR	*Process* specified in the PCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request the first four tests (the tests appropriate for *p* charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1693.

Input Data Sets

DATA= Data Set

You can read raw data (counts of nonconforming items) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the PCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the PCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *process* and a value for the *subgroup-variable*. The data set must contain one observation for each subgroup. Note that you can specify the DATAUNIT= option in the PCHART statement to read proportions or percentages of nonconforming items instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables

- ID variables

When you use a DATA= data set with the PCHART statement, the SUBGROUPN= option (which specifies the subgroup sample size) is required. By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating p Charts from Count Data" on page 1687.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   pchart Rejects*Batch / subgroupn= 100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLP_, _P_, and _UCLP_, which specify the control limits directly
- the variable _P_, without providing _LCLP_ and _UCLP_. The value of _P_ is used to calculate the control limits according to the equations in "Control Limits" on page 1710.

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1696.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set.

A HISTORY= data set used with the PCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable for each *process*
- a subgroup sample size variable for each *process*

The names of the proportion sample size variables must be the *process* name concatenated with the special suffix characters *P* and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   pchart (Rework Rejected)*Batch / subgroupn=50;
run;
```

The data set Summary must include the variables Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating p Charts from Summary Data" on page 1690.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.46 lists the variables required in a TABLE= data set used with the PCHART statement.

Table 19.46 Variables Required in a TABLE= Data Set

Variable	Description
LCLP	Lower control limit for proportion of nonconforming items
LIMITN	Nominal sample size associated with the control limits
_P_	Average proportion of nonconforming items
Subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBP	Subgroup proportion of nonconforming items
UCLP	Upper control limit for proportion of nonconforming items

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1693.

Examples: PCHART Statement

This section provides advanced examples of the PCHART statement.

Example 19.22: Applying Tests for Special Causes

NOTE: See *p Charts-Tests for Special Causes* in the SAS/QC Sample Library.

This example shows how you can apply tests for special causes to make *p* charts more sensitive to special causes of variation. The following statements create a SAS data set named Circuit3, which contains the number of failing circuits for 20 batches from the circuit manufacturing process introduced in "Creating p Charts from Count Data" on page 1687:

```
data Circuit3;
   input Batch Fail @@;
   datalines;
 1 12    2 21    3 16    4  9
 5  3    6  4    7  6    8  9
 9 11   10 13   11 12   12  7
13  2   14 14   15  9   16  8
17 14   18 10   19 11   20  9
;
```

The following statements create the *p* chart, apply several tests to the chart, and tabulate the results:

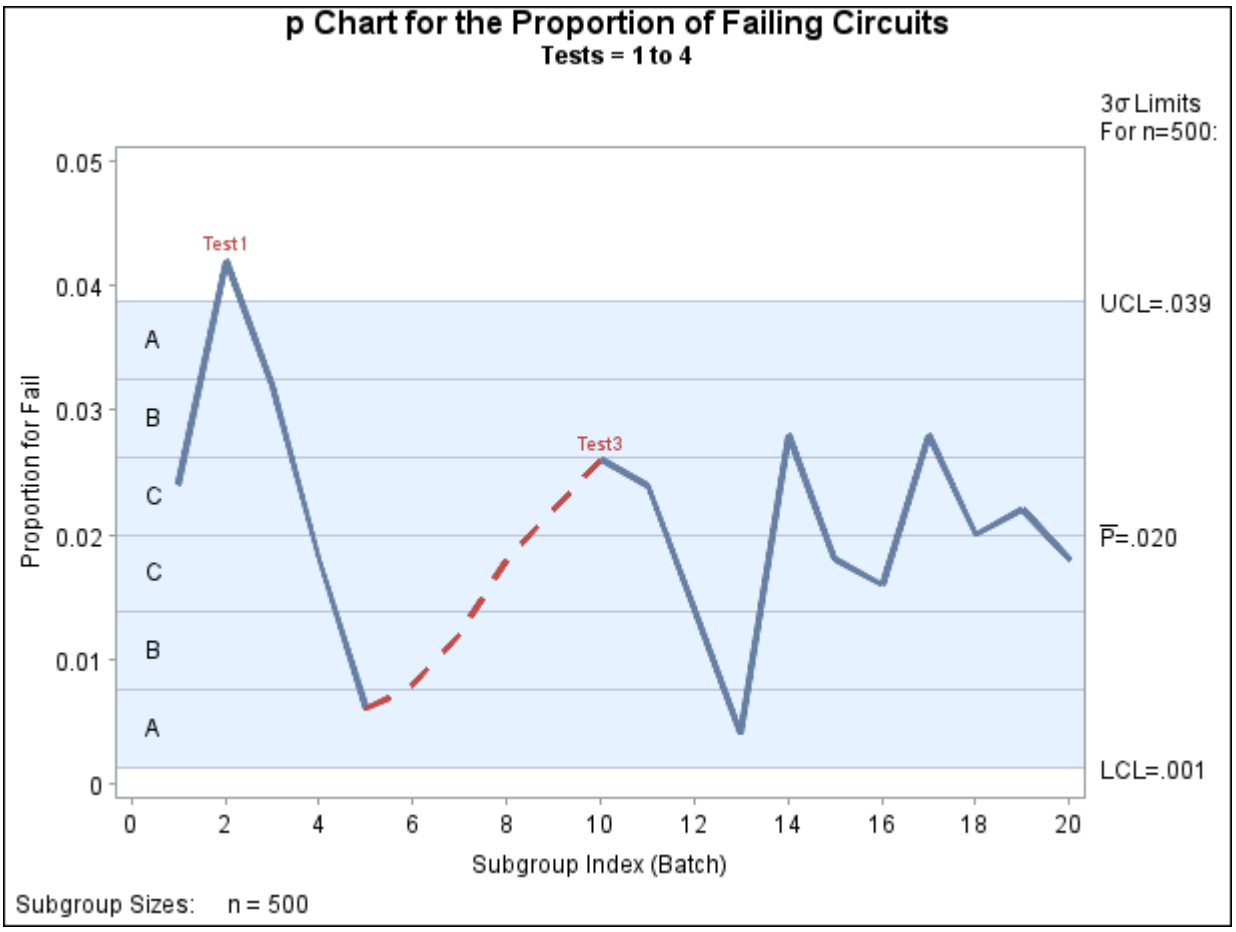
```
ods graphics off;
title1'p Chart for the Proportion of Failing Circuits';
title2 'Tests = 1 to 4';
proc shewhart data=Circuit3;
   pchart Fail*Batch / subgroupn = 500
                       tests     = 1 to 4
                       zones
                       zonelabels
                       ltests    = 20
                       table
                       tabletest
                       tablelegend;
run;
```

The chart is shown in Output 19.22.1, and the printed output is shown in Output 19.22.2. The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. The TABLETESTS option requests a table of proportions of nonconforming items and control limits, with a column indicating which subgroups tested positive for special causes. The TABLELEGEND option adds a legend describing the tests that are positive.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled.

Output 19.22.1 and Output 19.22.2 indicate that Test 1 is positive at batch 2 and Test 3 is positive at batch 10.

Output 19.22.1 Tests for Special Causes Displayed on p Chart



Output 19.22.2 Tabular Form of p Chart

p Chart for the Proportion of Failing Circuits
Tests = 1 to 4

The SHEWHART Procedure

p Chart Summary for Fail

3 Sigma Limits with n=500 for Proportion

Batch	Subgroup Sample Size	Lower Limit	Subgroup Proportion	Upper Limit	Special Tests Signaled
1	500	0.00121703	0.02400000	0.03878297	
2	500	0.00121703	0.04200000	0.03878297	1
3	500	0.00121703	0.03200000	0.03878297	
4	500	0.00121703	0.01800000	0.03878297	
5	500	0.00121703	0.00600000	0.03878297	
6	500	0.00121703	0.00800000	0.03878297	
7	500	0.00121703	0.01200000	0.03878297	
8	500	0.00121703	0.01800000	0.03878297	
9	500	0.00121703	0.02200000	0.03878297	
10	500	0.00121703	0.02600000	0.03878297	3
11	500	0.00121703	0.02400000	0.03878297	
12	500	0.00121703	0.01400000	0.03878297	
13	500	0.00121703	0.00400000	0.03878297	
14	500	0.00121703	0.02800000	0.03878297	
15	500	0.00121703	0.01800000	0.03878297	
16	500	0.00121703	0.01600000	0.03878297	
17	500	0.00121703	0.02800000	0.03878297	
18	500	0.00121703	0.02000000	0.03878297	
19	500	0.00121703	0.02200000	0.03878297	
20	500	0.00121703	0.01800000	0.03878297	

Test Descriptions	
Test 1	One point beyond Zone A (outside control limits)
Test 3	Six points in a row steadily increasing or decreasing

Example 19.23: Specifying Standard Average Proportion

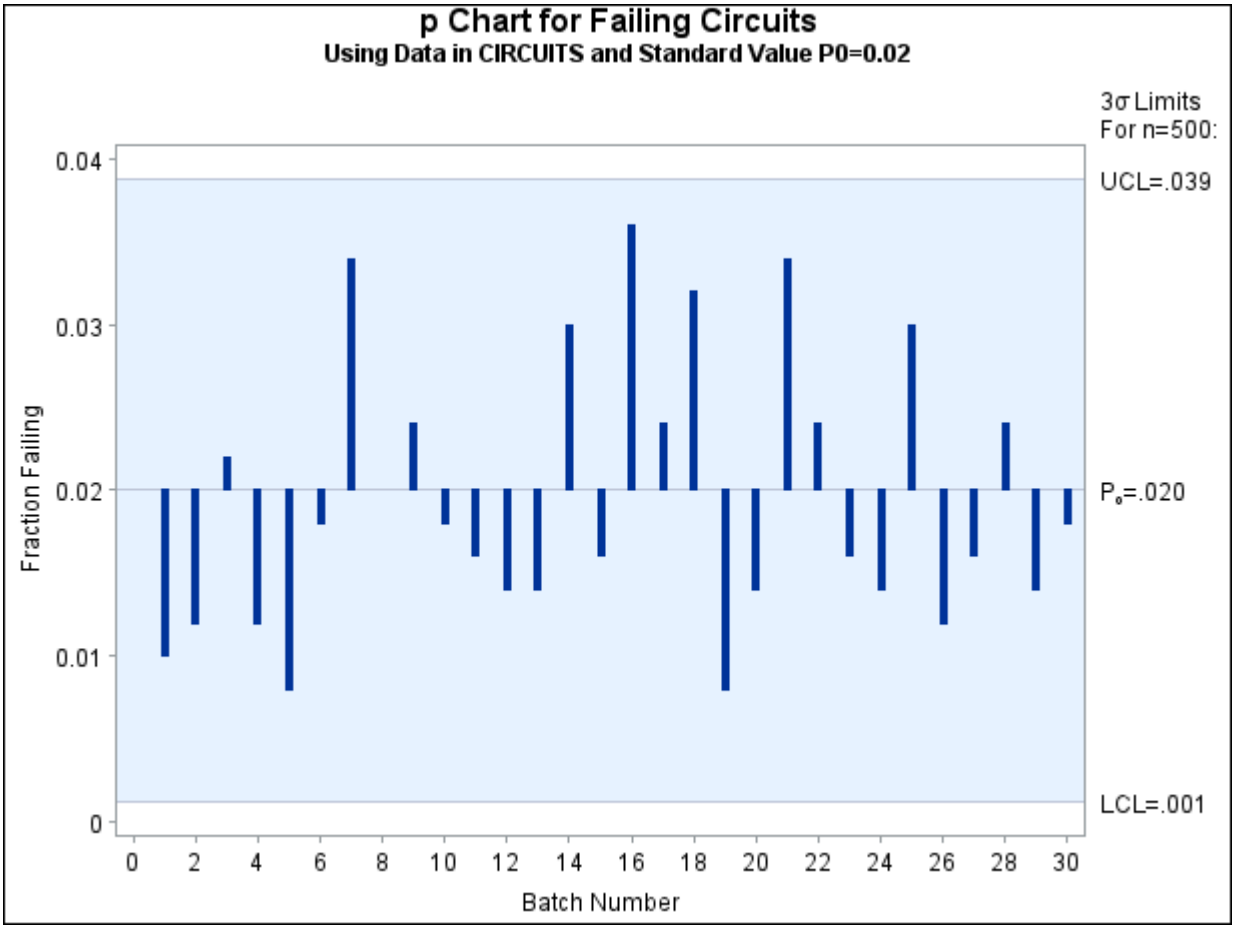
NOTE: See *p Charts-Specifying Std Average Proportion* in the SAS/QC Sample Library.

In some situations, a standard (known) value (p_0) is available for the expected proportion of nonconforming items, based on extensive testing or previous sampling. This example illustrates how you can specify p_0 to create a p chart.

A p chart is used to monitor the proportion of failing circuits in the data set Circuits, which is introduced in "Creating p Charts from Count Data" on page 1687. The expected proportion is known to be $p_0 = 0.02$. The following statements create a p chart, shown in Output 19.23.1, using p_0 to compute the control limits:

```
ods graphics off;
title1 'p Chart for Failing Circuits';
title2 'Using Data in CIRCUITS and Standard Value P0=0.02';
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500
                       p0        = 0.02
                       psymbol   = p0
                       wneedles  = 3
                       nolegend;
   label Batch = 'Batch Number'
         Fail  = 'Fraction Failing';
run;
```

Output 19.23.1 A p Chart with Standard Value p_0



The chart indicates that the process is in control. The P0= option specifies p_0. The PSYMBOL= option specifies a label for the central line indicating that the line represents a standard value. The NEEDLES option connects points to the central line with vertical needles. The NOLEGEND option suppresses the default

legend for subgroup sample sizes. Labels for the vertical and horizontal axes are provided with the LABEL statement. For details concerning axis labeling, see "Axis Labels" on page 1973.

Alternatively, you can specify p_0 using the variable _P_ in a LIMITS= data set, as follows:

```
data Climits;
   length _var_ _subgrp_ _type_ $8;
   _p_      = 0.02;
   _subgrp_ = 'Batch';
   _var_    = 'Fail';
   _type_   = 'STANDARD';
   _limitn_ = 500;
run;

proc shewhart data=Circuits limits=Climits;
   pchart Fail*Batch / subgroupn = 500
                       psymbol   = p0
                       nolegend
                       needles;
   label batch ='Batch Number'
         fail  ='Fraction Failing';
run;
```

The bookkeeping variable _TYPE_ indicates that _P_ has a standard value. The chart produced by these statements is identical to the chart in Output 19.23.1.

Example 19.24: Working with Unequal Subgroup Sample Sizes

NOTE: See *p Charts with Unequal Subgroup Sample Sizes* in the SAS/QC Sample Library.

The following statements create a SAS data set named Battery, which contains the number of alkaline batteries per lot failing an acceptance test. The number of batteries tested in each lot varies but is approximately 150.

```
data Battery;
   length lot $3;
   input lot nFailed Sampsize @@;
   label nFailed  = 'Number Failed'
         lot      = 'Lot Number'
         Sampsize = 'Number Sampled';
   datalines;
AE3  6  151     AE4  5  142     AE9  6  145
BR3  9  149     BR7  3  150     BR8  0  156
BR9  4  150     DB1  9  158     DB2  4  152
DB3  0  162     DB5  9  140     DB6  7  161
DS4  6  154     DS6  1  144     DS8  5  154
JG1  3  151     MC3  8  148     MC4  2  143
MK6  4  150     MM1  4  147     MM2  0  150
RT5  2  154     RT9  8  149     SP1  3  160
SP3  9  153
;
```

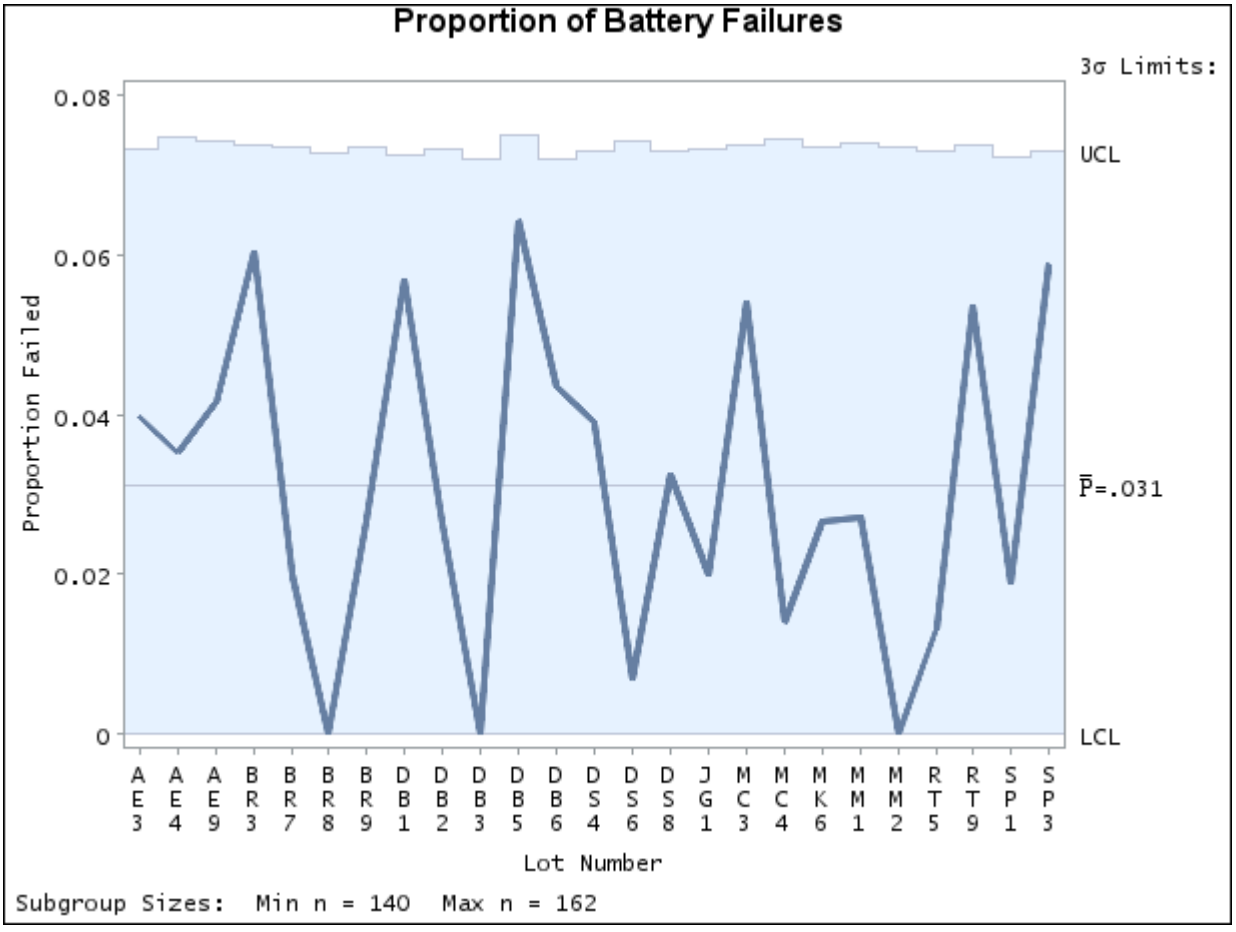
The variable nFailed contains the number of battery failures, the variable Lot contains the lot number, and the variable Sampsize contains the lot sample size. The following statements request a *p* chart for this data:

```
ods graphics off;
title 'Proportion of Battery Failures';
proc shewhart data=Battery;
   pchart nFailed*Lot / subgroupn = Sampsize
                        turnhlabels
                        font      = 'Lucida Console'
                        outlimits = Batlim;
   label nFailed = 'Proportion Failed';
run;
```

Here the FONT= option is used to specify the name of a hardware font to be used for the *p* chart. In this case the requested font is Lucida Console, a Windows TrueType font. See *SAS/GRAPH: Reference* and *SAS Companion for Microsoft Windows* for more information about hardware and TrueType fonts.

The chart is shown in Output 19.24.1 and the OUTLIMITS= data set Batlim is listed in Output 19.24.2.

Output 19.24.1 A *p* Chart with Varying Subgroup Sample Sizes



Note that the upper control limit varies with the subgroup sample size. The lower control limit is truncated at zero. The sample size legend indicates the minimum and maximum subgroup sample sizes.

Output 19.24.2 Listing of the Control Limits Data Set Batlim

Control Limits for Battery Failures

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
nFailed	lot	ESTIMATE	V	V	3	V	0.031010	V

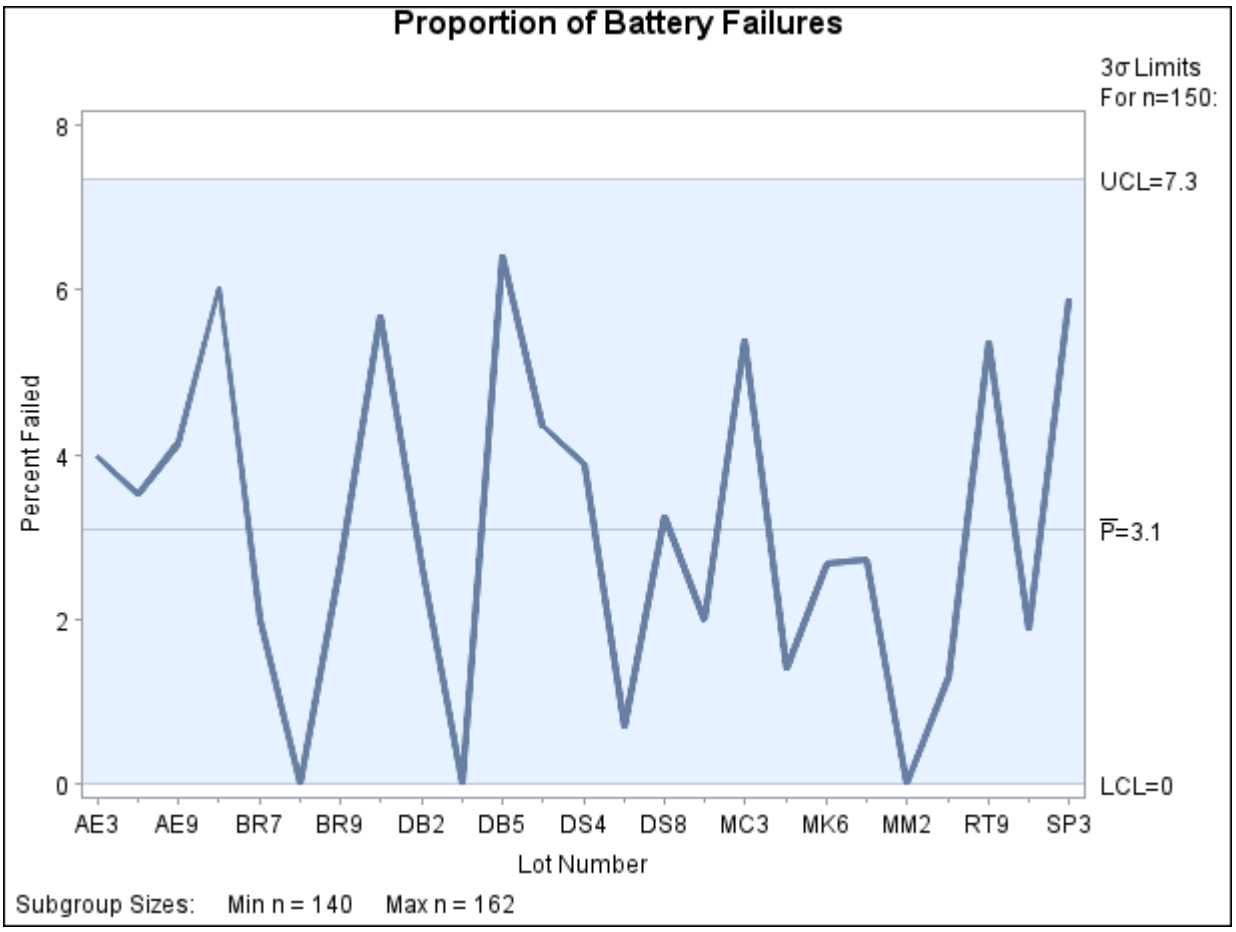
The variables in Batlim whose values vary with subgroup sample size are assigned the special missing value *V*.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing the control limits, as illustrated by the following statements:

```
title 'Proportion of Battery Failures';
proc shewhart data=Battery;
   pchart nFailed*Lot / subgroupn = Sampsize
                        limitn    = 150
                        alln
                        outlimits = Clim2
                        yscale    = percent;
   label nFailed = 'Percent Failed';
run;
```

The ALLN option specifies that all points (regardless of subgroup sample size) are to be displayed. By default, only points for subgroups whose sample size matches the LIMITN= value are displayed. The YSCALE= option specifies that the vertical axis is to be scaled in percentages rather than proportions. The chart is shown in Output 19.24.3.

Output 19.24.3 Control Limits Based on Fixed Subgroup Sample Size



All the points are inside the control limits, indicating that the process is in statistical control. Because there is relatively little variation in the sample sizes, the control limits in Output 19.24.3 provide a close approximation to the exact control limits in Output 19.24.1, and the same conclusions can be drawn from both charts. In general, care should be taken when interpreting charts that use a nominal sample size to compute control limits, because these limits are only approximate when the sample sizes vary.

Example 19.25: Creating a Chart with Revised Control Limits

NOTE: See *p Charts with Revised Control Limits* in the SAS/QC Sample Library.

The following statements create a SAS data set named CircOne, which contains the number of failing circuits for 30 batches produced by the circuit manufacturing process introduced in the section "Getting Started: PCHART Statement" on page 1687:

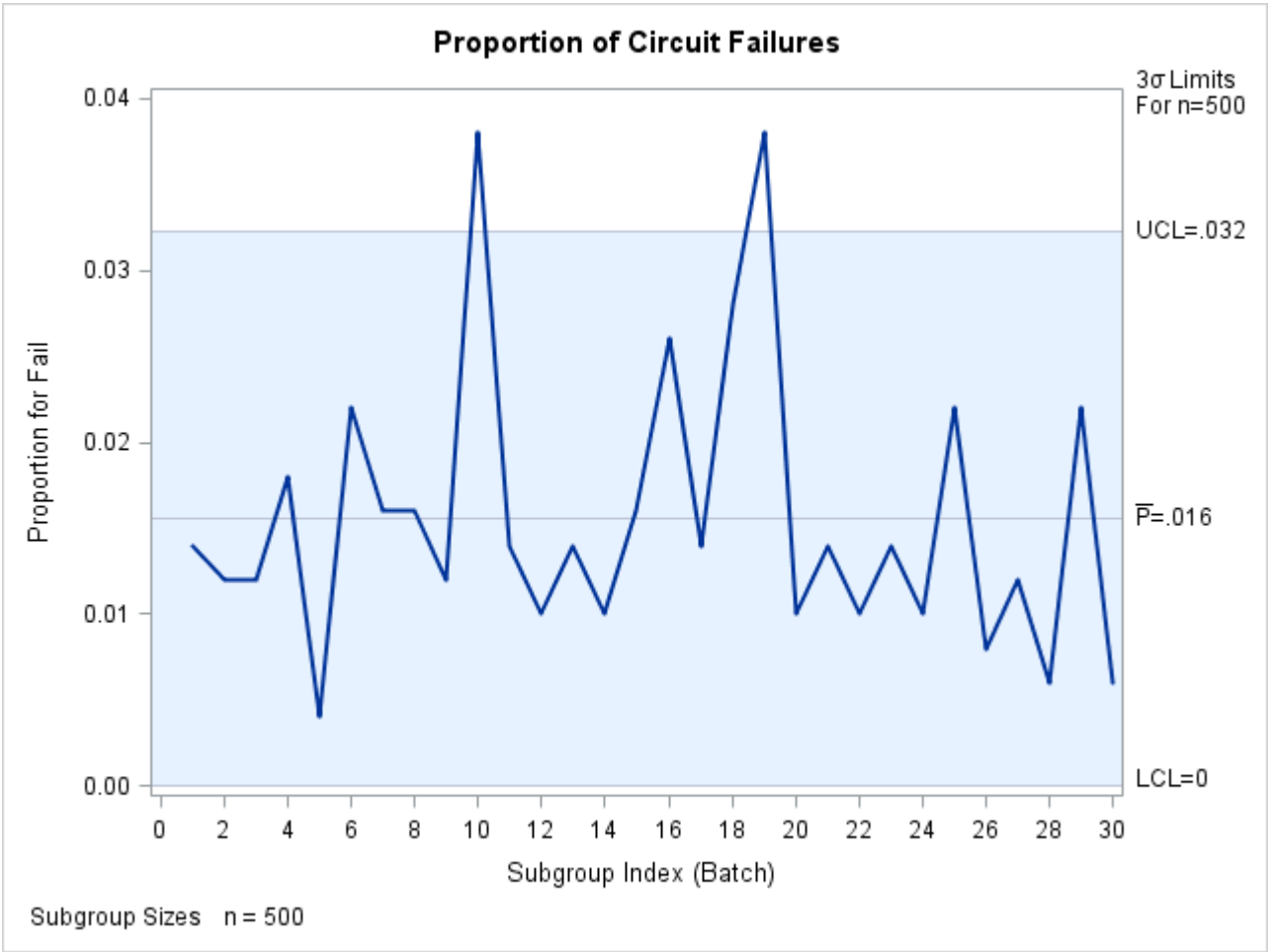
```
data CircOne;
   input Batch Fail @@;
   datalines;
 1   7   2   6   3   6   4   9    5   2
 6  11   7   8   8   8   9   6   10  19
```

```
11   7  12   5  13   7  14   5   15   8
16  13  17   7  18  14  19  19   20   5
21   7  22   5  23   7  24   5   25  11
26   4  27   6  28   3  29  11   30   3
;
```

A p chart is used to monitor the proportion of failing circuits. The following statements create the chart shown in Output 19.25.1:

```
ods graphics on;
title 'Proportion of Circuit Failures';
proc shewhart data=CircOne;
   pchart Fail*Batch / subgroupn = 500
                       outindex  = 'Trial Limits'
                       outlimits = Faillim1
                       odstitle  = title;
run;
```

Output 19.25.1 A p Chart for Circuit Failures



Batches 10 and 19 have unusually high proportions of failing circuits. Subsequent investigation identifies special causes for both batches, and it is decided to eliminate these batches from the data set and recompute the control limits. The following statements create a data set named Faillim2 that contains the revised control limits:

```
proc shewhart data=CircOne;
   where Batch ^= 10 and Batch ^= 19;
   pchart Fail*Batch / subgroupn = 500
                       nochart
                       outindex  ='Revised Limits'
                       outlimits = Faillim2;
run;


data Faillims;
   set Faillim1 Faillim2;
run;
```

The data set Faillims, which contains the true and revised control limits, is listed in Output 19.25.2.

Output 19.25.2 Listing of the Data Set Faillims

Proportion of Circuit Failures

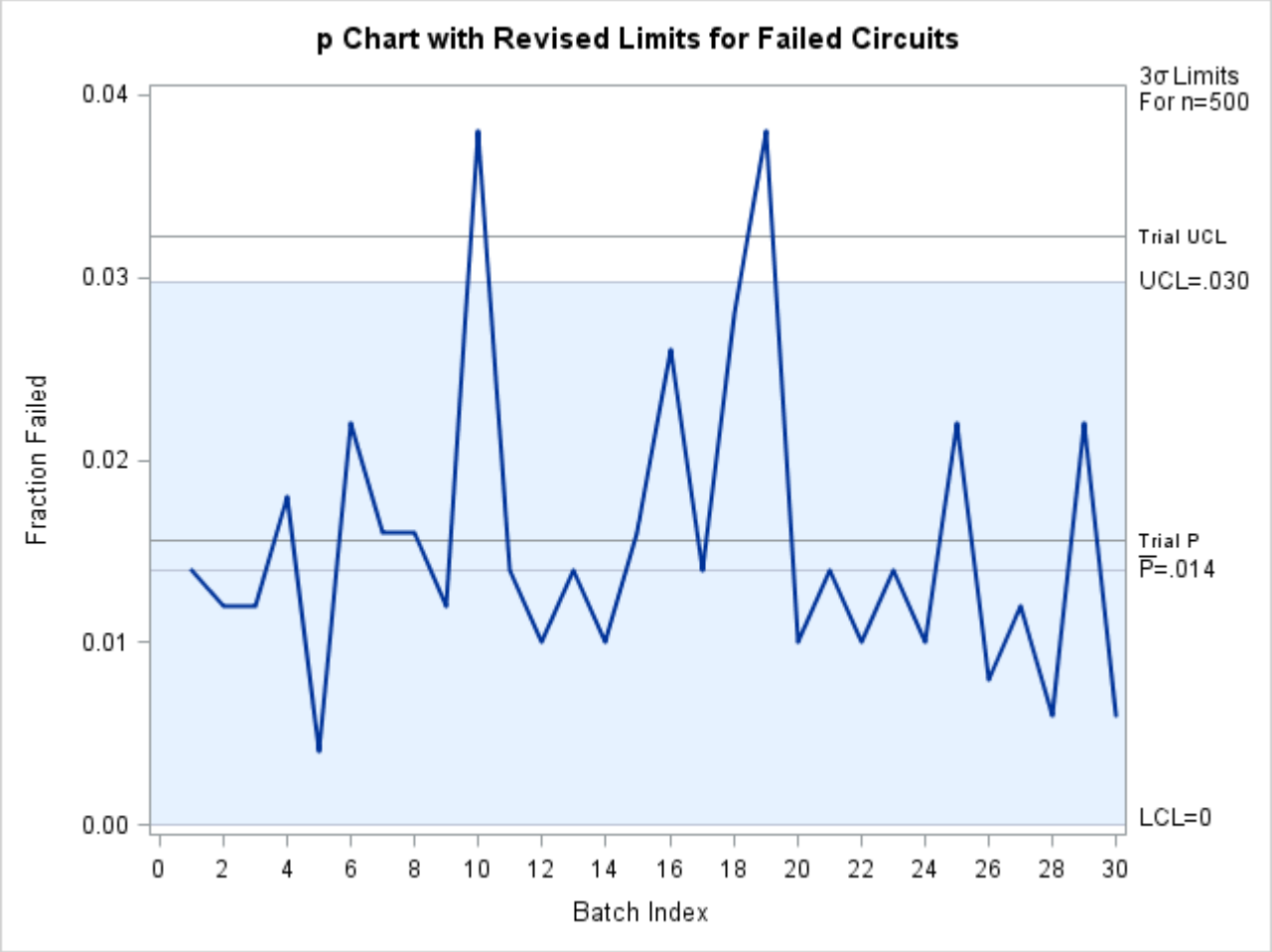
VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
Fail	Batch	Trial Limits	ESTIMATE	500	.002421958	3	0	0.0156	0.032226
Fail	Batch	Revised Limits	ESTIMATE	500	.002477491	3	0	0.0140	0.029763

The following statements create a *p* chart displaying both sets of control limits:

```
title 'p Chart with Revised Limits for Failed Circuits';
proc shewhart data=CircOne limits=Faillims;
   pchart Fail*Batch / subgroupn  = 500
                       readindex  = 'Revised Limits'
                       vref       = 0.0156 0.032226
                       vreflabels = ('Trial P' 'Trial UCL')
                       vreflabpos = 3
                       odstitle   = title
                       nolegend;
   label Fail  = 'Fraction Failed'
         Batch = 'Batch Index';
run;
ods graphics off;
```

The READINDEX= option is used to select the revised limits displayed on the *p* chart in Output 19.25.3. See "Displaying Multiple Sets of Control Limits" on page 2081. The VREF=, VREFLABELS=, and VREFLABPOS= options are used to display and label the trial limits. You can also pass in the values of the trial limits with macro variables. For an illustration of this technique, see Example 19.6.

Output 19.25.3 p Chart with Revised Limits



Example 19.26: OC Curve for Chart

NOTE: See *OC Curve for a p Chart* in the SAS/QC Sample Library.

This example uses the GPLOT procedure and the OUTLIMITS= data set Faillim2 from the previous example to plot an OC curve for the p chart shown in Output 19.25.3.

The OC curve displays β (the probability that p_i lies within the control limits) as a function of p (the true proportion nonconforming). The computations are exact, assuming that the process is in control and that the number of nonconforming items (X_i) has a binomial distribution.

The value of β is computed as follows:

$$\begin{aligned}
\beta &= P(p_i \leq \mathrm{UCL}) - P(p_i < \mathrm{LCL}) \\
&= P(X_i \leq n\mathrm{UCL}) - P(X_i < n\mathrm{LCL}) \\
&= P(X_i < n\mathrm{UCL}) + P(X_i = n\mathrm{UCL}) - P(X_i < n\mathrm{LCL}) \\
&= I_{1-p}(n+1-n\mathrm{UCL}, n\mathrm{UCL}) + P(X_i = n\mathrm{UCL}) - I_{1-p}(n+1-n\mathrm{LCL}, n\mathrm{LCL}) \\
&= I_p(n\mathrm{LCL}, n+1-n\mathrm{LCL}) + P(X_i = n\mathrm{UCL}) - I_p(n\mathrm{UCL}, n+1-n\mathrm{UCL})
\end{aligned}$$

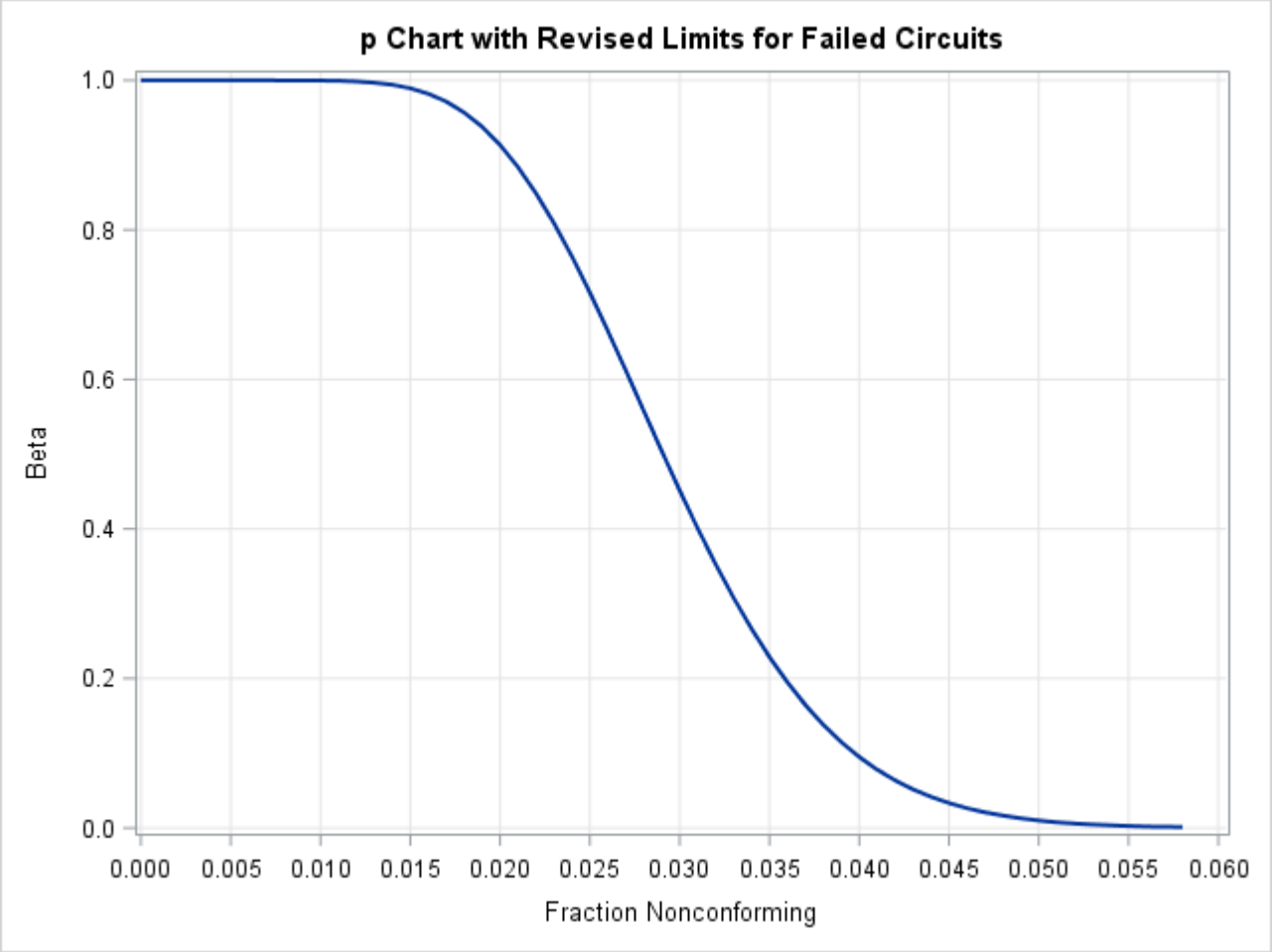
Here, $I_p(\cdot,\cdot)$ denotes the incomplete beta function. The following DATA step computes β (the variable BETA) as a function of p (the variable p):

```
data ocpchart;
   set Faillim2;
   keep beta fraction _lclp_ _p_ _uclp_;
   nucl=_limitn_*_uclp_;
   nlcl=_limitn_*_lclp_;
   do p=0 to 500;
      fraction=p/1000;
      if nucl=floor(nucl) then
         adjust=probbnml(fraction,_limitn_,nucl) -
                probbnml(fraction,_limitn_,nucl-1);
      else adjust=0;
      if nlcl=0 then
         beta=1 - probbeta(fraction,nucl,_limitn_-nucl+1) + adjust;
      else beta=probbeta(fraction,nlcl,_limitn_-nlcl+1) -
                probbeta(fraction,nucl,_limitn_-nucl+1) +
                adjust;
      if beta >= 0.001 then output;
   end;
   call symput('lcl', put(_lclp_,5.3));
   call symput('mean',put(_p_,   5.3));
   call symput('ucl', put(_uclp_,5.3));
run;
```

The following statements display the OC curve shown in Output 19.26.1:

```
proc sgplot data=ocpchart;
   series x=fraction y=beta / lineattrs=(thickness=2);
   xaxis values=(0 to 0.06 by 0.005) grid;
   yaxis grid;
   label fraction = 'Fraction Nonconforming'
         beta     = 'Beta';
run;
```

Output 19.26.1 OC Curve for p Chart



RCHART Statement: SHEWHART Procedure

Overview: RCHART Statement

The RCHART statement creates an *R* chart for subgroup ranges, which is used to analyze the variability of a process.[7]

You can use options in the RCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted ranges or as probability limits
- tabulate subgroup sample sizes, subgroup ranges, control limits, and other information

[7] You can also use *s* charts for this purpose; see "SCHART Statement: SHEWHART Procedure" on page 1767. In general, *s* charts are recommended with large subgroup sample sizes ($n_i \geq 10$).

- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- specify the method for estimating the process standard deviation
- specify a known (standard) process standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *R* charts with the RCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: RCHART Statement

This section introduces the RCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the RCHART statement is presented in the section "Syntax: RCHART Statement" on page 1742, and advanced examples are given in the section "Examples: RCHART Statement" on page 1761.

Creating Range Charts from Raw Data

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

A disk drive manufacturer performs a battery of tests to evaluate its drives. The following statements create a data set named Disks, which contains the time (in milliseconds) required to complete one of these tests for six drives in each of 25 lots:

```
data Disks;
   input Lot @;
   do i=1 to 6;
      input Time @;
      output;
   end;
   drop i;
   datalines;
 1 8.05 7.90 8.04 8.06 8.01 7.99
 2 8.03 8.06 8.02 8.02 7.97 8.03
 3 8.00 7.94 7.97 7.95 8.00 8.01
 4 8.00 8.06 8.06 7.99 7.97 7.96
 5 7.93 8.01 8.00 8.09 8.06 8.02
 6 7.98 7.99 8.01 8.09 8.00 7.97
 7 8.00 7.94 7.93 8.03 7.93 8.08
 8 8.01 7.98 7.98 8.07 8.05 8.09
 9 7.97 7.96 8.01 8.11 8.06 8.07
10 7.93 8.03 8.03 8.00 7.93 8.03
11 8.00 8.00 8.02 7.92 7.98 8.01
12 7.98 7.93 8.01 7.97 8.02 8.00
13 8.06 7.93 7.98 7.98 8.02 7.96
14 8.05 7.98 8.05 7.99 7.95 7.99
15 7.94 8.01 7.97 8.04 7.91 8.03
16 8.03 8.03 8.02 8.06 8.00 7.97
17 8.03 7.94 8.05 8.05 8.04 7.94
18 7.99 7.99 7.86 7.99 8.06 8.03
19 7.95 7.96 7.99 7.96 7.94 8.12
20 8.03 8.07 7.98 7.97 8.00 8.04
21 8.04 7.90 8.03 8.02 7.98 7.97
22 7.95 8.05 7.98 8.01 7.97 8.15
23 8.06 8.00 8.03 8.02 7.99 7.95
24 7.97 8.02 8.00 7.96 7.96 8.00
25 8.12 7.97 7.99 8.09 8.05 8.00
;
```

A partial listing of Disks is shown in Figure 19.70.

Figure 19.70 Partial Listing of the Data Set Disks

The Data Set DISKS

Lot	Time
1	8.05
1	7.90
1	8.04
1	8.06
1	8.01
1	7.99
2	8.03
2	8.06
2	8.02
2	8.02
2	7.97
2	8.03
3	8.00
3	7.94
3	7.97
3	7.95
3	8.00
3	8.01

The data set Disks is said to be in "strung-out" form because each observation contains the lot number and test time for a single disk drive. The first five observations contain the times for the first lot, the second five observations contain the times for the second lot, and so on. Because the variable Lot classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Time contains the time measurements and is referred to as the *process variable* (or *process* for short).

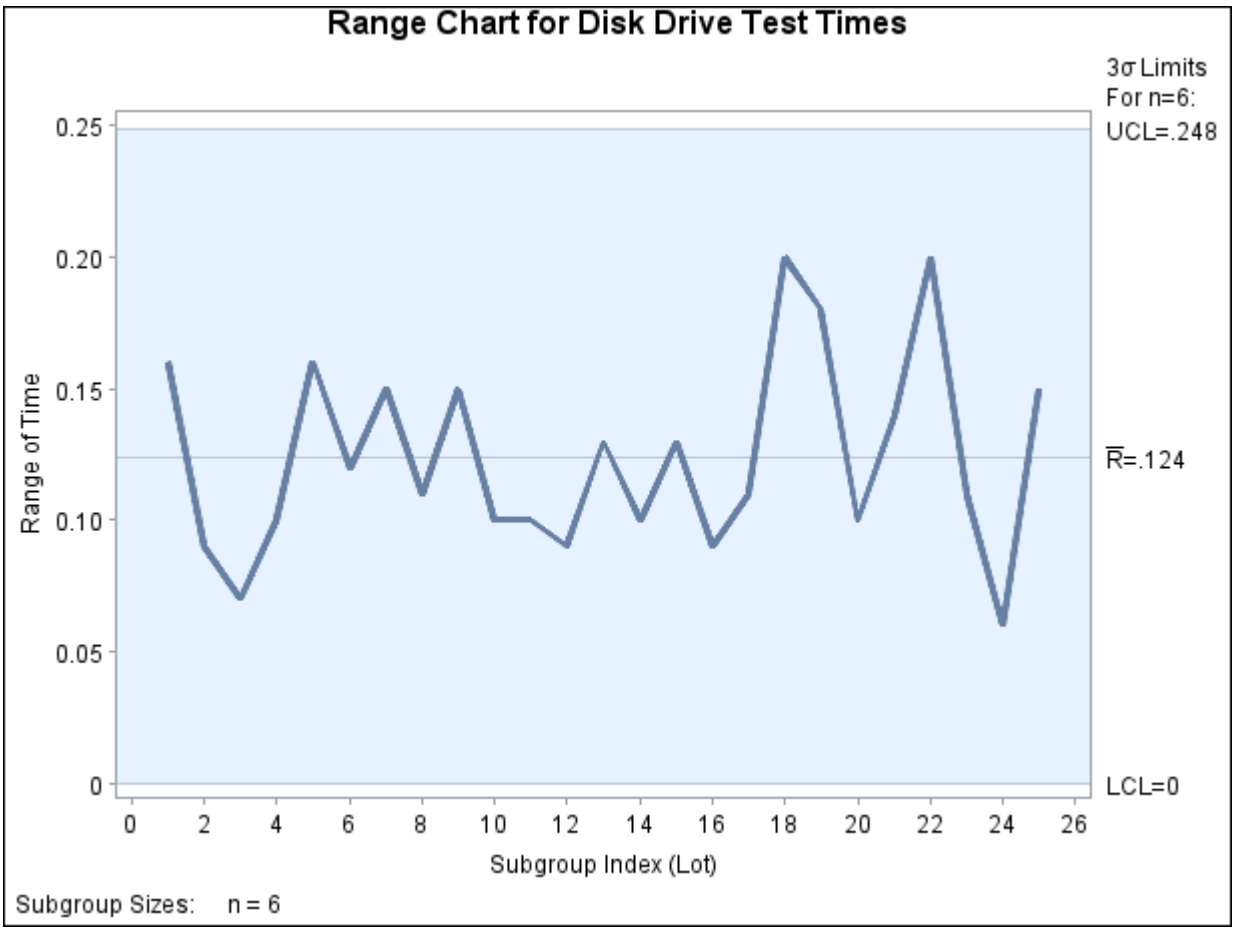
You can use an *R* chart to determine whether the variability in the performance of the disk drives is in control. The following statements create the *R* chart shown in Figure 19.71:

```
ods graphics off;
title 'Range Chart for Disk Drive Test Times';
proc shewhart data=Disks;
   rchart Time*Lot;
run;
```

This example illustrates the basic form of the RCHART statement. After the keyword RCHART, you specify the *process* to analyze (in this case, Time), followed by an asterisk and the *subgroup-variable* (Lot).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 19.71 *R* Chart for the Data Set Disks (Traditional Graphics)



Each point on the *R* chart represents the range of the measurements for a particular lot. For instance, the range plotted for the first lot is $8.06 - 7.90 = 0.16$. Because all of the subgroup ranges lie within the control limits, you can conclude that the variability in the performance of the disk drives is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.49. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1740.

For computational details, see "Constructing Range Charts" on page 1753. For more details on reading raw data, see "DATA= Data Set" on page 1757.

Creating Range Charts from Summary Data

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create *R* charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the RCHART statement with data of this type.

The following data set (Disksum) provides the data from the preceding example in summarized form:

```
data Disksum;
   input Lot TimeX TimeR;
   TimeN=6;
   datalines;
  1  8.00833  0.16
  2  8.02167  0.09
  3  7.97833  0.07
  4  8.00667  0.10
  5  8.01833  0.16
  6  8.00667  0.12
  7  7.98500  0.15
  8  8.03000  0.11
  9  8.03000  0.15
 10  7.99167  0.10
 11  7.98833  0.10
 12  7.98500  0.09
 13  7.98833  0.13
 14  8.00167  0.10
 15  7.98333  0.13
 16  8.01833  0.09
 17  8.00833  0.11
 18  7.98667  0.20
 19  7.98667  0.18
 20  8.01500  0.10
 21  7.99000  0.14
 22  8.01833  0.20
 23  8.00833  0.11
 24  7.98500  0.06
 25  8.03667  0.15
;
```

A partial listing of Disksum is shown in Figure 19.72. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Lot). The variable TimeX contains the subgroup means, the variable TimeR contains the subgroup ranges, and the variable TimeN contains the subgroup sample sizes (these are all six).

Figure 19.72 The Summary Data Set Disksum

The Summary Data Set of Disk Drive Test Times

Lot	TimeX	TimeR	TimeN
1	8.00833	0.16	6
2	8.02167	0.09	6
3	7.97833	0.07	6
4	8.00667	0.10	6
5	8.01833	0.16	6

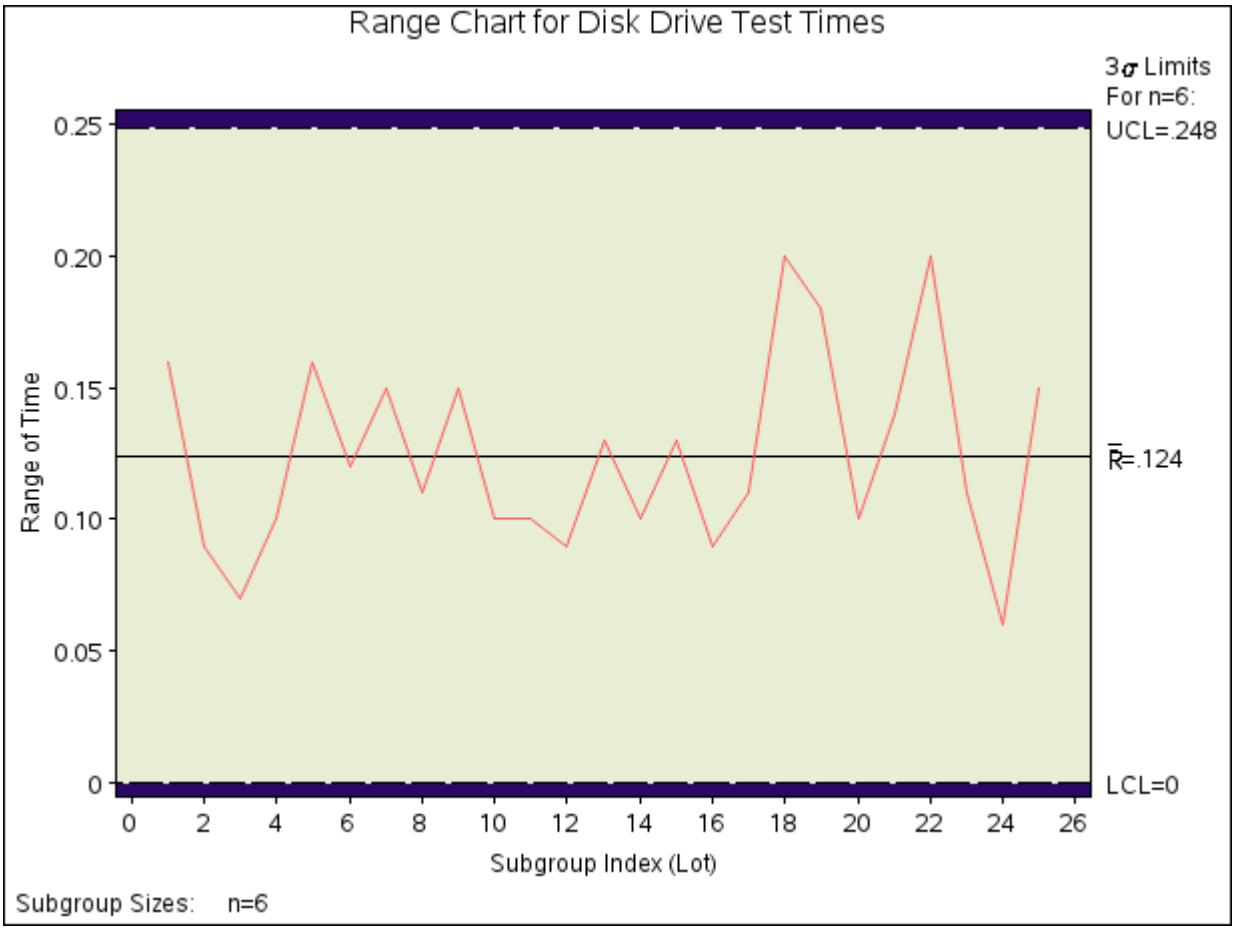
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
symbol color = rose h = .8;
title 'Range Chart for Disk Drive Test Times';
proc shewhart history=Disksum;
   rchart Time*Lot / cframe   = vipb
                     cinfill  = ywh
                     cconnect = rose;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and RCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *R* chart is shown in Figure 19.73.

Note that Time is *not* the name of a SAS variable in the data set Disksum but is, instead, the common prefix for the names of the SAS variables TimeR and TimeN. The suffix characters *R* and *N* indicate *range* and *sample size*, respectively. Thus, you can specify two subgroup summary variables in the HISTORY= data set with a single name (Time), which is referred to as the *process*. The name Lot specified after the asterisk is the name of the *subgroup-variable*.

Figure 19.73 *R* Chart from the Summary Data Set Disksum (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the RCHART statement must contain the following variables:

- subgroup variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup range and sample size variables must begin with the *process* name specified in the RCHART statement and end with the special suffix characters *R* and *N*, respectively. If the names do not follow this convention, you can use the RENAME optionin the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1887).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1758.

Saving Summary Statistics

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

In this example, the RCHART statement procedure is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Disks and create a summary data set named Diskhist:

```
proc shewhart data=Disks;
   rchart Time*Lot / outhistory = Diskhist
                     nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.71. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the RCHART statement. A complete list of options is presented in the section "Syntax: RCHART Statement" on page 1742.

Figure 19.74 contains a partial listing of Diskhist.

Figure 19.74 The Summary Data Set Diskhist

Summary Data Set for Disk Times

Lot	TimeX	TimeR	TimeN
1	8.00833	0.16	6
2	8.02167	0.09	6
3	7.97833	0.07	6
4	8.00667	0.10	6
5	8.01833	0.16	6

There are four variables in the data set Diskhist.

- Lot contains the subgroup index.
- TimeX contains the subgroup means.
- TimeR contains the subgroup ranges.
- TimeN contains the subgroup sample sizes.

The subgroup mean variable is included in the OUTHISTORY= data set even though it is not required by the RCHART statement. This enables the data set to be used as a HISTORY= data set with the BOXCHART, XCHART, and XRCHART statements, as well as with the RCHART statement. Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* Time specified in the RCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1755.

Saving Control Limits

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

You can save the control limits for an *R* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1740) or modify the limits with a DATA step program.

The following statements read measurements from the data set Disks (see "Creating Range Charts from Raw Data" on page 1731) and save the control limits displayed in Figure 19.71 in a data set named Disklim:

```
title 'Control Limits for Disk Times';
proc shewhart data=Disks;
   rchart Time*Lot / outlimits = Disklim
                     nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Disklim is listed in Figure 19.75.

Figure 19.75 The Data Set Disklim Containing Control Limit Information

Control Limits for Disk Times

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Time	Lot	ESTIMATE	6	.004447667	3	7.94314	8.00307	8.06299

LCLR	_R_	_UCLR_	_STDDEV_
0	0.124	0.24847	0.048927

The data set Disklim contains one observation with the limits for *process* Time. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits, and the variable _R_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. The variables _LCLX_ and _UCLX_, which contain the lower and upper control limits for subgroup means, are included so that the data set Disklim can be used to create an $\bar{X}$ chart (see "XRCHART Statement: SHEWHART Procedure" on page 1881). For more information, see "OUTLIMITS= Data Set" on page 1754.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limit Information';
proc shewhart data=Disks;
  rchart Time*Lot / outtable=Disktab
                    nochart;
run;
```

The data set Disktab is listed in Figure 19.76.

Figure 19.76 The Data Set Disktab

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLR_	_SUBR_	_R_	_UCLR_	_STDDEV_	_EXLIM_
Time	1	3	6	6	0	0.16	0.124	0.24847	0.048927	
Time	2	3	6	6	0	0.09	0.124	0.24847	0.048927	
Time	3	3	6	6	0	0.07	0.124	0.24847	0.048927	
Time	4	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	5	3	6	6	0	0.16	0.124	0.24847	0.048927	
Time	6	3	6	6	0	0.12	0.124	0.24847	0.048927	
Time	7	3	6	6	0	0.15	0.124	0.24847	0.048927	
Time	8	3	6	6	0	0.11	0.124	0.24847	0.048927	
Time	9	3	6	6	0	0.15	0.124	0.24847	0.048927	
Time	10	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	11	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	12	3	6	6	0	0.09	0.124	0.24847	0.048927	
Time	13	3	6	6	0	0.13	0.124	0.24847	0.048927	
Time	14	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	15	3	6	6	0	0.13	0.124	0.24847	0.048927	
Time	16	3	6	6	0	0.09	0.124	0.24847	0.048927	
Time	17	3	6	6	0	0.11	0.124	0.24847	0.048927	
Time	18	3	6	6	0	0.20	0.124	0.24847	0.048927	
Time	19	3	6	6	0	0.18	0.124	0.24847	0.048927	
Time	20	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	21	3	6	6	0	0.14	0.124	0.24847	0.048927	
Time	22	3	6	6	0	0.20	0.124	0.24847	0.048927	
Time	23	3	6	6	0	0.11	0.124	0.24847	0.048927	
Time	24	3	6	6	0	0.06	0.124	0.24847	0.048927	
Time	25	3	6	6	0	0.15	0.124	0.24847	0.048927	

This data set contains one observation for each subgroup sample. The variables _SUBR_ and _SUBN_ contain the subgroup ranges and subgroup sample sizes. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits, and the variable _R_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1756. An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Disktab and display an *R* chart (not shown here) identical to the chart in Figure 19.71:

```
title 'Range Chart for Disk Drive Test Times';
proc shewhart table=Disktab;
   rchart Time*Lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1759.

Reading Preestablished Control Limits

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Disklim saved control limits computed from the measurements in Disks. This example shows how these limits can be applied to new data provided in the following data set:

```
data Disks2;
   input Lot @;
   do i=1 to 6;
      input Time @;
      output;
   end;
   drop i;
   datalines;
26 7.93 7.97 7.89 7.81 7.88 7.92
27 7.86 7.91 7.87 7.89 7.83 7.87
28 7.93 7.95 7.90 7.89 7.88 7.90
29 7.97 8.00 7.86 7.89 7.84 7.78
30 7.91 7.93 7.98 7.93 7.83 7.88
31 7.85 7.94 7.88 7.98 7.96 7.84
32 7.86 8.01 7.88 7.95 7.90 7.89
33 7.87 7.93 7.96 7.89 7.81 8.00
34 7.87 7.97 7.95 7.89 7.92 7.84
35 7.92 7.97 7.90 7.88 7.89 7.86
36 7.96 7.90 7.90 7.84 7.90 8.00
37 7.92 7.90 7.98 7.92 7.94 7.94
38 7.88 7.99 8.02 7.98 7.88 7.92
39 7.89 7.91 7.92 7.90 7.94 7.94
40 7.84 7.88 7.91 7.98 7.87 7.93
41 7.91 7.87 7.96 7.91 7.89 7.92
42 7.96 7.93 7.86 7.93 7.86 7.94
43 7.84 7.82 7.87 7.91 7.91 8.01
44 7.93 7.91 7.92 7.88 7.91 7.86
45 7.95 7.92 7.93 7.90 7.86 8.00
;
```

The following statements create an *R* chart using the control limits in Disklim:

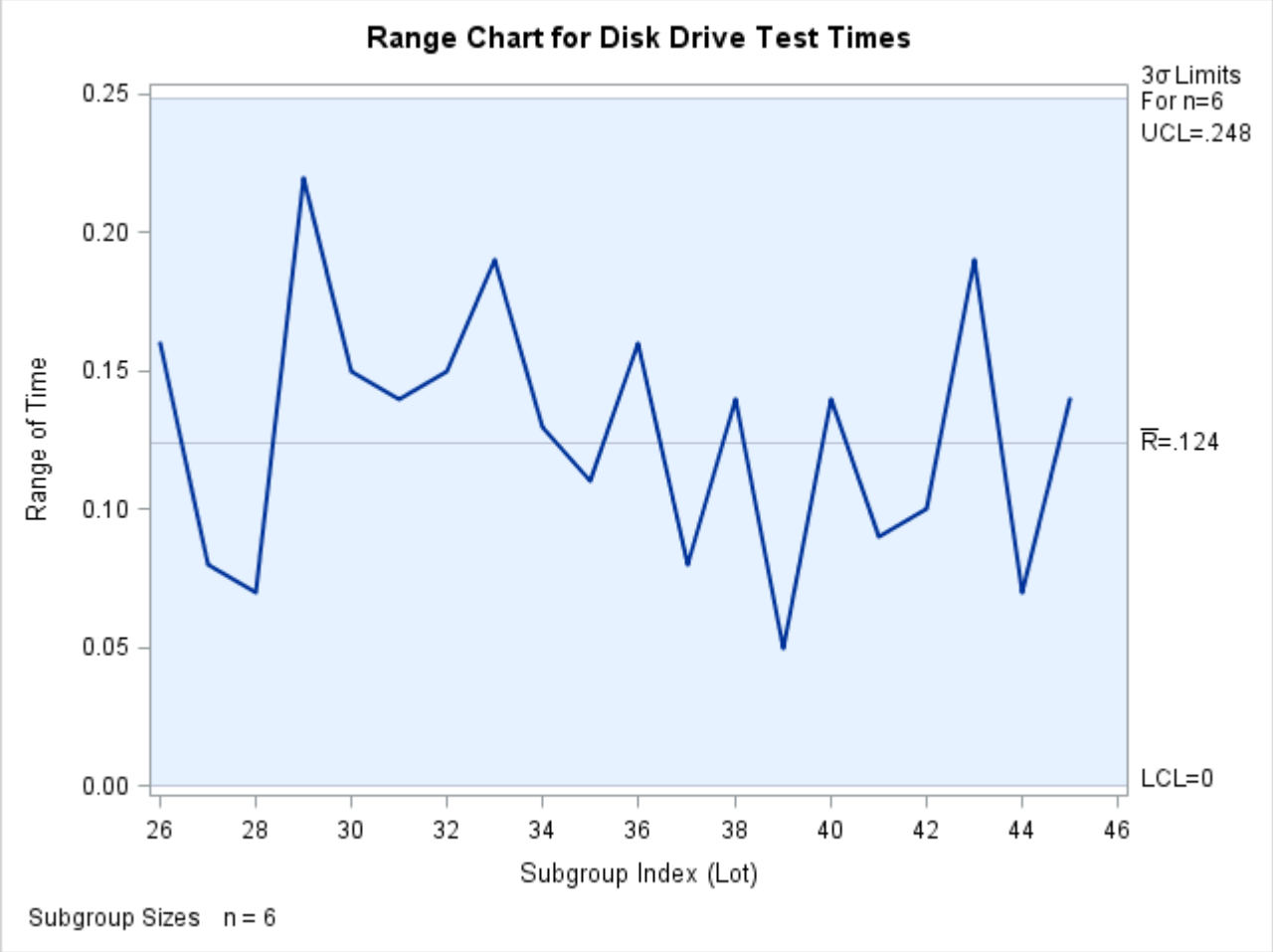
```
ods graphics on;
title 'Range Chart for Disk Drive Test Times';
proc shewhart data=Disks2 limits=Disklim;
   rchart Time*Lot / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *R* chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 19.77.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Time
- the value of _SUBGRP_ matches the *subgroup-variable* name Lot

Figure 19.77 *R* Chart for Second Set of Disk Drive Test Times (ODS Graphics)



All the ranges lie within the control limits, indicating that the variability in disk drive performance is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See Example 19.28 and "LIMITS= Data Set" on page 1758 for details concerning the variables that you must provide.

Syntax: RCHART Statement

The basic syntax for the RCHART statement is as follows:

RCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

RCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of RCHART statements in the SHEWHART procedure. The components of the RCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Range Charts from Raw Data" on page 1731.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Range Charts from Summary Data" on page 1733.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1737.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct *R* charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   rchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding RCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the ranges.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create an *R* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   rchart Weight*Day='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the RCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.47 RCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)

Table 19.47 *continued*

Option	Description
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMIT0	Suppresses display of zero lower control limit on R chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
RSYMBOL=	Specifies label for central line on R chart
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
Process Mean and Standard Deviation Options	
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL2=	Labels every point on R chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line

Table 19.47 *continued*

Option	Description
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL2=	Labels points outside control limits on *R* chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS2=	Specifies tests for special causes for the *R* chart
TEST2RESET=	Enables tests for special causes to be reset for the *R* chart
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied to the *R* chart
ZONE2LABELS	Adds labels A, B, and C to zone lines for *R* chart
ZONES2	Adds lines to *R* chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONE2VALUES	Labels *R* zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive

Table 19.47 *continued*

Option	Description
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default to *R* chart
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis

Table 19.47 *continued*

Option	Description
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays R chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= options
CVREF=	Specifies color for lines requested by VREF= = options
HREF=	Specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines

Table 19.47 *continued*

Option	Description
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to chart
DESCRIPTION=	Specifies description of *R* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of *R* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color

Table 19.47 *continued*

Option	Description
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY2=	Specifies a subset of subgroups to be plotted with markers in the *R* chart
MARKERLABEL2=	Specifies labels for subgroups that are plotted with markers in the *R* chart
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency

Table 19.47 *continued*

Option	Description
URL=	Specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend

Table 19.47 *continued*

Option	Description
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars

Table 19.47 *continued*

Option	Description
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: RCHART Statement

The following sections provide details that are specific to the RCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Range Charts

The following notation is used in this section:

σ	Process standard deviation (standard deviation of the population of measurements)
R_i	Range of measurements in ith subgroup
n_i	Sample size of ith subgroup
$d_2(n)$	Expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	Standard error of the range of n independent observations from a normal population with unit standard deviation
$D_p(n)$	100pth percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on an R chart indicates the value of a subgroup range (R_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $R_{10} = 19 - 12 = 7$.

Central Line

By default, the central line for the ith subgroup indicates an estimate of the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that R_i exceeds the limits

The following table provides the formulas for the limits:

Table 19.49 Limits for R Charts

Control Limits
LCL = lower limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$
Probability Limits
LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$ UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

The formulas assume that the data are normally distributed. Note that the control limits vary with n_i and that the probability limits for R_i are asymmetric around the central line. If a standard value σ_0 is available for σ, replace $\hat{\sigma}$ with σ_0 in Table 19.49.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.50 lists the variables that are saved.

Table 19.50 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_CP_	Capability index C_p
CPK	Capability index C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
INDEX	Optional identifier for the control limits with the OUTINDEX= option
LCLR	Lower control limit for subgroup range
LCLX	Lower control limit for subgroup mean
LIMITN	Sample size associated with the control limits
LSL	Lower specification limit
MEAN	Process mean ($\overline{\overline{X}}$)
_R_	Value of central line on R chart
SIGMAS	Multiple (k) of standard error of R_i
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the RCHART statement
TARGET	Target value
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	Upper control limit for subgroup range
UCLX	Upper control limit for subgroup mean
USL	Upper specification limit
VAR	*Process* specified in the RCHART statement

Notes:

1. The variables _LCLX_, _MEAN_, and _UCLX_ are saved to enable the OUTLIMITS= data set to be

used as a LIMITS= data set with the BOXCHART, XCHART, and XRCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, and _UCLR_.

3. If the limits are defined in terms of a multiple *k* of the standard error of R_i, the value of _ALPHA_ is computed as

$$F_R(\text{\_LCLR\_}/\text{\_STDDEV\_}) + 1 - F_R(\text{\_UCLR\_}/\text{\_STDDEV\_})$$

   where $F_R(\cdot)$ is the cumulative distribution function of the range of a sample of *n* observations from a normal population with unit standard deviation, and *n* is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLR\_} - \text{\_R\_})/e$, where *e* is the standard error of the range of *n* observations from a normal population with unit standard deviation. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the RCHART statement. For an example, see "Saving Control Limits" on page 1737.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

The subgroup mean variable is saved so that the data set can be reused as a HISTORY= data set with the BOXCHART, XCHART, and XRCHART statements, as well as the RCHART statement.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the RCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   rchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthR, WidthN, DiameterX, DiameterR, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1736.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.51 lists the variables that are saved.

Table 19.51 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on *R* chart
LCLR	Lower control limit for range
LIMITN	Nominal sample size associated with the control limits
_R_	Average range
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample sizes
SUBR	Subgroup range
TESTS2	Tests for special causes signaled on *R* chart
UCLR	Upper control limit for range
VAR	*Process* specified in the RCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables

- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS2_ is saved if you specify the TESTS2= option.
3. The variables _EXLIM_ and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1737.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the RCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped in subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the RCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a raw measurement for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating Range Charts from Raw Data" on page 1731.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   rchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 19.50. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variable _STDDEV_, which is used to calculate the control limits according to the equations in Table 19.49

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1740.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the RCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup range and subgroup sample size variables must be the prefix *process* concatenated with the special suffix characters *R* and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   rchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightR, WeightN, YieldstrengthR, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as phases) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Range Charts from Summary Data" on page 1733.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.52 lists the variables required in a TABLE= data set used with the RCHART statement.

Table 19.52 Variables Required in a TABLE= Data Set

Variable	Description
LCLR	Lower control limit for range
LIMITN	Nominal sample size associated with the control limits
_R_	Average range
subgroup-variable	Values of the *subgroup-variable*

Table 19.52 *continued*

Variable	Description
SUBN	Subgroup sample size
SUBR	Subgroup range
UCLR	Upper control limit for range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1737.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, two methods (referred to as default and MVLUE) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i} (x_{ij}) - \min_{1 \leq j \leq n_i} (x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

Examples: RCHART Statement

This section provides advanced examples of the RCHART statement.

Example 19.27: Computing Probability Limits

NOTE: See *An R Chart with Probability Limits* in the SAS/QC Sample Library.

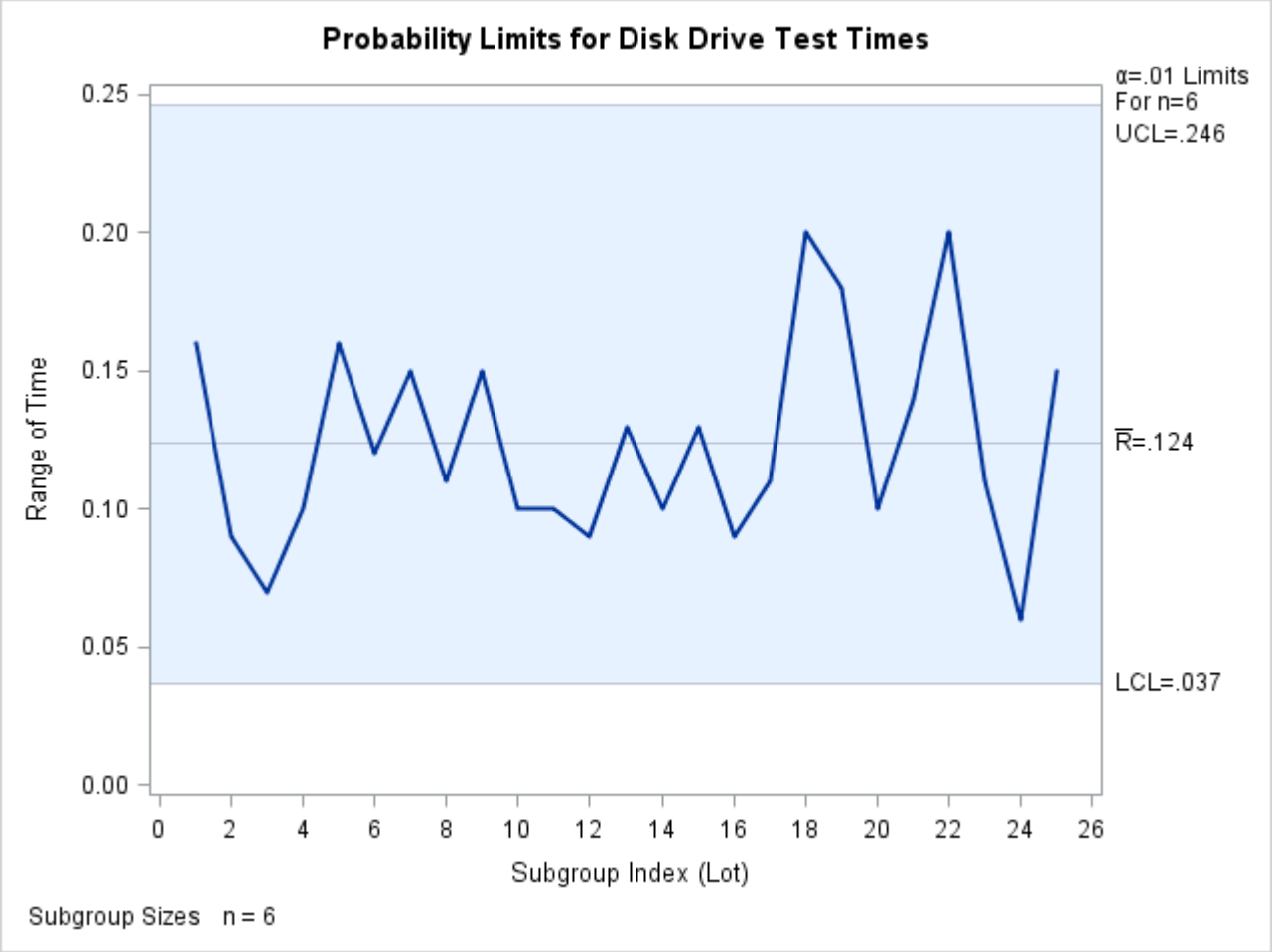
This example demonstrates how to create *R* charts with probability limits. The following statements read the disk drive test times from the data set Disks (see "Creating Range Charts from Raw Data" on page 1731) and create the *R* chart shown in Output 19.27.1:

```
ods graphics on;
title 'Probability Limits for Disk Drive Test Times';
proc shewhart data=Disks;
   rchart Time*Lot / alpha     = .01
                     outlimits = Dlimits
                     odstitle  = title;
run;
```

The ALPHA= option specifies the probability (α) that a subgroup range exceeds its limits. Here, the limits are computed so that the probability that a range is less than the lower limit is $\alpha/2 = 0.005$, and the probability that a range is greater than the upper limit is $\alpha/2 = 0.005$. This assumes that the measurements are normally distributed. The OUTLIMITS= option names an output data set that saves the probability limits. A listing of Dlimits is shown in Output 19.27.2.

The variable _ALPHA_ saves the value of α. Note that, in this case, the upper probability limit is equivalent to an upper 2.95σ limit.

Output 19.27.1 *R* Chart with Probability Limits



Output 19.27.2 Probability Limits Data Set

Probablity Limits for Disk Drive Test Times

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Time	Lot	ESTIMATE	6	0.01	2.94688	7.95162	8.00307	8.05452

LCLR	_R_	_UCLR_	_STDDEV_
0.036645	0.124	0.24627	0.048927

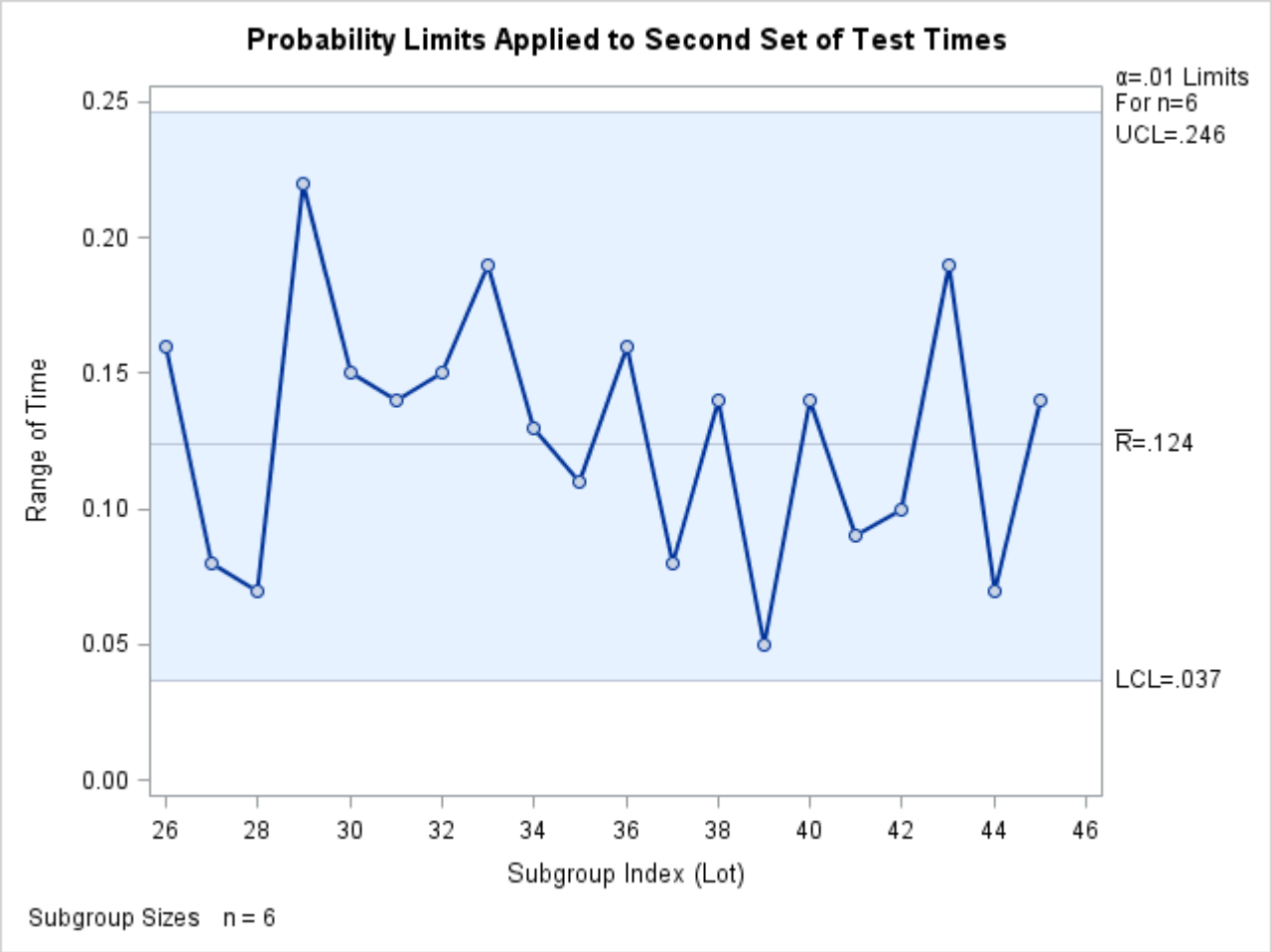
Because all the points fall within the probability limits, it can be concluded that the variability in the disk drive performance is in statistical control.

The following statements apply the limits in Dlimits to the times in the data set Disks2 (see "Reading Preestablished Control Limits" on page 1740):

```
title 'Probability Limits Applied to Second Set of Test Times';
proc shewhart data=Disks2 limits=Dlimits;
   rchart Time*Lot / readalpha
                     markers
                     odstitle = title;
run;
```

The READALPHA option specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from the LIMITS= data set. Thus the limits displayed in the chart, shown in Output 19.27.3, are probability limits.

Output 19.27.3 Reading Probability Limits from a LIMITS= Data Set



Example 19.28: Specifying Control Limit Information

NOTE: See *Specifying Control Limit Info for R Chart* in the SAS/QC Sample Library.

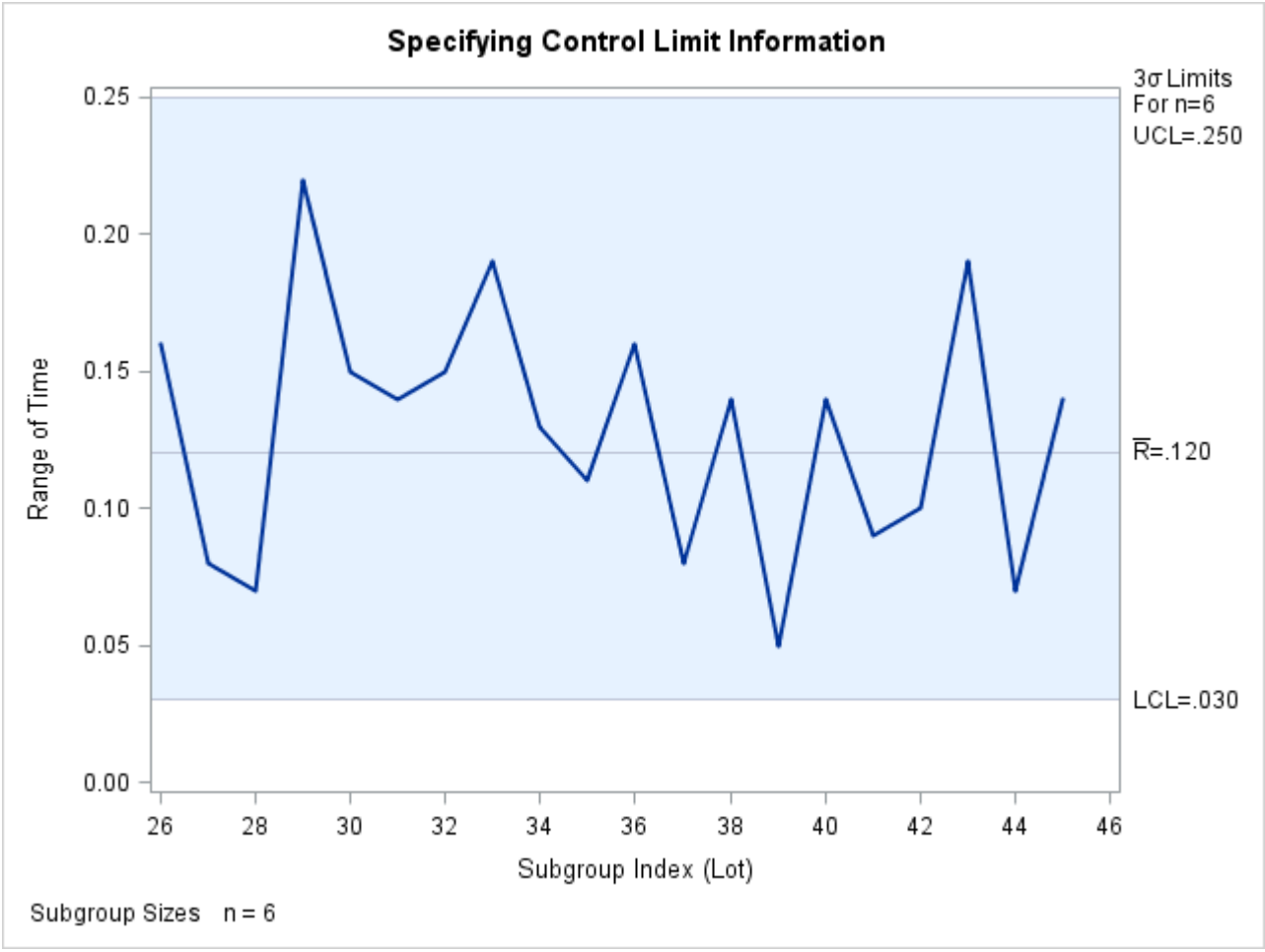
This example illustrates how you can use a DATA step program to create a LIMITS= data set. You can provide previously established values for the limits and central line with the variables _LCLR_, _R_, and _UCLR_, as in the following statements:

```
data Dlimits2;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Time';
   _subgrp_ = 'Lot';
   _type_   = 'STANDARD';
   _limitn_ = 6;
   _lclr_   = .03;
   _r_      = .12;
   _uclr_   = .25;
run;
```

The following statements apply the control limits in Dlimits2 to the measurements in Disks2 (see "Reading Preestablished Control Limits" on page 1740) and create the *R* chart shown in Output 19.28.1:

```
ods graphics on;
title 'Specifying Control Limit Information';
proc shewhart data=Disks2 limits=Dlimits2;
   rchart Time*Lot / odstitle=title;
run;
```

Output 19.28.1 Reading Control Limits from Dlimits2



In some cases, a standard value (σ_0) might be available for the process standard deviation. The following DATA step creates a data set named Dlimits3 that provides this value:

```
data Dlimits3;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Time';
   _subgrp_ = 'Lot';
   _stddev_ = .045;
   _limitn_ = 6;
   _type_   = 'STDSIGMA';
run;
```

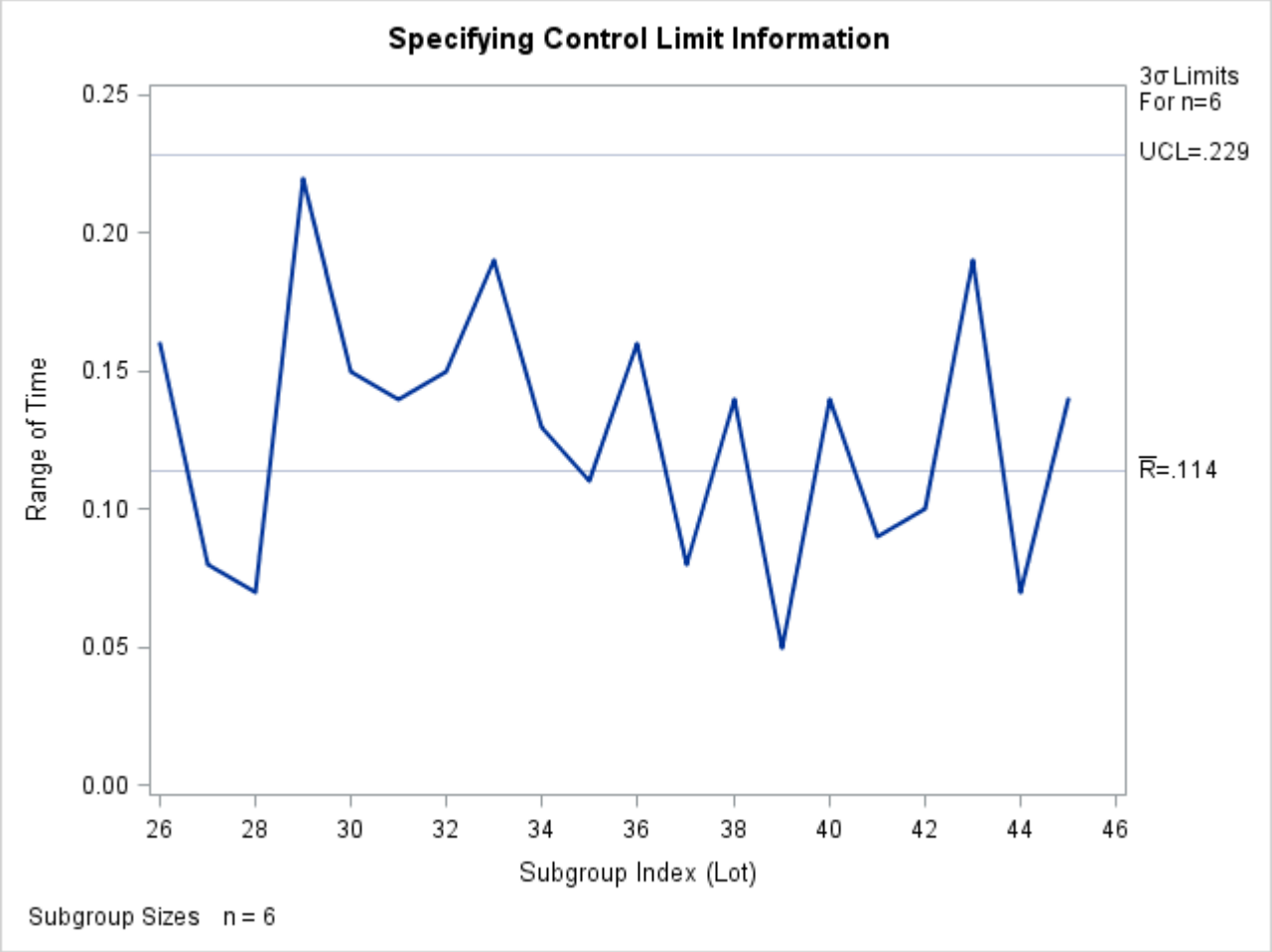
The variable _TYPE_ is a bookkeeping variable whose value indicates that the value of _STDDEV_ is a standard value rather than an estimate.

The following statements read the value of σ_0 from Dlimits3 and create the *R* chart shown in Output 19.28.2:

```
title 'Specifying Control Limit Information';
proc shewhart data=Disks2 limits=Dlimits3;
   rchart Time*Lot / nolimit0 odstitle=title;
run;
```

The NOLIMIT0 option suppresses the display of a fixed lower control limit if the value of the limit is zero (which is the case in this example).

Output 19.28.2 Reading in Standard Value for Process Standard Deviation



Instead of specifying σ_0 with the variable _STDDEV_ in a LIMITS= data set, you can use the SIGMA0= option in the RCHART statement. The following statements create an *R* chart identical to the chart shown in Output 19.28.2:

```
proc shewhart data=Disks;
   rchart Time*Lot / sigma0=.045 nolimit0;
run;
```

For more information, see "LIMITS= Data Set" on page 1758.

SCHART Statement: SHEWHART Procedure

Overview: SCHART Statement

The SCHART statement creates an *s* chart for subgroup standard deviations, which is used to analyze the variability of a process.[8]

You can use options in the SCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted standard deviations or as probability limits
- tabulate subgroup sample sizes, subgroup standard deviations, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read preestablished control limits from a data set
- specify a method for estimating the process standard deviation
- specify a known (standard) process standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *s* charts with the SCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

[8] You can also use *R* charts for this purpose; see "RCHART Statement: SHEWHART Procedure" on page 1729. In general, *s* charts are recommended with large subgroup sample sizes ($n_i \geq 10$).

Getting Started: SCHART Statement

This section introduces the SCHART statement with simple examples that illustrate commonly used options. Complete syntax for the SCHART statement is presented in the section "Syntax: SCHART Statement" on page 1778, and advanced examples are given in the section "Examples: SCHART Statement" on page 1798.

Creating Standard Deviation Charts from Raw Data

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

A petroleum company uses a turbine to heat water into steam, which is then pumped into the ground to make oil less viscous and easier to extract. This heating process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set named Turbine, which contains the power output measurements for 20 days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
   end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711

   ... more lines ...

23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
```

A partial listing of Turbine is shown in Figure 19.78.

Figure 19.78 Partial Listing of the Data Set Turbine

Kilowatt Power Output Data

Obs	Day	KWatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
5	04JUL	3583
6	04JUL	3617
7	04JUL	3789
8	04JUL	3180
9	04JUL	3505
10	04JUL	3454
11	04JUL	3417
12	04JUL	3199
13	04JUL	3613
14	04JUL	3384
15	04JUL	3475
16	04JUL	3316
17	04JUL	3556
18	04JUL	3607
19	04JUL	3364
20	04JUL	3721
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
24	05JUL	3193
25	05JUL	3635

The data set Turbine is said to be in "strung-out" form, because each observation contains the day and power output for a single heating. The first 20 observations contain the power outputs for the first day, the second 20 observations contain the power outputs for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWatts contains the power output measurements and is referred to as the *process variable* (or *process* for short).

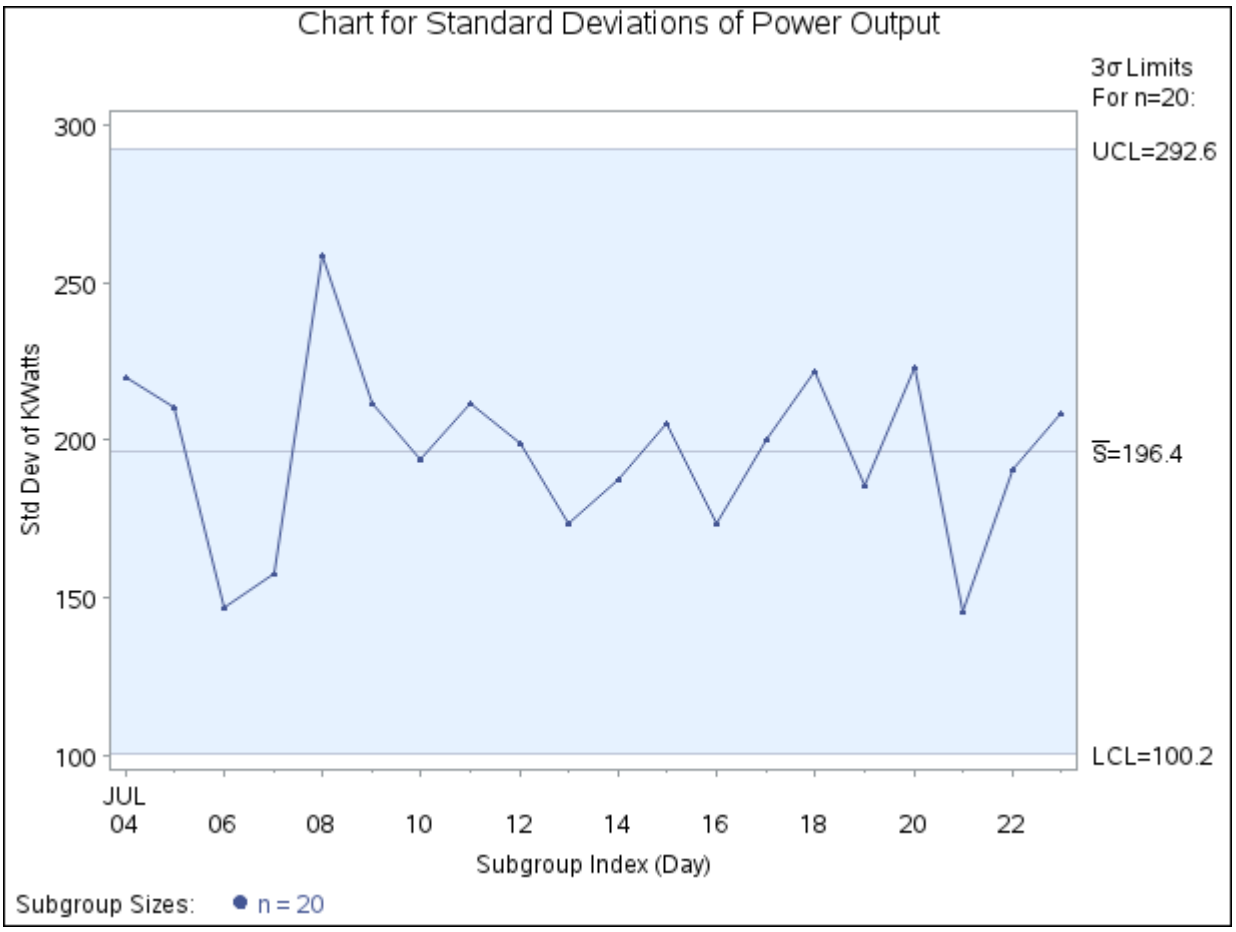
You can use an *s* chart to determine whether the variability in the heating process is in control. The following statements create the *s* chart shown in Figure 19.79:

```
ods graphics off;
symbol v=dot h=.8;
title 'Chart for Standard Deviations of Power Output';
proc shewhart data=Turbine;
   schart KWatts*Day;
run;
```

This example illustrates the basic form of the SCHART statement. After the keyword SCHART, you specify the *process* to analyze (in this case, KWatts), followed by an asterisk and the *subgroup-variable* (Day).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 19.79 *s* Chart for Power Output Data (Traditional Graphics)



Each point on the chart represents the standard deviation of the measurements for a particular day. For instance, the standard deviation plotted for the first day is

$$\sqrt{\frac{(3196-3487.4)^2+(3507-3487.4)^2+\cdots+(3721-3487.4)^2}{19}}=220.26$$

Because all of the subgroup standard deviations lie within the control limits, you can conclude that the variability of the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.55. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1776.

For computational details, see "Constructing Charts for Standard Deviations" on page 1789. For more details on reading raw data, see "DATA= Data Set" on page 1794.

Creating Standard Deviation Charts from Subgroup Summary Data

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

The previous example illustrates how you can create *s* charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the SCHART statement with data of this type.

The following data set (Oilsum) provides the data from the preceding example in summarized form:

```
data Oilsum;
   input Day KWattsX KWattsS KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day   ='Date of Measurement';
   datalines;
 04JUL94 3487.40 220.260 20
 05JUL94 3471.65 210.427 20
 06JUL94 3488.30 147.025 20
 07JUL94 3434.20 157.637 20
 08JUL94 3475.80 258.949 20
 09JUL94 3518.10 211.566 20
 10JUL94 3492.65 193.779 20
 11JUL94 3496.40 212.024 20
 12JUL94 3398.50 199.201 20
 13JUL94 3456.05 173.455 20
 14JUL94 3493.60 187.465 20
 15JUL94 3563.30 205.472 20
 16JUL94 3519.05 173.676 20
 17JUL94 3474.20 200.576 20
 18JUL94 3443.60 222.084 20
 19JUL94 3586.35 185.724 20
 20JUL94 3486.45 223.474 20
 21JUL94 3492.90 145.267 20
 22JUL94 3432.80 190.994 20
 23JUL94 3496.90 208.858 20
;
```

A partial listing of Oilsum is shown in Figure 19.80. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day). The variable KWattsX contains the subgroup means, the variable KWattsS contains the subgroup standard deviations, and the variable KWattsN contains the subgroup sample sizes (these are all 20).

Figure 19.80 The Summary Data Set Oilsum

Summary Data Set for Power Outputs

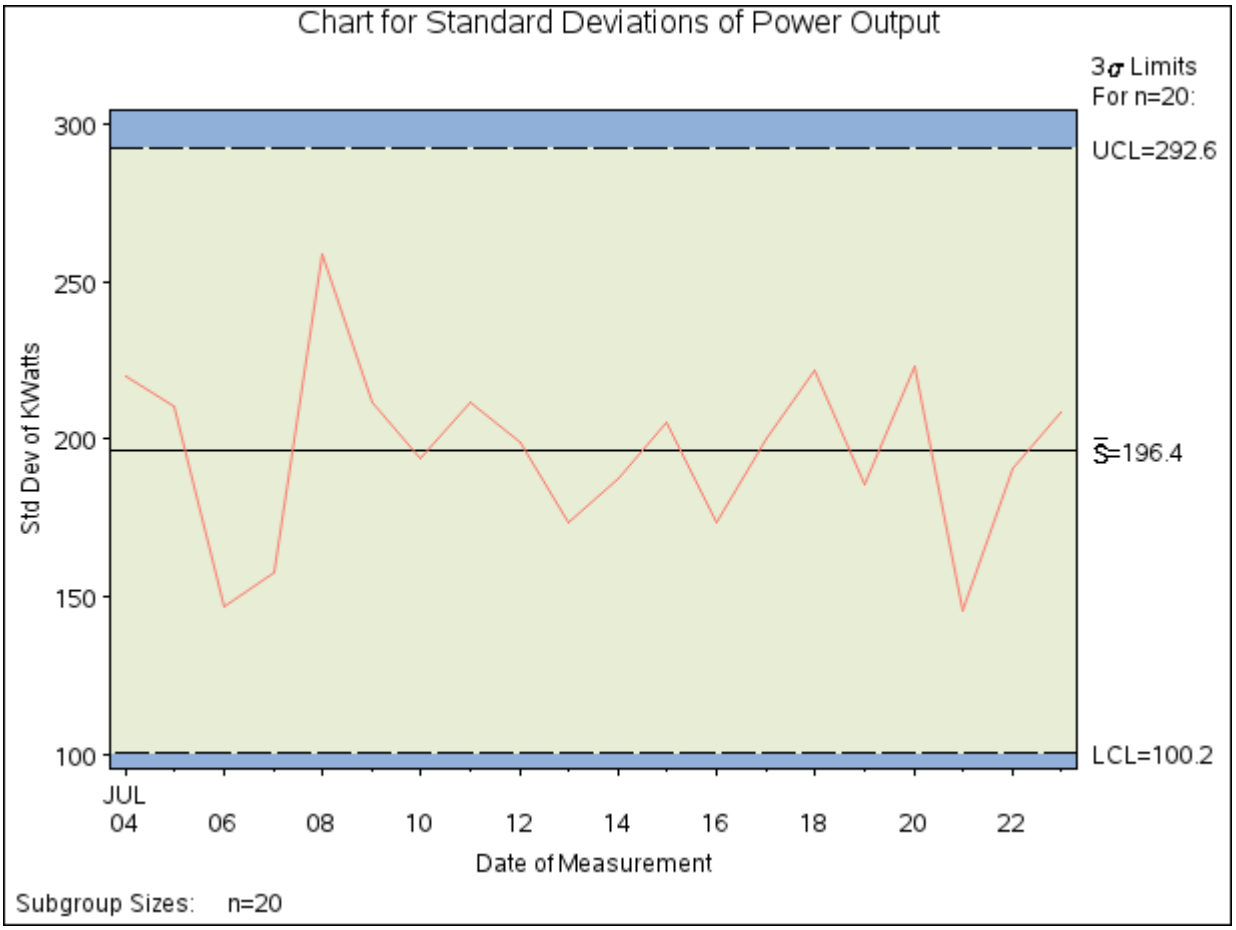
Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
title 'Chart for Standard Deviations of Power Output';
proc shewhart history=Oilsum;
   schart KWatts*Day / cframe  = vligb
                       cinfill = ywh
                       cconnect = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *s* chart is shown in Figure 19.81.

Figure 19.81 *s* Chart for Power Output Data (Traditional Graphics with NOGSTYLE)



Note that KWatts is *not* the name of a SAS variable in the data set Oilsum but is, instead, the common prefix for the names of the SAS variables KWattsS and KWattsN. The suffix characters *S* and *N* indicate *standard deviation* and *sample size*, respectively. Thus, you can specify two subgroup summary variables in

the HISTORY= data set with a single name (KWatts), which is referred to as the *process*. The name Day, specified after the asterisk, is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the SCHART statement must contain the following variables:

- subgroup variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup standard deviation and sample size variables must begin with the *process* name specified in the SCHART statement and end with the special suffix characters *S* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1885).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1795.

Saving Summary Statistics

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

In this example, the SCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Turbine and create a summary data set named Turbhist:

```
proc shewhart data=Turbine;
   schart KWatts*Day / outhistory = Turbhist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.79. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the SCHART statement. A complete list of options is presented in the section "Syntax: SCHART Statement" on page 1778.

Figure 19.82 contains a partial listing of Turbhist.

Figure 19.82 The Summary Data Set Turbhist

Summary Data Set for Power Output

Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

There are four variables in the data set Turbhist.

- Day contains the subgroup index.
- KWattsX contains the subgroup means.
- KWattsS contains the subgroup standard deviations.
- KWattsN contains the subgroup sample sizes.

The subgroup mean variable is included even though it is not required by the SCHART statement. This enables the data set to be used as a HISTORY= data set with the BOXCHART, XCHART, and XSCHART statements, as well as with the SCHART statement. Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *process* KWatts specified in the SCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1792.

Saving Control Limits

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

You can save the control limits for an *s* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1776) or modify the limits with a DATA step program.

The following statements read measurements from the data set Turbine (see "Creating Standard Deviation Charts from Raw Data" on page 1768) and save the control limits displayed in Figure 19.79 in a data set named Turblim:

```
proc shewhart data=Turbine;
   schart KWatts*Day / outlimits=Turblim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Turblim is listed in Figure 19.83.

Figure 19.83 The Data Set Turblim Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	.002792725	3	3351.92	3485.41	3618.90

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

The data set Turblim contains one observation with the limits for *process* KWatts. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits, and the variable _S_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. The variables _LCLX_ and _UCLX_, which contain the lower and upper control limits for subgroup means, are included so that the data set Turblim can be used to create an $\bar{X}$ chart (see "XSCHART Statement: SHEWHART Procedure" on page 1925). For more information, see "OUTLIMITS= Data Set" on page 1790.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Turbine;
   schart KWatts*Day / outtable=Turbtab
                       nochart;
run;
```

The data set Turbtab is listed in Figure 19.84.

Figure 19.84 The OUTTABLE= Data Set Turbtab

Summary Statistics and Control Limit Information

VAR	Day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLS_	_SUBS_	_S_	_UCLS_	_STDDEV_	_EXLIM_
KWatts	04JUL	3	20	20	100.207	220.260	196.396	292.584	198.996	
KWatts	05JUL	3	20	20	100.207	210.427	196.396	292.584	198.996	
KWatts	06JUL	3	20	20	100.207	147.025	196.396	292.584	198.996	
KWatts	07JUL	3	20	20	100.207	157.637	196.396	292.584	198.996	
KWatts	08JUL	3	20	20	100.207	258.949	196.396	292.584	198.996	
KWatts	09JUL	3	20	20	100.207	211.566	196.396	292.584	198.996	
KWatts	10JUL	3	20	20	100.207	193.779	196.396	292.584	198.996	
KWatts	11JUL	3	20	20	100.207	212.024	196.396	292.584	198.996	
KWatts	12JUL	3	20	20	100.207	199.201	196.396	292.584	198.996	
KWatts	13JUL	3	20	20	100.207	173.455	196.396	292.584	198.996	
KWatts	14JUL	3	20	20	100.207	187.465	196.396	292.584	198.996	
KWatts	15JUL	3	20	20	100.207	205.472	196.396	292.584	198.996	
KWatts	16JUL	3	20	20	100.207	173.676	196.396	292.584	198.996	
KWatts	17JUL	3	20	20	100.207	200.576	196.396	292.584	198.996	
KWatts	18JUL	3	20	20	100.207	222.084	196.396	292.584	198.996	
KWatts	19JUL	3	20	20	100.207	185.724	196.396	292.584	198.996	
KWatts	20JUL	3	20	20	100.207	223.474	196.396	292.584	198.996	
KWatts	21JUL	3	20	20	100.207	145.267	196.396	292.584	198.996	
KWatts	22JUL	3	20	20	100.207	190.994	196.396	292.584	198.996	
KWatts	23JUL	3	20	20	100.207	208.858	196.396	292.584	198.996	

This data set contains one observation for each subgroup sample. The variables _SUBS_ and _SUBN_ contain the subgroup standard deviations and subgroup sample sizes. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits, and the variable _S_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1793.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Turbtab and display an *s* chart (not shown here) identical to the chart in Figure 19.79:

```
title 'Chart for Standard Deviations of Power Output';
symbol v=dot;
proc shewhart table=Turbtab;
   schart KWatts*Day;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1796.

Reading Preestablished Control Limits

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Turblim saved control limits computed from the measurements in Turbine. This example shows how these limits can be applied to new data.

The following statements create an *s* chart for new measurements in the data set Turbine2 (not listed here) using the control limits in Turblim:

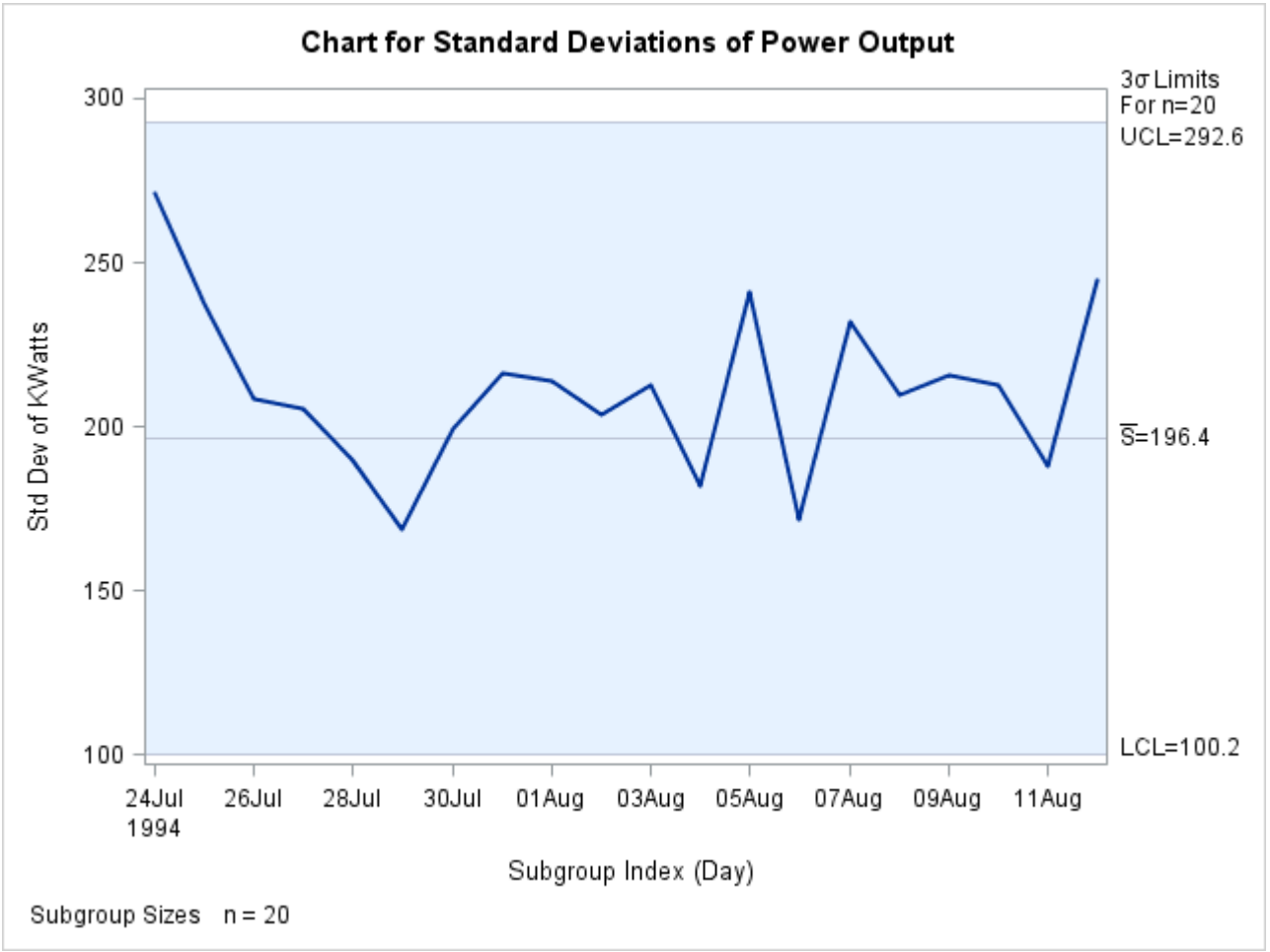
```
ods graphics on;
title 'Chart for Standard Deviations of Power Output';
proc shewhart data=Turbine2 limits=Turblim;
   schart KWatts*Day / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *s* chart is created by using ODS Graphics instead of traditional graphics. The chart is shown in Figure 19.85.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWatts
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

Figure 19.85 *s* Chart for Second Set of Power Output Data (ODS Graphics)



All the standard deviations lie within the control limits, indicating that the variability of the heating process is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1794 for details concerning the variables that you must provide.

Syntax: SCHART Statement

The basic syntax for the SCHART statement is as follows:

SCHART *process* * *subgroup-variable* **;**

The general form of this syntax is as follows:

SCHART *processes* * *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of SCHART statements in the SHEWHART procedure. The components of the SCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Standard Deviation Charts from Raw Data" on page 1768.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Standard Deviation Charts from Subgroup Summary Data" on page 1771.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1774.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct *s* charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   schart (Weight Length Width)*Day;
run;
```

subgroup-variable
is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding SCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables
are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the subgroup standard deviations.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create an *s* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   schart Weight*Day='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the SCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.53 SCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMIT0	Suppresses display of zero lower control limit on s chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
SSYMBOL=	Specifies label for central line on s chart
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
Process Mean and Standard Deviation Options	
SIGMA0=	Specifies known value σ_0 for process standard deviation σ

Table 19.53 *continued*

Option	Description
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL2=	Labels every point on *s* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL2=	Labels points outside control limits on *s* chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS2=	Specifies tests for special causes for the *s* chart
TEST2RESET=	Enables tests for special causes to be reset for the *s* chart
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive

Table 19.53 *continued*

Option	Description
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied to the *s* chart
ZONE2LABELS	Adds labels A, B, and C to zone lines for *s* chart
ZONES2	Adds lines to *s* chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONE2VALUES	Labels *s* zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis

Table 19.53 *continued*

Option	Description
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default to *s* chart
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel

Table 19.53 *continued*

Option	Description
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays s chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= options
CVREF=	Specifies color for lines requested by VREF= = options
HREF=	Specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to chart

Table 19.53 *continued*

Option	Description
DESCRIPTION=	Specifies description of *s* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of *s* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY2=	Specifies a subset of subgroups to be plotted with markers in the *s* chart
MARKERLABEL2=	Specifies labels for subgroups that are plotted with markers in the *s* chart
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote

Table 19.53 *continued*

Option	Description
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive

Table 19.53 *continued*

Option	Description
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars

Table 19.53 *continued*

Option	Description
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*

Table 19.53 *continued*

Option	Description
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: SCHART Statement

The following sections provide details that are specific to the SCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Standard Deviations

The following notation is used in this section:

σ	Process standard deviation (standard deviation of the population of measurements)
s_i	Standard deviation of measurements in ith subgroup

$$s_i = \sqrt{(1/(n_i - 1))((x_{i1} - \bar{X}_i)^2 + \cdots + (x_{in_i} - \bar{X}_i)^2)}$$

n_i	Sample size of ith subgroup
$c_4(n)$	Expected value of the standard deviation of n independent normally distributed variables with unit standard deviation
$c_5(n)$	Standard error of the standard deviation of n independent observations from a normal population with unit standard deviation
$\chi^2_p(n)$	100pth percentile ($0 < p < 1$) of the χ^2 distribution with n degrees of freedom

Plotted Points

Each point on an s chart indicates the value of a subgroup standard deviation (s_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 13, the value plotted for this subgroup is

$$s_{10} = \sqrt{((12-15)^2 + (15-15)^2 + (19-15)^2 + (16-15)^2 + (13-15)^2)/4} = 2.739$$

Central Line

By default, the central line for the ith subgroup indicates an estimate for the expected value of s_i, which is computed as $c_4(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $c_4(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of s_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that s_i exceeds the limits

The following table provides the formulas for the limits:

Table 19.55 Limits for s Charts

Control Limits
LCL = lower limit = $\max(c_4(n_i)\hat{\sigma} - kc_5(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $c_4(n_i)\hat{\sigma} + kc_5(n_i)\hat{\sigma}$
Probability Limits
LCL = lower limit = $\hat{\sigma}\sqrt{\chi^2_{\alpha/2}(n_i - 1)/(n_i - 1)}$ UCL = upper limit = $\hat{\sigma}\sqrt{\chi^2_{1-\alpha/2}(n_i - 1)/(n_i - 1)}$

The formulas assume that the data are normally distributed. If a standard value σ_0 is available for σ, replace $\hat{\sigma}$ with σ_0 in Table 19.55. Note that the upper and lower limits vary with n_i and that the probability limits are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 19.56 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or s_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the SCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the SCHART statement

Notes:

1. The variables _LCLX_, _MEAN_, and _UCLX_ are saved to enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XCHART, and XSCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLS_, _S_, and _UCLS_.

3. If the limits are defined in terms of a multiple k of the standard error of s_i, the value of _ALPHA_ is computed as

$$F_S(\text{\_LCLS\_}/\text{\_STDDEV\_}) + 1 - F_S(\text{\_UCLS\_}/\text{\_STDDEV\_})$$

   where $F_S(\cdot)$ is the cumulative distribution function of the standard deviation of a sample of n observations from a normal population with unit standard deviation, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _ALPHA_.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLS\_} - \text{\_S\_})/e$, where e is the standard error of the standard deviation of n observations from a normal population with unit standard deviation. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the SCHART statement. For an example, see "Saving Control Limits" on page 1774.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

The subgroup mean variable is included so that the data set can be reused as a HISTORY= data set with the BOXCHART, XCHART, and XSCHART statements, as well as the SCHART statement.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the SCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   schart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthS, WidthN, DiameterX, DiameterS, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1773.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on *s* chart
LCLS	lower control limit for standard deviation
LIMITN	nominal sample size associated with the control limits
_S_	average standard deviation
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
TESTS2	tests for special causes signaled on *s* chart
UCLS	upper control limit for standard deviation
VAR	*process* specified in the SCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS2_ is saved if you specify the TESTS2= option.
3. The variables _EXLIM_ and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1774.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the SCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements, which must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the SCHART statement, must also be a SAS variable in the DATA= data set.

Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079.

For an example of a DATA= data set, see "Creating Standard Deviation Charts from Raw Data" on page 1768.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   schart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 19.56. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLS_, _S_, and _UCLS_, which specify the control limits directly
- the variable _STDDEV_ , which is used to calculate the control limits according to the equations in Table 19.55

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'ESTIMATE', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1776.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the SCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup standard deviation variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup standard deviation and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *S* and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=Summary;
   schart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightS, WeightN, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*

- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Standard Deviation Charts from Subgroup Summary Data" on page 1771.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.57 lists the variables required in a TABLE= data set used with the SCHART statement.

Table 19.57 Variables Required in a TABLE= Data Set

Variable	Description
LCLS	Lower control limit for standard deviation
LIMITN	Nominal sample size associated with the control limits
_S_	Average standard deviation
subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
UCLS	Upper control limit for standard deviation

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.

- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1774.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Examples: SCHART Statement

This section provides advanced examples of the SCHART statement.

Example 19.29: Specifying a Known Standard Deviation

NOTE: See *s Chart with Known Standard Deviation* in the SAS/QC Sample Library.

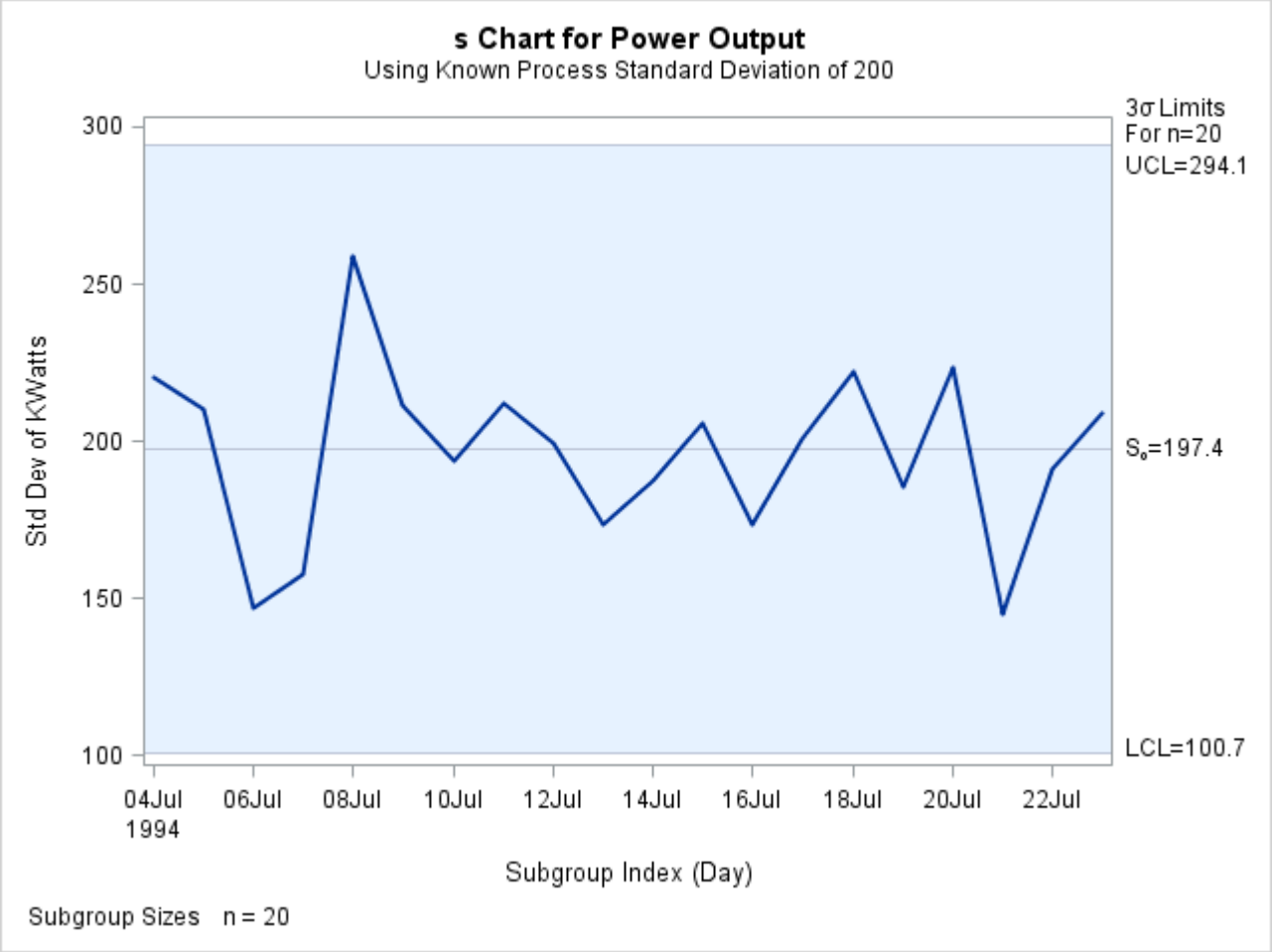
In some applications, a standard value σ_0 might be available for the process standard deviation σ. This example shows how you can specify σ_0 to compute the control limits.

Suppose that the amount of power needed to heat water in the heating process described in "Creating Standard Deviation Charts from Raw Data" on page 1768 has a known standard deviation of 200. The following statements specify this known value and create an *s* chart, shown in Output 19.29.1, for the power output measurements in the data set Turbine:

```
ods graphics on;
title 's Chart for Power Output';
title2 'Using Known Process Standard Deviation of 200';
proc shewhart data=Turbine;
   schart KWatts*Day / sigma0    = 200
                       ssymbol   = s0
                       odstitle  = title
                       odstitle2 = title2;
run;
```

The SIGMA0= option specifies σ_0, and the SSYMBOL= option specifies a label for the central line indicating that the central line is computed from σ_0. Because all the points lie within the limits, you can conclude that the variability of the process is stable.

Output 19.29.1 Reading in Standard Value for Process Standard Deviation



You can also specify σ_0 as the value of the variable _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data Plimits;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'KWatts';
   _subgrp_ = 'Day';
   _type_   = 'STDSIGMA';
   _limitn_ = 20;
   _stddev_ = 200;
run;

title 'Chart Using Known Process Standard Deviation';
proc shewhart data=Turbine limits=Plimits;
   schart KWatts*Day / ssymbol=s0;
run;
```

The resulting *s* chart (not shown here) is identical to the one shown in Output 19.29.1. For more information, see "LIMITS= Data Set" on page 1794.

Example 19.30: Computing Average Run Lengths for s Charts

NOTE: See *Computing Average Run Lengths for s Charts* in the SAS/QC Sample Library.

This example illustrates how you can compute the average run length of an *s* chart. The data used here are the power measurements in the data set Turbine, which is introduced in "Creating Standard Deviation Charts from Raw Data" on page 1768.

The in-control average run length of a Shewhart chart is ARL $= \frac{1}{p}$, where p is the probability that a single point exceeds its control limits. Because this probability is saved as the value of the variable _ALPHA_ in an OUTLIMITS= data set, you can compute ARL for an *s* chart as follows:

```
title 'Average In-Control Run Length';
proc shewhart data=Turbine;
   schart KWatts*Day / outlimits=Turblim nochart;

data ARLcomp;
   keep _var_ _sigmas_ _alpha_ arl;
   set Turblim;
   arl = 1 / _alpha_;
run;
```

The data set ARLcomp is listed in Output 19.30.1, which shows that the ARL is equal to 358.

Output 19.30.1 The Data Set ARLcomp

Average In-Control Run Length

VAR	_ALPHA_	_SIGMAS_	arl
KWatts	.002792725	3	358.073

To compute out-of-control average run lengths, define f as the slippage factor for the process standard deviation σ, where $f > 1$. In other words, the "shifted" standard deviation to be detected by the chart is $f\sigma$. The following statements compute the ARL as a function of f:

```
data ARLshift;
   keep f f_std p arl_f;
   set Turblim;
   df = _limitn_ - 1;
   do f = 1 to 1.5 by 0.05;
      f_std = f * _stddev_;
      low   = df * ( _lcls_ / f_std )**2;
      upp   = df * ( _ucls_ / f_std )**2;
      p     = probchi( low, df ) + 1 - probchi( upp, df );
      arl_f = 1 / p;
      output;
   end;
run;
```

The data set ARLshift is listed in Output 19.30.2. For example, on average, 53 samples are required to detect a ten percent increase in σ (a shifted standard deviation of approximately 219). The computations use the fact that $(n_i - 1)s_i^2/\sigma^2$ has a χ^2 distribution with $n_i - 1$ degrees of freedom, assuming that the measurements are normally distributed.

Output 19.30.2 The Data Set ARLshift

Average Run Length Analysis

f	f_std	p	arl_f
1.00	198.996	0.00279	358.073
1.05	208.945	0.00758	131.922
1.10	218.895	0.01875	53.322
1.15	228.845	0.03984	25.102
1.20	238.795	0.07388	13.535
1.25	248.745	0.12239	8.171
1.30	258.694	0.18475	5.413
1.35	268.644	0.25834	3.871
1.40	278.594	0.33923	2.948
1.45	288.544	0.42298	2.364
1.50	298.494	0.50546	1.978

UCHART Statement: SHEWHART Procedure

Overview: UCHART Statement

The UCHART statement creates *u* charts for the numbers of nonconformities (defects) per inspection unit in subgroup samples containing arbitrary numbers of units.

You can use options in the UCHART statement to

- specify the number of inspection units per subgroup
- compute control limits from the data based on a multiple of the standard error of the plotted values or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors

- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *u* charts with the UCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: UCHART Statement

This section introduces the UCHART statement with simple examples that illustrate commonly used options. Complete syntax for the UCHART statement is presented in the section "Syntax: UCHART Statement" on page 1812, and advanced examples are given in the section "Examples: UCHART Statement" on page 1831.

Creating u Charts from Defect Count Data

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

A textile company uses a *u* chart to monitor the number of defects per square meter of fabric. The fabric is spooled onto rolls as it is inspected for defects. Each piece of fabric is one meter wide and 30 meters in length. The following statements create a SAS data set named Fabric, which contains the defect counts for 20 rolls:

```
data Fabric;
   input Roll Defects @@;
   datalines;
 1 12    2 11    3  9    4 15
 5  7    6  6    7  5    8 10
 9  8   10  8   11 14   12  5
13  9   14 13   15  7   16  5
17  8   18 11   19  7   20 12
;
```

A partial listing of Fabric is shown in Figure 19.86.

Figure 19.86 The Data Set Fabric

Number of Fabric Defects

Roll	Defects
1	12
2	11
3	9
4	15
5	7

There is a single observation per roll. The variable Roll identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Defects contains the number of nonconformities (defect count) for each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *u* chart shown in Figure 19.87:

```
ods graphics off;
title 'u Chart for Fabric Defects';
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30;
run;
```

This example illustrates the basic form of the UCHART statement. After the keyword UCHART, you specify the *process* to analyze (in this case, Defects), followed by an asterisk and the *subgroup-variable* (Roll).

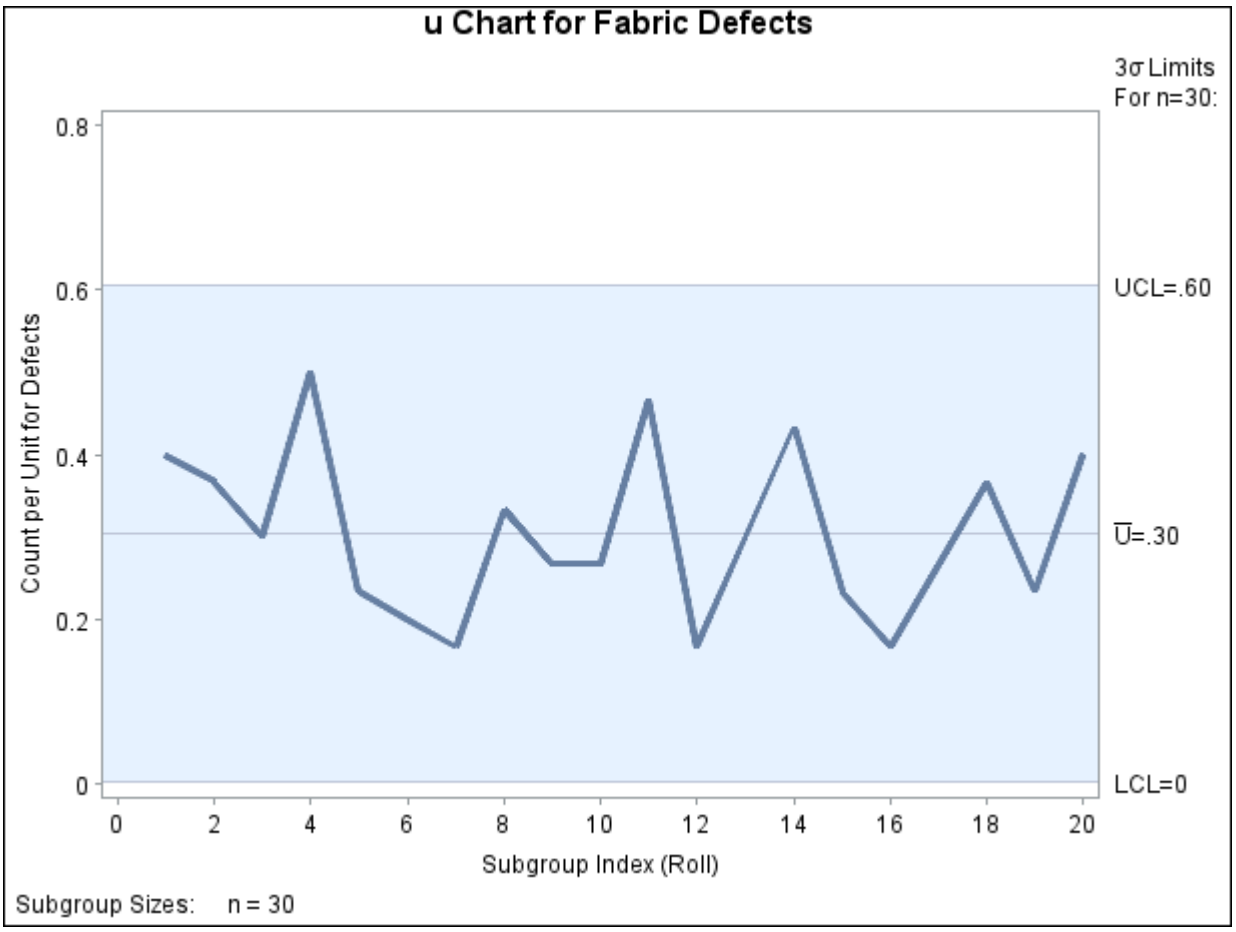
The SUBGROUPN= option specifies the number of inspection units in each subgroup sample and is required if the input data set is a DATA= data set. In this example, each square meter of fabric is an inspection unit, and each roll is a subgroup sample. The number of inspection units per subgroup can be thought of as the subgroup sample size.

You can use the SUBGROUPN= option to specify one of the following:

- a constant subgroup sample size (as in this example)
- an input variable name whose values contain the subgroup sample sizes (for an example, see "Saving Nonconformities per Unit" on page 1811)

Options such as SUBGROUPN= are specified after the slash (/) in the UCHART statement. A complete list of options is presented in the section "Syntax: UCHART Statement" on page 1812.

Figure 19.87 u Chart Example (Traditional Graphics)



The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Each point on the u chart represents the number of nonconformities per inspection unit for a particular subgroup. For instance, the value plotted for the first subgroup is $12/30 = 0.4$ (because there are 12 defects on the first roll and this roll contains 30 square meters of fabric). By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1824. Because none of the points exceed the 3σ limits, the u chart indicates that the fabric manufacturing process is in statistical control.

See "Constructing Charts for Nonconformities per Unit (u Charts)" on page 1823 for details concerning u charts. For more details on reading defect count data, see "DATA= Data Set" on page 1828.

Saving Control Limits

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a *u* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1807) or modify the limits with a DATA step program.

The following statements read defect counts from the data set Fabric (see "Creating u Charts from Defect Count Data" on page 1802) and save the control limits displayed in Figure 19.87 in a data set named Fablim:

```
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30
                         outlimits = Fablim
                         nochart;
run;
```

The SUBGROUPN= option specifies the number of inspection units in each subgroup sample. The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Fablim is listed in Figure 19.88.

Figure 19.88 The Data Set Fablim Containing Control Limit Information

Control Limits Data Set FABLIM

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
Defects	Roll	ESTIMATE	30	.002421390	3	.001671271	0.30333	0.60500

The data set Fablim contains one observation with the limits for *process* Defects. The variables _LCLU_ and _UCLU_ contain the lower and upper control limits, and the variable _U_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see "OUTLIMITS= Data Set" on page 1825.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30
                         outtable  = Fabtab
                         nochart;
run;
```

The data set Fabtab is listed in Figure 19.89.

Figure 19.89 The Data Set Fabtab

Number of Defects Per Square Meter and Control Limits

VAR	Roll	_SIGMAS_	_LIMITN_	_SUBN_	_LCLU_	_SUBU_	_U_	_UCLU_	_EXLIM_
Defects	1	3	30	30	.001671271	0.40000	0.30333	0.60500	
Defects	2	3	30	30	.001671271	0.36667	0.30333	0.60500	
Defects	3	3	30	30	.001671271	0.30000	0.30333	0.60500	
Defects	4	3	30	30	.001671271	0.50000	0.30333	0.60500	
Defects	5	3	30	30	.001671271	0.23333	0.30333	0.60500	
Defects	6	3	30	30	.001671271	0.20000	0.30333	0.60500	
Defects	7	3	30	30	.001671271	0.16667	0.30333	0.60500	
Defects	8	3	30	30	.001671271	0.33333	0.30333	0.60500	
Defects	9	3	30	30	.001671271	0.26667	0.30333	0.60500	
Defects	10	3	30	30	.001671271	0.26667	0.30333	0.60500	
Defects	11	3	30	30	.001671271	0.46667	0.30333	0.60500	
Defects	12	3	30	30	.001671271	0.16667	0.30333	0.60500	
Defects	13	3	30	30	.001671271	0.30000	0.30333	0.60500	
Defects	14	3	30	30	.001671271	0.43333	0.30333	0.60500	
Defects	15	3	30	30	.001671271	0.23333	0.30333	0.60500	
Defects	16	3	30	30	.001671271	0.16667	0.30333	0.60500	
Defects	17	3	30	30	.001671271	0.26667	0.30333	0.60500	
Defects	18	3	30	30	.001671271	0.36667	0.30333	0.60500	
Defects	19	3	30	30	.001671271	0.23333	0.30333	0.60500	
Defects	20	3	30	30	.001671271	0.40000	0.30333	0.60500	

This data set contains one observation for each subgroup sample. The variables _SUBU_ and _SUBN_ contain the number of nonconformities per unit in each subgroup and the number of inspection units per subgroup. The variables _LCLU_ and _UCLU_ contain the lower and upper control limits, and the variable _U_ contains the central line. The variables _VAR_ and Roll contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1827.

An OUTTABLE= data set can be read later as a TABLE= data set by the SHEWHART procedure. For example, the following statements read Fabtab and display a *u* chart (not shown here) identical to the chart in Figure 19.87:

```
title 'u Chart for Fabric Defects';
proc shewhart table=Fabtab;
   uchart Defects*Roll / subgroupn=30;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1830.

Reading Preestablished Control Limits

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

In the previous example, control limits were saved in a SAS data set named Fablim. This example shows how these limits can be applied to defect counts for an additional 20 rolls of fabric, which are provided in the following data set:

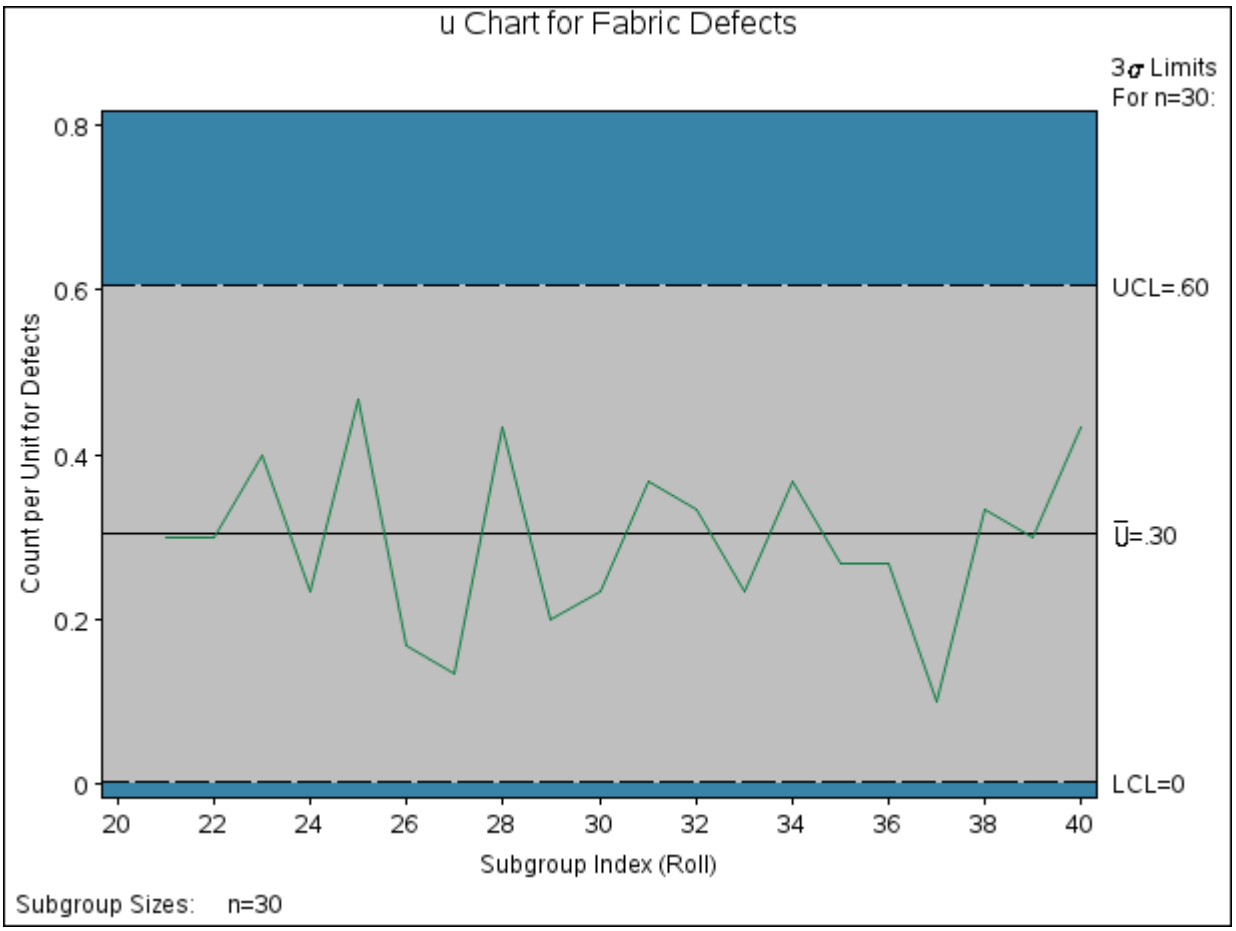
```
data Fabric2;
   input Roll Defects @@;
   datalines;
21  9    22  9    23 12    24  7    25 14
26  5    27  4    28 13    29  6    30  7
31 11    32 10    33  7    34 11    35  8
36  8    37  3    38 10    39  9    40 13
;
```

The following statements create a *u* chart for the second group of rolls using the control limits in Fablim:

```
options nogstyle;
goptions ftext='albany amt';
symbol color = vig h = .8;
title 'u Chart for Fabric Defects';
proc shewhart data=Fabric2 limits=Fablim;
   uchart Defects*Roll / subgroupn = 30
                         cframe    = steel
                         cinfill   = ligr
                         cconnect  = vig
                         coutfill  = yellow;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and UCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The chart is shown in Figure 19.90 and indicates that the process is in control.

Figure 19.90 A *u* Chart for Second Set of Fabric Rolls (Traditional Graphics with NOGSTYLE)



The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* Defects
- the value of _SUBGRP_ matches the *subgroup-variable* name Roll

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1829 for details concerning the variables that you must provide.

Creating u Charts from Nonconformities per Unit

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

In the previous example, the input data set provided the number of nonconformities for each subgroup sample. However, in some applications, as illustrated here, the data provide the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Because the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in

each box is divided by ten and then recorded. The following statements create a SAS data set named Shirts, which contains the average number of flaws per shirt for 25 boxes:

```
data Shirts;
   input Box AvgdefU @@;
   AvgdefN=10;
   datalines;
 1  0.4    2  0.7    3  0.5    4  1.0    5  0.3
 6  0.2    7  0.0    8  0.4    9  0.4   10  0.6
11  0.2   12  0.7   13  0.3   14  0.1   15  0.3
16  0.6   17  0.6   18  0.3   19  0.7   20  0.3
21  0.0   22  0.1   23  0.5   24  0.6   25  0.4
;
```

Note that this is the same data set used in "Getting Started: CCHART Statement" on page 1483 of "CCHART Statement: SHEWHART Procedure" on page 1482. A partial listing of Shirts is shown in Figure 19.91.

Figure 19.91 The Data Set Shirts

Average Number of Shirt Flaws

Box	AvgdefU	AvgdefN
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10

The data set Shirts contains three variables: the box number (Box), the average number of flaws per shirt (AvgdefU), and the number of shirts per box (AvgdefN). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup contains ten inspection units.

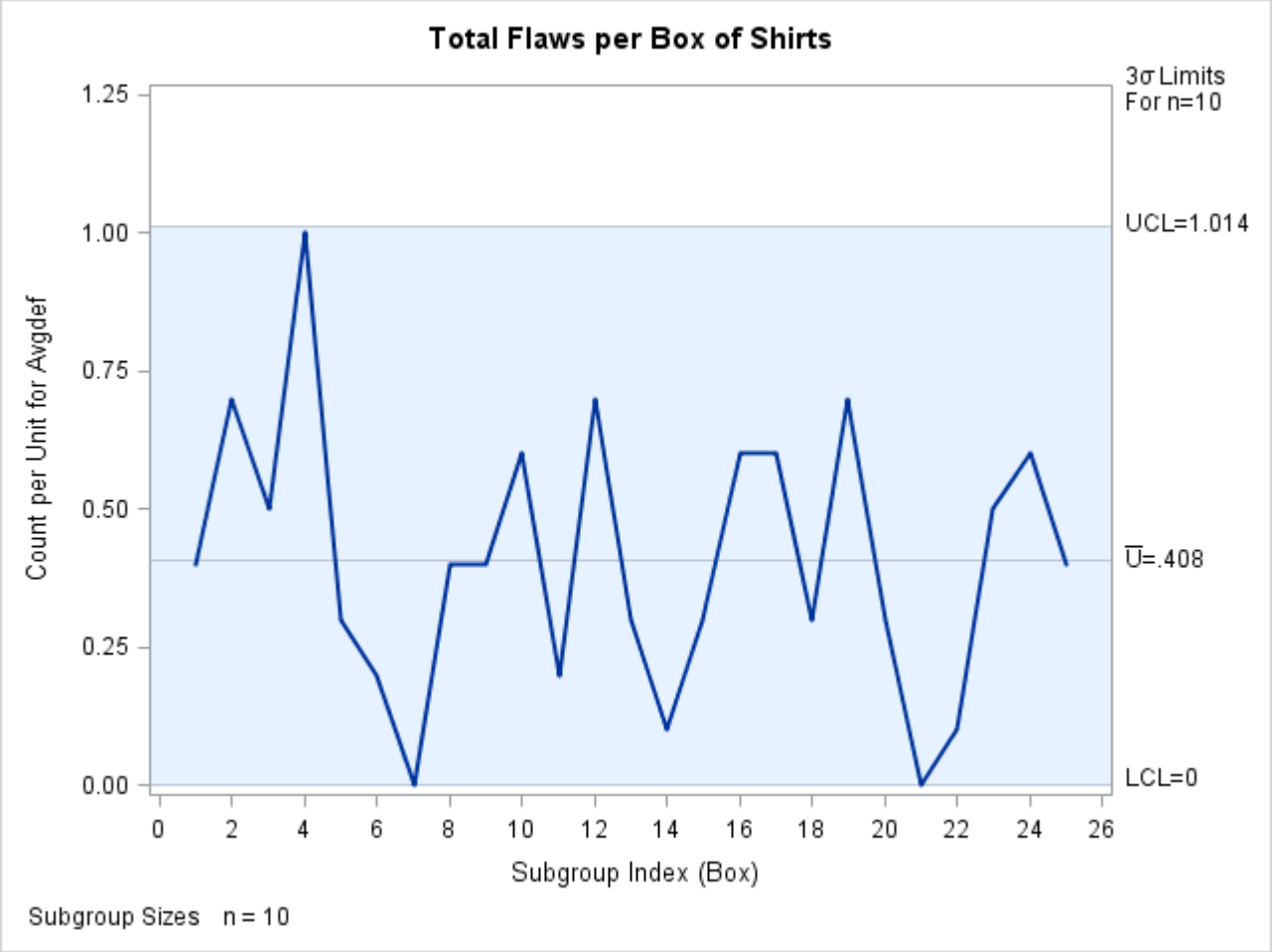
To create a *u* chart for the average number of flaws per shirt in each box, you can specify Shirts as a HISTORY= data set.

```
ods graphics on;
title 'Total Flaws per Box of Shirts';
proc shewhart history=Shirts;
   uchart Avgdef*Box / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *u* chart is created by using ODS Graphics instead of traditional graphics.

Note that Avgdef is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the SAS variables AvgdefU and AvgdefN. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name, which is referred to as the *process*. The name Box, specified after the asterisk, is the name of the *subgroup-variable*. The *u* chart is shown in Figure 19.92.

Figure 19.92 *u* Chart for Boxes of Shirts (ODS Graphics)



In general, a HISTORY= input data set used with the UCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the UCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables AvgdefU and AvgdefN, the data set Shirts contained the variables Shirtdef and Sizes. The following statements temporarily rename Shirtdef and Sizes to AvgdefU and AvgdefN:

```
proc shewhart
   history=Shirts (rename=(Shirtdef = AvgdefU
                           Sizes    = AvgdefN ));
   uchart Avgdef*Box;
run;
```

For more information, see "HISTORY= Data Set" on page 1829.

Saving Nonconformities per Unit

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

In this example, the UCHART statement is used to create a summary data set containing the number of nonconformities per unit. This data set can be read later by the SHEWHART procedure (as in the preceding example).

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named Shirts2, which contains the total defects per box for 20 boxes:

```
data Shirts2;
   input Box Flaws nShirts @@;
   datalines;
 1  3  10    2  8  10    3 15  25    4 20  25
 5  9  25    6  1  10    7  1  10    8 21  50
 9  3  10   10  7  10   11  1  10   12 21  25
13  9  25   14  3  25   15 12  50   16 18  50
17  7  10   18  4  10   19  8  10   20  4  10
;
```

A partial listing of Shirts2 is shown in Figure 19.93.

Figure 19.93 The Data Set Shirts2

Number of Shirt Flaws per Box

Box	AvgdefU	AvgdefN
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10

The variable Box contains the box number, the variable Flaws contains the number of flaws in each box, and the variable nShirts contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=Shirts2;
   uchart Flaws*Box / subgroupn  = nShirts
                      outhistory = Shirthist
                      nochart;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of Shirthist is shown in Figure 19.94.

Figure 19.94 The Data Set Shirthist

Average Defects Per Tee Shirt

Box	FlawsU	FlawsN
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25

There are three variables in the data set Shirthist.

- Box contains the subgroup index.
- FlawsU contains the numbers of nonconformities per inspection unit.
- FlawsN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* Flaws specified in the UCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1826.

Syntax: UCHART Statement

The basic syntax for the UCHART statement is as follows:

UCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

UCHART *processes* ∗ *subgroup-variable* <**(***block-variables***)**>
<**=***symbol-variable* | **=***'character'*> / <*options*> **;**

You can use any number of UCHART statements in the SHEWHART procedure. The components of the UCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities. For an example, see "Creating u Charts from Defect Count Data" on page 1802.
- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set. For an example, see "Creating u Charts from Nonconformities per Unit" on page 1808.
- If numbers of nonconformities per item, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1805.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *u* charts for Defects and Flaws:

```
proc shewhart data=Measures;
   uchart (Defects Flaws)*Sample / subgroupn=50;
run;
```

Note that when data are read from a DATA= data set with the UCHART statement, the SUBGROUPN= option (which specifies the number of inspection units per subgroup) is required.

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding UCHART statement, Sample is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities per unit.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a *u* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   uchart Defects*Sample='*' / subgroupn=100;
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the UCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.58 UCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	Requests probability limits at discrete values
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	Displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line

Table 19.58 *continued*

Option	Description
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
UCLLABEL=	Specifies label for upper control limit
USYMBOL=	Specifies label for central line
WLIMITS=	Specifies width for control limits and central line
Standard Value Options	
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U0=	Specifies known average number of nonconformities per unit
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on *u* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically

Table 19.58 *continued*

Option	Description
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive

Table 19.58 *continued*

Option	Description
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis
VFORMAT=	Specifies format for vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur

Table 19.58 *continued*

Option	Description
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart
ZEROSTD	Displays u chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= options
CVREF=	Specifies color for lines requested by VREF= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	Specifies labels for HREF= lines
HREFLABPOS=	Specifies position of HREFLABELS= labels
LHREF=	Specifies line type for HREF= lines
LVREF=	Specifies line type for VREF= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	Specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option

Table 19.58 *continued*

Option	Description
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features chart
DESCRIPTION=	Specifies description of *u* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of *u* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers

Table 19.58 *continued*

Option	Description
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
SUBGROUPN	Specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits

Table 19.58 *continued*

Option	Description
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles

Table 19.58 *continued*

Option	Description
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for overlay line segments
COVERLAY=	Specifies colors for overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on chart
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with overlay points
OVERLAYID=	Specifies labels for overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for overlays
OVERLAYSYMHT=	Specifies symbol heights for overlays
WOVERLAY=	Specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data

Table 19.58 *continued*

Option	Description
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: UCHART Statement

The following sections provide details that are specific to the UCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Nonconformities per Unit (u Charts)

The following notation is used in this section:

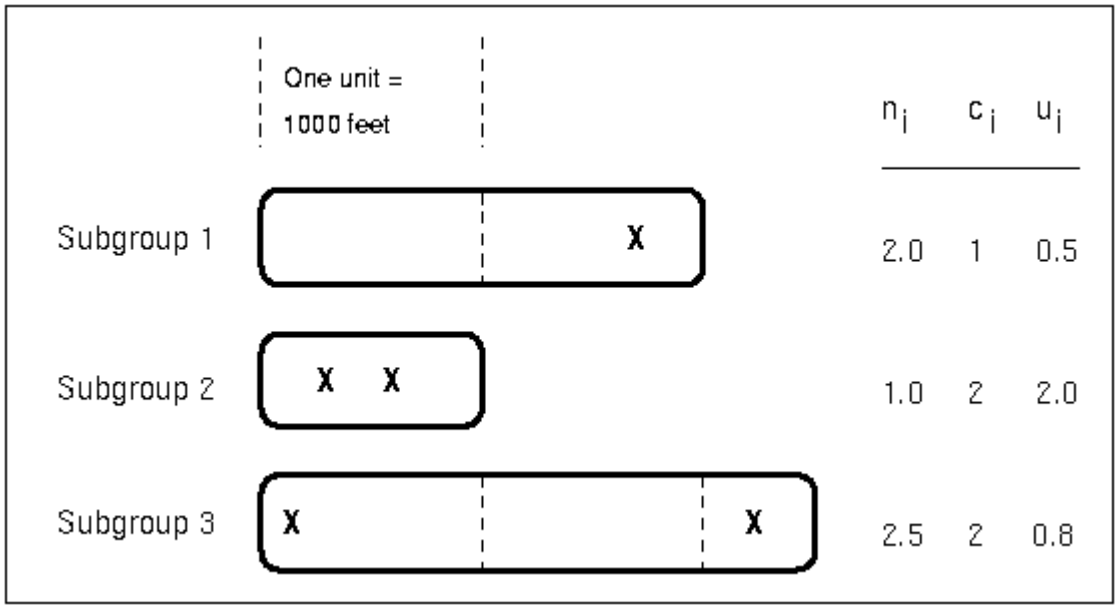
u	Expected number of nonconformities per unit produced by process
u_i	Number of nonconformities per unit in the ith subgroup. In general, $u_i = c_i/n_i$.
c_i	Total number of nonconformities in the ith subgroup
n_i	Number of inspection units in the ith subgroup
$\bar{u}$	Average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1u_1 + \cdots + n_Nu_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	Number of subgroups
χ^2_ν	Has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a u chart indicates the number of nonconformities per unit (u_i) in a subgroup. For example, Figure 19.95 displays three sections of pipeline that are inspected for defective welds (indicated by an X).

Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long sections. The number of units in the ith subgroup is denoted by n_i, which is the subgroup sample size.

Figure 19.95 Terminology for c Charts and u Charts



The *number of nonconformities* in the ith subgroup is denoted by c_i. The *number of nonconformities per unit* in the ith subgroup is denoted by $u_i = c_i/n_i$. In Figure 19.95, the number of defective welds per unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A u chart plots the quantity u_i for the ith subgroup. A c chart plots the quantity c_i for the ith subgroup (see "CCHART Statement: SHEWHART Procedure" on page 1482). An advantage of a u chart is that the value of the central line at the ith subgroup does not depend on n_i. This is not the case for a c chart, and consequently, a u chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a u chart, the central line indicates an estimate of u, which is computed as $\bar{u}$ by default. If you specify a known value (u_0) for u, the central line indicates the value of u_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of u_i above and below the central line. The default limits are computed with k = 3 (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that u_i exceeds the limits

The lower and upper control limits, LCLU and UCLU, respectively, are given by

$$\begin{aligned} \text{LCLU} &= \max\left(\bar{u} - k\sqrt{\bar{u}/n_i}\ , 0\right) \\ \text{UCLU} &= \bar{u} + k\sqrt{\bar{u}/n_i} \end{aligned}$$

The limits vary with n_i.

The upper probability limit UCLU for u_i can be determined using the fact that

$$\begin{aligned} P\{u_i > \text{UCLU}\} &= 1 - P\{u_i \leq \text{UCLU}\} \\ &= 1 - P\{c_i \leq n_i \text{UCLU}\} \\ &= 1 - P\{\chi^2_{2(n_i \times \text{UCLU}+1)} \geq 2n_i\bar{u}\} \end{aligned}$$

The limit UCLU is then calculated by setting

$$1 - P\{\chi^2_{2(n_i \times \text{UCLU}+1)} \geq 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLU.

Likewise, the lower probability limit LCLU for u_i can be determined using the fact that

$$\begin{aligned} P\{u_i < \text{LCLU}\} &= P\{u_i \leq \text{LCLU} - 1\} \\ &= P\{c_i \leq n_i \text{LCLU} - 1\} \\ &= P\{\chi^2_{2(n_i \times (\text{LCLU}-1)+1)} > 2n_i\bar{u}\} \\ &= P\{\chi^2_{2(n_i \text{LCLU}} > 2n_i\bar{u}\} \end{aligned}$$

The limit LCLU is then calculated by setting

$$P\{\chi^2_{2(n_i \text{LCLU})} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLU. For more information, refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that c_i has a Poisson distribution. Note that the probability limits vary with n_i and are asymmetric around the central line. If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.60 lists the variables that can be saved.

Table 19.60 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLU	Lower control limit for number of nonconformities per unit
LIMITN	Sample size associated with the control limits
SIGMAS	Multiple (k) of standard error of u_i
SUBGRP	*Subgroup-variable* specified in the UCHART statement
TYPE	Type (estimate or standard value) of _U_
_U_	Value of central line of u chart ($\bar{u}$ or u_0)
UCLU	Upper control limit for number of nonconformities per unit
VAR	*Process* specified in the UCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value 'V' is assigned to the variables _LCLU_, _UCLU_, and _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of u_i, the value of _ALPHA_ is computed as $P\{u_i < \text{\_LCLU\_}\} + P\{u_i > \text{\_UCLU\_}\}$, provided that n_i is a constant. Otherwise, _ALPHA_ is assigned the special missing value 'V'.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLU\_} - \text{\_U\_})/\sqrt{\text{\_U\_}/\text{\_LIMITN\_}}$, provided that n_i is a constant. Otherwise, _SIGMAS_ is assigned the special missing value V.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the UCHART statement. For an example, see "Saving Control Limits" on page 1805.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup number of nonconformities per unit variable named by *process* suffixed with U
- a subgroup sample size variable named by *process* suffixed with N

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the UCHART statement. For example, consider the following statements:

```
proc shewhart data=Fabric;
   uchart (Flaws nDefects)*lot / outhistory=Summary
                                 subgroupn = 10;
run;
```

The data set Summary contains the variables Lot, FlawsU, FlawsN, nDefectsU, and nDefectsN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Nonconformities per Unit" on page 1811. Note that an OUTHISTORY= data set created with the UCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.61 lists the variables that are saved.

Table 19.61 OUTTABLE= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on u chart
LCLU	Lower control limit for number of nonconformities per unit
LIMITN	Nominal sample size associated with the control limits
SIGMAS	Multiple (k) of the standard error associated with the control limits
Subgroup	Values of the subgroup variable
SUBU	Subgroup number of nonconformities per unit
SUBN	Subgroup sample size
TESTS	Tests for special causes signaled on u chart
_U_	Average number of nonconformities per unit
UCLU	Upper control limit for number of nonconformities per unit
VAR	*Process* specified in the UCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*

- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for *u* charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1805.

Input Data Sets

DATA= Data Set

You can read defect counts for subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the UCHART statement must be a SAS variable in the data set. This variable provides the defect count (number of nonconformities) for subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, specified in the UCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set should contain one observation per subgroup. When you use a DATA= data set with the UCHART statement, the SUBGROUPN= option (which specifies the number of inspection units per subgroup) is required. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating u Charts from Defect Count Data" on page 1802.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   uchart Defects*Lot / subgroupn = 10;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLU_, _U_, and _UCLU_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see "Control Limits" on page 1824)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1807.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures.

A HISTORY= data set used with the UCHART statement must contain the following variables:

- *subgroup-variable*
- subgroup number of nonconformities per unit variable for each *process*
- subgroup sample size variable (number of units per subgroup) for each *process*

The names of the variables containing the number of nonconformities per unit and subgroup sample sizes must be the *process* name concatenated with the special suffix characters *U* and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=Summary;
   uchart (Flaws nDefects)*Lot;
run;
```

The data set Summary must include the variables Lot, FlawsU, FlawsN, nDefectsU, and nDefectsN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating u Charts from Nonconformities per Unit" on page 1808.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.62 lists the variables required in a TABLE= data set used with the UCHART statement.

Table 19.62 Variables Required in a TABLE= Data Set

Variable	Description
LCLU	Lower control limit for nonconformities per unit
LIMITN	Nominal sample size associated with the control limits
Subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBU	Subgroup number of nonconformities per unit
_U_	Average number of nonconformities per unit
UCLU	Upper control limit for nonconformities per unit

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1805.

Examples: UCHART Statement

This section provides advanced examples of the UCHART statement.

Example 19.31: Applying Tests for Special Causes

NOTE: See *u Chart-Applying Tests for Special Causes* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make *u* charts more sensitive to special causes of variation.

A textile company inspects rolls of fabric for defects. The rolls are one meter wide and 30 meters long. The following statements create a SAS data set named Fabric3, which contains the number of fabric defects for 20 rolls of fabric:

```
data Fabric3;
   input Roll Defects @@;
   datalines;
 1   6    2   9    3  14    4  17
 5   3    6   8    7   9    8   2
 9  14   10   1   11   3   12   5
13   6   14   9   15  10   16  12
17  11   18   4   19   9   20   4
;
```

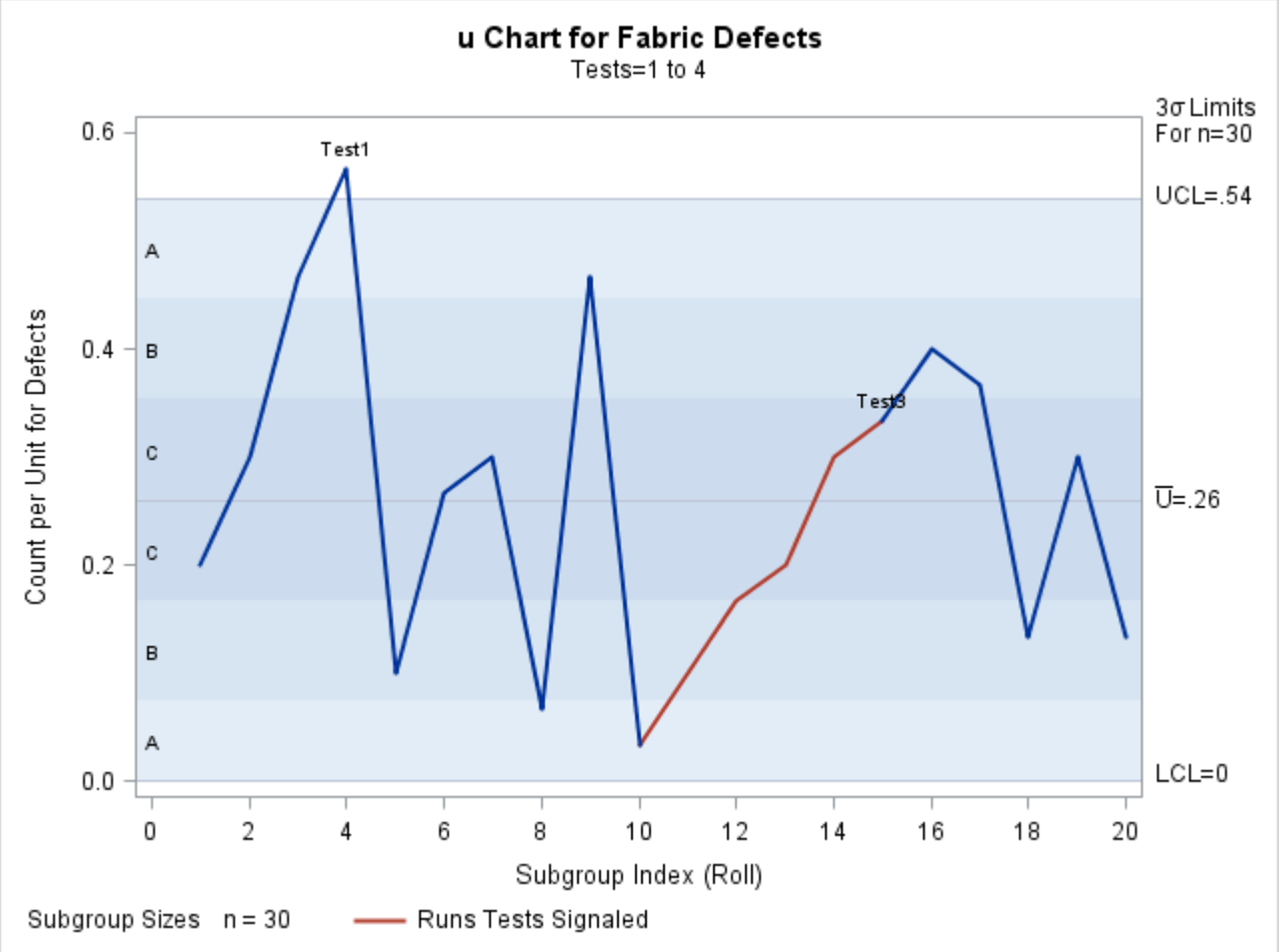
The following statements create a *u* chart and tabulate the information on the chart. The chart and tables are shown in Output 19.31.1 and Output 19.31.2.

```
ods graphics on;
title1 'u Chart for Fabric Defects';
title2 'Tests=1 to 4';
proc shewhart data=Fabric3;
   uchart Defects*Roll / subgroupn = 30
                         tests     = 1 to 4
                         odstitle  = title
                         odstitle2 = title2
                         tabletests
                         zonelabels;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. Only Tests 1, 2, 3, and 4 are recommended for u charts. The ZONELABELS option requests the zone lines, which are used to define the tests, and displays labels for the zones. The TABLETESTS option requests a table of the values of u_i and the control limits, together with a column indicating the subgroups at which the tests are positive.

Output 19.31.1 and Output 19.31.2 indicate that Test 1 is positive for Roll 4 and Test 3 is positive at Roll 15.

Output 19.31.1 Tests for Special Causes Displayed on u Chart



Output 19.31.2 Tabular Form of u Chart

u Chart for Fabric Defects
Tests=1 to 4

The SHEWHART Procedure

u Chart Summary for Defects

Roll	Subgroup Sample Size	3 Sigma Limits with n=30 for Count per Unit: Lower Limit	Subgroup Count per Unit	Upper Limit	Special Tests Signaled
1	30.0000	0	0.20000000	0.53928480	
2	30.0000	0	0.30000000	0.53928480	
3	30.0000	0	0.46666667	0.53928480	
4	30.0000	0	0.56666667	0.53928480	1
5	30.0000	0	0.10000000	0.53928480	
6	30.0000	0	0.26666667	0.53928480	
7	30.0000	0	0.30000000	0.53928480	
8	30.0000	0	0.06666667	0.53928480	
9	30.0000	0	0.46666667	0.53928480	
10	30.0000	0	0.03333333	0.53928480	
11	30.0000	0	0.10000000	0.53928480	
12	30.0000	0	0.16666667	0.53928480	
13	30.0000	0	0.20000000	0.53928480	
14	30.0000	0	0.30000000	0.53928480	
15	30.0000	0	0.33333333	0.53928480	3
16	30.0000	0	0.40000000	0.53928480	
17	30.0000	0	0.36666667	0.53928480	
18	30.0000	0	0.13333333	0.53928480	
19	30.0000	0	0.30000000	0.53928480	
20	30.0000	0	0.13333333	0.53928480	

Example 19.32: Specifying a Known Expected Number of Nonconformities

NOTE: See *u Chart-Known Expected Number of Nonconformities* in the SAS/QC Sample Library.

This example illustrates how you can create a u chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

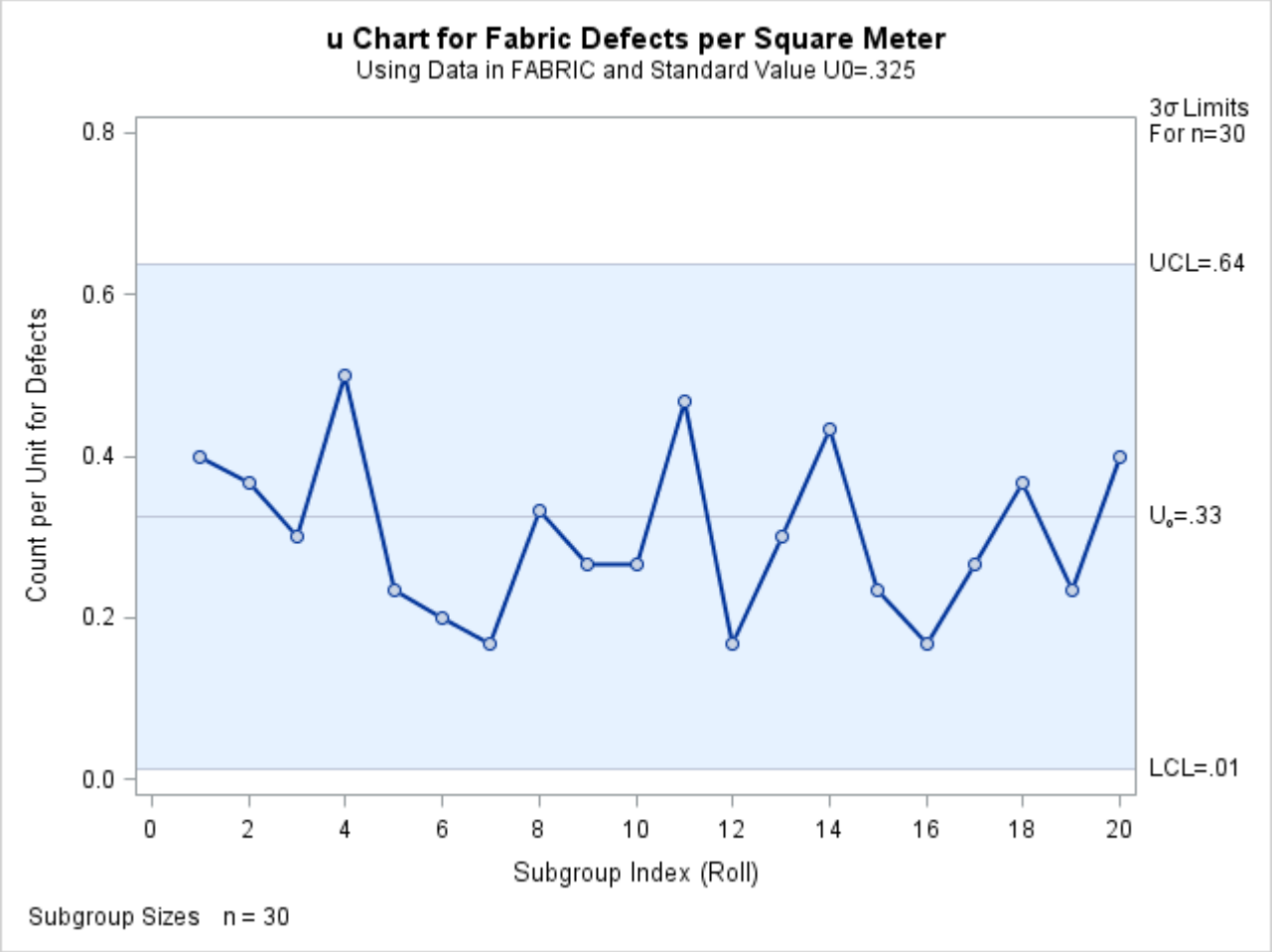
A u chart is used to monitor the number of defects per square meter of fabric. The defect counts are provided as values of the variable Defects in the data set Fabric (see "Creating u Charts from Defect Count Data" on page 1802). Based on previous testing, it is known that $u_0 = 0.325$. The following statements create a u chart with control limits derived from this value:

```
ods graphics on;
title  'u Chart for Fabric Defects per Square Meter';
title2 'Using Data in FABRIC and Standard Value U0=.325';
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30
                         u0        = 0.325
```

```
                                usymbol   = u0
                                odstitle  = title
                                odstitle2 = title2
                                markers;
run;
```

The chart is shown in Output 19.32.1. The U0= option specifies u_0, and the USYMBOL= option requests a label for the central line indicating that the line represents a standard value.

Output 19.32.1 A u Chart with Standard Value u_0



Because all the points lie within the control limits, the process is in statistical control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data Tlimits;
   length _subgrp_ _var_ _type_ $8;
   _u_      = .325;
   _limitn_ = 30;
   _type_   = 'STANDARD';
   _subgrp_ = 'Roll';
   _var_    = 'Defects';
```

```
proc shewhart data=Fabric limits=Tlimits;
   uchart Defects*Roll / subgroupn=30
                         usymbol  =u0;
run;
```

The chart produced by these statements is identical to the one in Output 19.32.1. For further details, see "LIMITS= Data Set" on page 1829.

Example 19.33: Creating u Charts for Varying Numbers of Units

NOTE: See *u Charts-Varying Number of Inspection Units* in the SAS/QC Sample Library.

In the fabric manufacturing process described in "Creating u Charts from Defect Count Data" on page 1802, each roll of fabric is 30 meters long, and an inspection unit is defined as one square meter. Thus, there are 30 inspection units in each subgroup sample. Suppose now that the length of each piece of fabric varies. The following statements create a SAS data set (Fabrics2) that contains the number of fabric defects and size (in square meters) of 25 pieces of fabric:

```
data Fabrics2;
   input Roll Defects Squaremeters @@;
   datalines;
 1  7 30.0   2 11 27.6   3 15 30.4   4  6 34.8   5 11 26.0
 6 15 28.6   7  5 28.0   8 10 30.2   9  8 28.2  10  3 31.4
11  3 30.3  12 14 27.8  13  3 27.0  14  9 30.0  15  7 32.1
16  6 34.8  17  7 26.5  18  5 30.0  19 14 31.3  20 13 31.6
21 11 29.4  22  6 28.6  23  6 27.5  24  9 32.6  25 11 31.7
;
```

A partial listing of Fabrics2 is shown in Output 19.33.1.

Output 19.33.1 The Data Set Fabrics2

Number of Fabric Defects

Roll	Defects	Squaremeters
1	7	30.0
2	11	27.6
3	15	30.4
4	6	34.8
5	11	26.0

The variable Roll contains the roll number, the variable Defects contains the number of defects in each piece of fabric, and the variable Squaremeters contains the size of each piece.

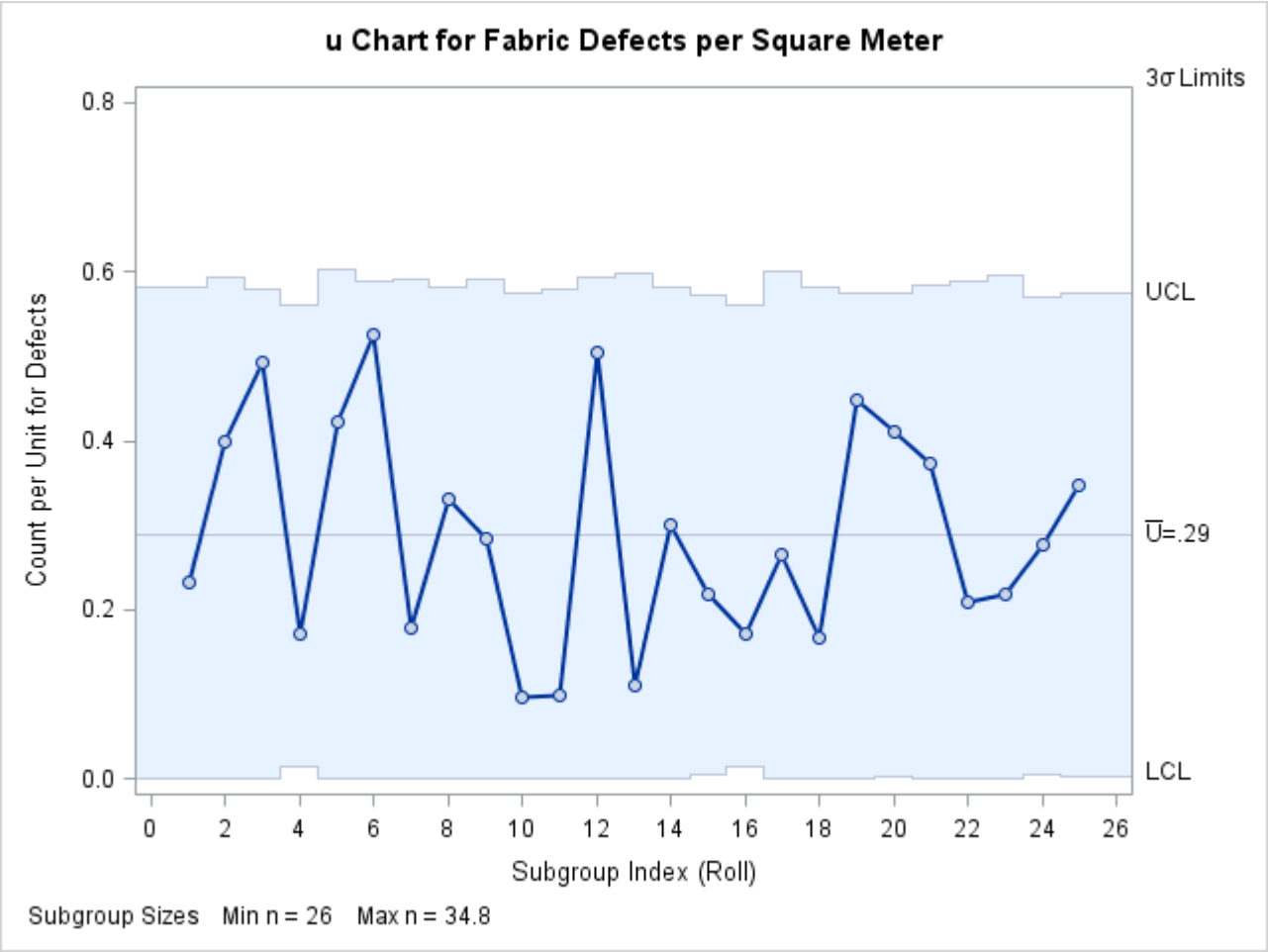
The following statements request a *u* chart for the number of defects per square meter:

```
ods graphics on;
title 'u Chart for Fabric Defects per Square Meter';
proc shewhart data=Fabrics2;
   uchart Defects*Roll / subgroupn = Squaremeters
                         outlimits = Fablimits
                         odstitle  = title
```

```
                              markers;
run;
```

The *u* chart is shown in Output 19.33.2, and the data set Fablimits is listed in Output 19.33.3.

Output 19.33.2 A *u* Chart with Varying Number of Units per Subgroup



Note that the control limits vary with the number of units per subgroup (subgroup sample size). The legend in the lower left corner indicates the minimum and maximum subgroup sample sizes.

Output 19.33.3 The Control Limits Data Set Fablimits

Control Limits for Fabric Defects

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
Defects	Roll	ESTIMATE	V	V	3	V	0.28805	V

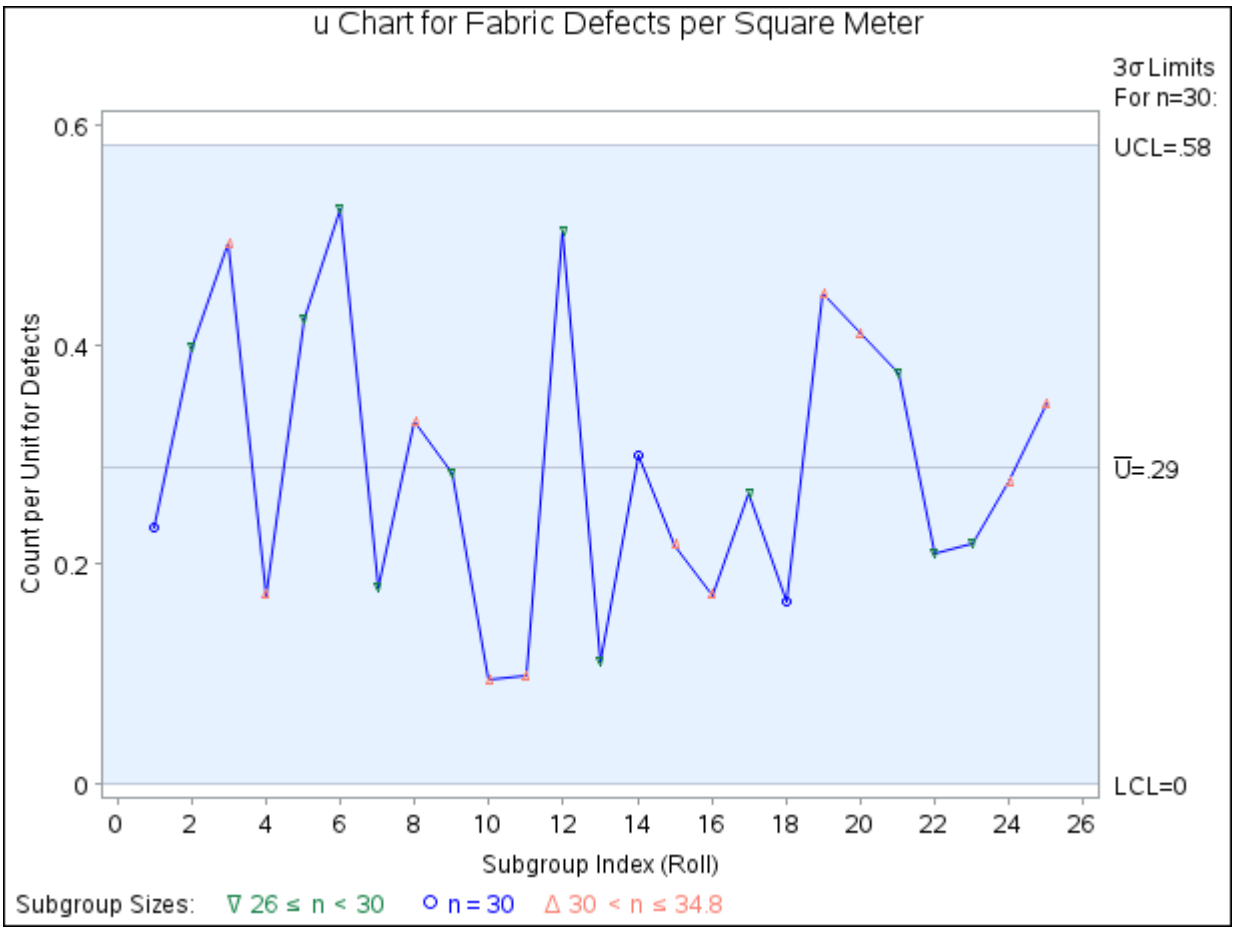
Output 19.33.3 shows that the variables _LIMITN_, _ALPHA_, _LCLU_, and _UCLU_ have the special missing value *V*, indicating that these variables vary with the sample size.

The following statements request a *u* chart with a fixed sample size of 30.0 for the control limits. In other words, the control limits are computed as if each piece of fabric were 30 meters long.

```
ods graphics off;
symbol1 c=blue v=circle;
symbol2 c=vig;
symbol3 c=salmon;
title 'u Chart for Fabric Defects per Square Meter';
proc shewhart data=Fabrics2;
   uchart Defects*Roll / subgroupn = Squaremeters
                         outlimits = Fablimits2
                         limitn    = 30
                         alln
                         nmarkers;
run;
```

The ALLN option specifies that points are to be displayed for all subgroups, regardless of their sample size. By default, when you specify the LIMITN= option, only points for subgroups whose sample size matches the LIMITN= value are displayed. The NMARKERS option requests special symbols that identify points for which the subgroup sample size differs from the nominal sample size of 30. The chart is shown in Output 19.33.4.

Output 19.33.4 Control Limits Based on Fixed Subgroup Sample Size



In Output 19.33.4, no points lie outside the control limits, indicating that the process is in control. However, you should be careful when interpreting charts that use a nominal sample size, because the fixed control

limits based on this value are only an approximation. Output 19.33.5 lists the data set Fablimits2, which contains the fixed control limits displayed in Output 19.33.4.

Output 19.33.5 The Fixed Control Limits Data Set Fablimits2

Fixed Control Limits for Fabric Defects

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
Defects	Roll	ESTIMATE	30	.002444992	3	0	0.28805	0.58201

XCHART Statement: SHEWHART Procedure

Overview: XCHART Statement

The XCHART statement creates an $\bar{X}$ chart for subgroup means, which is used to analyze the central tendency of a process.

You can use options in the XCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means or as probability limits
- tabulate subgroup sample sizes, subgroup means, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and subgroup means in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 2100)
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable

- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing $\bar{X}$ charts with the XCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: When working with variables data, you should analyze the variability of the process as well as its central tendency. You can use the XRCHART statement or the XSCHART statement in the SHEWHART procedure for this purpose.

Getting Started: XCHART Statement

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the section "Syntax: XCHART Statement" on page 1851, and advanced examples are given in the section "Examples: XCHART Statement" on page 1873.

Creating Charts for Means from Raw Data

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

Subgroup samples of five parts are taken from the manufacturing process at regular intervals, and the width of a critical gap in each part is measured in millimeters. The following statements create a SAS data set named Partgaps, which contains the gap width measurements for 21 samples:

```
data Partgaps;
   input Sample @;
   do i=1 to 5;
      input Partgap @;
      output;
   end;
   drop i;
   label Partgap='Gap Width'
         Sample ='Sample Index';
   datalines;
 1 255 270 268 290 267
 2 260 240 265 262 263
 3 238 236 260 250 256
```

```
 4 260 242 281 254 263
 5 268 260 279 289 269
 6 270 249 265 253 263
 7 280 260 256 256 243
 8 229 266 250 243 252
 9 250 270 245 273 262
10 248 258 247 266 256
11 280 251 252 270 287
12 245 253 243 279 245
13 268 260 289 275 273
14 264 286 275 271 279
15 271 257 263 247 247
16 291 250 273 265 266
17 228 253 240 260 264
18 270 260 269 245 276
19 259 257 246 271 257
20 252 244 230 266 248
21 254 251 239 233 263
;
```

A partial listing of Partgaps is shown in Figure 19.96.

Figure 19.96 Partial Listing of the Data Set Partgaps

The Data Set PARTGAPS

Sample	Partgap
1	255
1	270
1	268
1	290
1	267
2	260
2	240
2	265
2	262
2	263

The data set Partgaps is said to be in "strung-out" form, because each observation contains the sample number and gap width measurement for a single part. The first five observations contain the gap widths for the first sample, the second five observations contain the gap widths for the second sample, and so on. Because the variable Sample classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Partgap contains the gap width measurements and is referred to as the *process variable* (or *process* for short).

The within-subgroup variability of the gap widths is known to be stable. You can use an $\bar{X}$ chart to determine whether their mean level is in control. The following statements create the $\bar{X}$ chart shown in Figure 19.97:

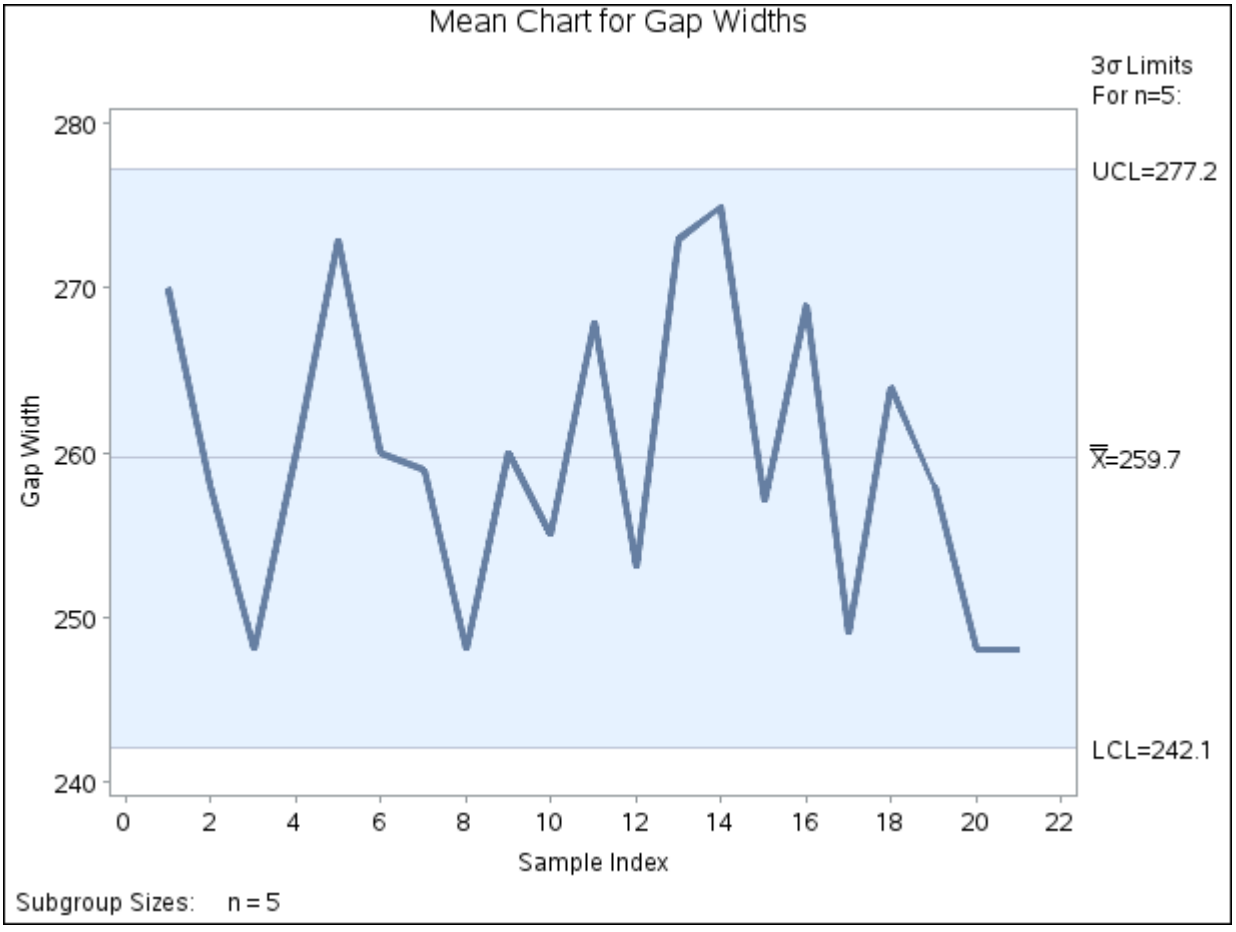
```
ods graphics off;
title 'Mean Chart for Gap Widths';
proc shewhart data=Partgaps;
   xchart Partgap*Sample;
run;
```

This example illustrates the basic form of the XCHART statement. After the keyword XCHART, you specify the *process* to analyze (in this case, Partgap) followed by an asterisk and the *subgroup-variable* (Sample). The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Each point on the $\bar{X}$ chart represents the average (mean) of the measurements for a particular sample. For instance, the mean plotted for the first sample is

$$\frac{255 + 270 + 268 + 290 + 267}{5} = 270$$

Figure 19.97 $\bar{X}$ Chart for Gap Width Data (Traditional Graphics)



Because all of the subgroup means lie within the control limits, it can be concluded that the mean level of the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.65. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1849.

For computational details, see "Constructing Charts for Means" on page 1863. For details on reading raw measurements, see "DATA= Data Set" on page 1868.

Creating Charts for Means from Subgroup Summary Data

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create $\bar{X}$ charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the XCHART statement with data of this type.

The following data set (Parts) provides the data from the preceding example in summarized form:

```
data Parts;
   input Sample PartgapX PartgapR;
   PartgapN=5;
   label PartgapX='Mean of Gap Width'
         Sample  ='Sample Index';
   datalines;
 1  270  35
 2  258  25
 3  248  24
 4  260  39
 5  273  29
 6  260  21
 7  259  37
 8  248  37
 9  260  28
10  255  19
11  268  36
12  253  36
13  273  29
14  275  22
15  257  24
16  269  41
17  249  36
18  264  31
19  258  25
20  248  36
21  248  30
;
```

A partial listing of Parts is shown in Figure 19.98. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Sample). The variable PartgapX contains the subgroup means, the variable PartgapR contains the subgroup ranges, and the variable PartgapN contains the subgroup sample sizes (these are all five).

Figure 19.98 The Summary Data Set Parts

The Data Set PARTS

Sample	PartgapX	PartgapR	PartgapN
1	270	35	5
2	258	25	5
3	248	24	5
4	260	39	5
5	273	29	5

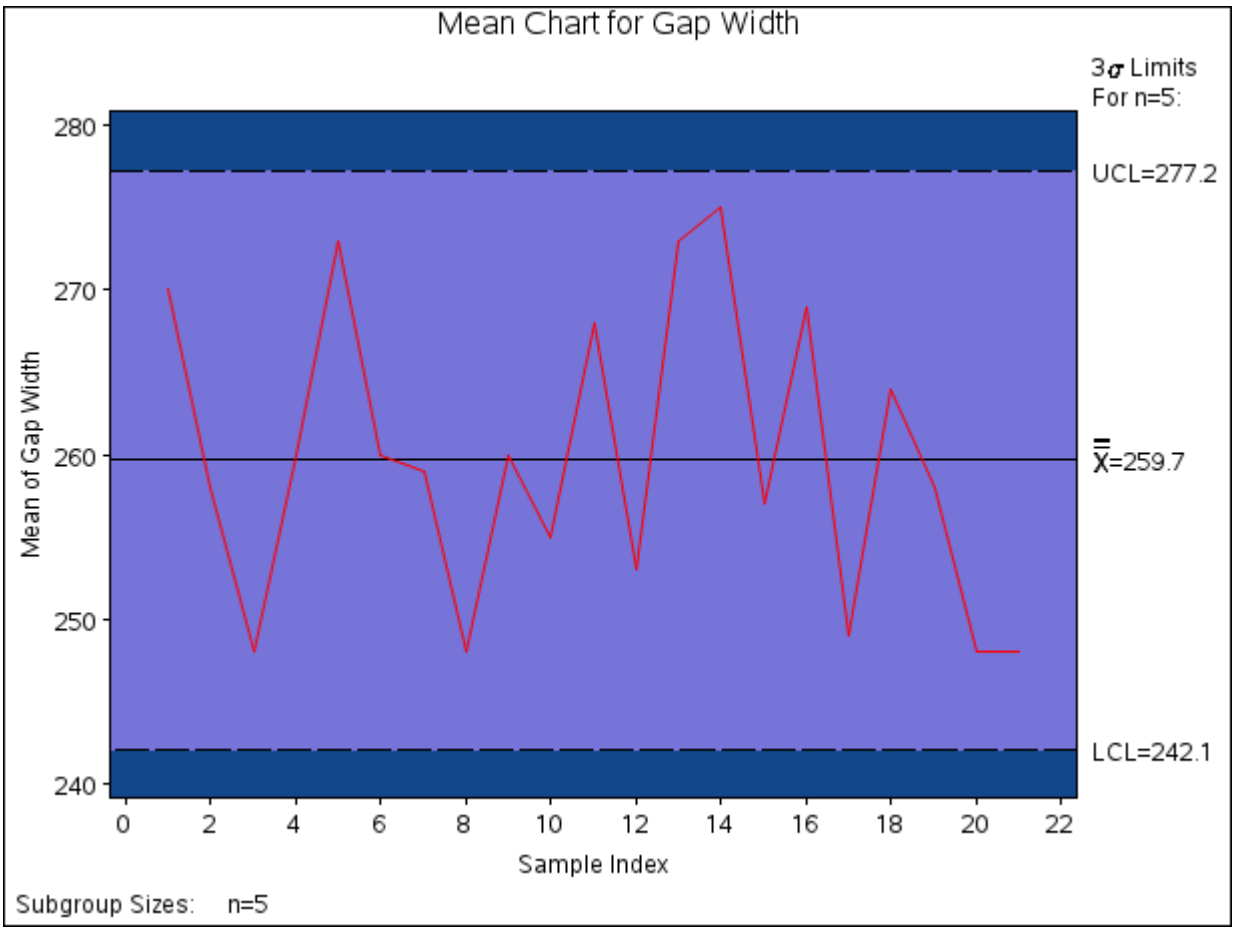
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
title 'Mean Chart for Gap Width';
proc shewhart history=Parts;
   xchart Partgap*Sample / cframe   = vigb
                           cinfill  = vlib
                           cconnect = red;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the XCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting $\bar{X}$ chart is shown in Figure 19.99.

Note that Partgap is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables PartgapX, PartgapR, and PartgapN. The suffix characters *X*, *R*, and *N* indicate *mean*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (Partgap), which is referred to as the *process*. The name Sample specified after the asterisk is the name of the *subgroup-variable*.

Figure 19.99 $\bar{X}$ Chart from the Summary Data Set Parts (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the XCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- either a subgroup range variable or a subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, range (or standard deviation), and sample size variables must begin with the *process* name specified in the XCHART statement and end with the special suffix characters *X*, *R* (or *S*), and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1885).

If you specify the STDDEVIATIONS option in the XCHART statement, the HISTORY= data set must contain a subgroup standard deviation variable; otherwise, the HISTORY= data set must contain a subgroup range variable. The STDDEVIATIONS option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations rather than subgroup ranges. For example, in the following statements, the data set Parts2 must contain a subgroup standard deviation variable named PartgapS:

```
title 'Mean Chart for Gap Width';
proc shewhart history=Parts2;
   xchart Partgap*Sample='*' / stddeviations;
run;
```

Options such as STDDEVIATIONS are specified after the slash (/) in the XCHART statement. A complete list of options is presented in the section "Syntax: XCHART Statement" on page 1851.

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1869.

Saving Summary Statistics

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

In this example, the XCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Partgaps and create a summary data set named Gaphist:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outhistory = Gaphist
                           nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 19.97.

Figure 19.100 contains a partial listing of Gaphist.

Figure 19.100 The Summary Data Set Gaphist

Summary Data Set for Gap Widths

Sample	PartgapX	PartgapR	PartgapN
1	270	35	5
2	258	25	5
3	248	24	5
4	260	39	5
5	273	29	5

There are four variables in the data set Gaphist.

- Sample contains the subgroup index.
- PartgapX contains the subgroup means.
- PartgapR contains the subgroup ranges.
- PartgapN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* Partgap specified in the XCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the STDDEVIATIONS option, the OUTHISTORY= data set includes a subgroup standard deviation variable rather than a subgroup range variable, as demonstrated by the following statements:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outhistory = Gaphist2
                           stddeviations
                           nochart;
run;
```

Figure 19.101 contains a partial listing of Gaphist2.

Figure 19.101 The Summary Data Set Gaphist2

Summary Data Set with Subgroup Standard Deviations

Sample	PartgapX	PartgapS	PartgapN
1	270	12.6293	5
2	258	10.2225	5
3	248	10.6771	5
4	260	14.2302	5
5	273	11.2027	5

The variable PartgapS, which contains the subgroup standard deviations, is named by adding the suffix character *S* to the *process* Partgap.

For more information, see "OUTHISTORY= Data Set" on page 1866.

Saving Control Limits

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for an $\bar{X}$ chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1849) or modify the limits with a DATA step program.

The following statements read measurements from the data set Partgaps (see "Creating Charts for Means from Raw Data" on page 1839) and save the control limits displayed in Figure 19.97 in a data set named Gaplim:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outlimits = Gaplim
                           nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Gaplim is listed in Figure 19.102.

Figure 19.102 The Data Set Gaplim Containing Control Limit Information

Control Limits for Gap Width Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Partgap	Sample	ESTIMATE	5	.002699796	3	242.087	259.667	277.246

LCLR	_R_	_UCLR_	_STDDEV_
0	30.4762	64.4419	13.1028

The data set Gaplim contains one observation with the limits for *process* Partgap. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLR_, _R_, and _UCLR_ are not used to create $\bar{X}$ charts, but they are included so the data set Gaplim can be used to create an R chart; see "XRCHART Statement: SHEWHART Procedure" on page 1881. If you specify the STDDEVIATIONS option in the XCHART statement, the variables _LCLS_, _S_, and _UCLS_ are included in the OUTLIMITS= data set. These variables can be used to create an s chart; see "XSCHART Statement: SHEWHART Procedure" on page 1925. For more information, see "OUTLIMITS= Data Set" on page 1864.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outtable=Gaptable
                           nochart;
run;
```

The data set Gaptable is listed in Figure 19.103.

Figure 19.103 The Data Set Gaptable

Summary Statistics and Control Limit Information

VAR	Sample	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_	_EXLIM_
Partgap	1	3	5	5	242.087	270	259.667	277.246	13.1028	
Partgap	2	3	5	5	242.087	258	259.667	277.246	13.1028	
Partgap	3	3	5	5	242.087	248	259.667	277.246	13.1028	
Partgap	4	3	5	5	242.087	260	259.667	277.246	13.1028	
Partgap	5	3	5	5	242.087	273	259.667	277.246	13.1028	
Partgap	6	3	5	5	242.087	260	259.667	277.246	13.1028	
Partgap	7	3	5	5	242.087	259	259.667	277.246	13.1028	
Partgap	8	3	5	5	242.087	248	259.667	277.246	13.1028	
Partgap	9	3	5	5	242.087	260	259.667	277.246	13.1028	
Partgap	10	3	5	5	242.087	255	259.667	277.246	13.1028	
Partgap	11	3	5	5	242.087	268	259.667	277.246	13.1028	
Partgap	12	3	5	5	242.087	253	259.667	277.246	13.1028	
Partgap	13	3	5	5	242.087	273	259.667	277.246	13.1028	
Partgap	14	3	5	5	242.087	275	259.667	277.246	13.1028	
Partgap	15	3	5	5	242.087	257	259.667	277.246	13.1028	
Partgap	16	3	5	5	242.087	269	259.667	277.246	13.1028	
Partgap	17	3	5	5	242.087	249	259.667	277.246	13.1028	
Partgap	18	3	5	5	242.087	264	259.667	277.246	13.1028	
Partgap	19	3	5	5	242.087	258	259.667	277.246	13.1028	
Partgap	20	3	5	5	242.087	248	259.667	277.246	13.1028	
Partgap	21	3	5	5	242.087	248	259.667	277.246	13.1028	

This data set contains one observation for each subgroup sample. The variables _SUBX_ and _SUBN_ contain the subgroup means and sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Sample contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1867.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Gaptable and display an $\bar{X}$ chart (not shown here) identical to the chart in Figure 19.97:

```
title 'Mean Chart for Gap Widths';
proc shewhart table=Gaptable;
   xchart Partgap*Sample;
label _SUBX_ = 'Gap Width';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143).

For more information, see "TABLE= Data Set" on page 1870.

Reading Preestablished Control Limits

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Gaplim saved control limits computed from the measurements in Partgaps. This example shows how these limits can be applied to new data provided in the following data set:

```
data Gaps2;
   input Sample @;
   do i=1 to 5;
      input Partgap @;
      output;
   end;
   drop i;
   datalines;
22 287 265 248 263 271
23 267 253 285 251 271
24 249 252 277 269 241
25 243 248 263 282 261
26 287 266 256 278 242
27 251 262 243 274 245
28 256 245 244 243 272
29 262 247 252 277 266
30 244 269 263 278 261
31 245 264 246 242 273
32 272 257 277 265 241
33 251 249 240 260 261
34 289 277 275 273 261
35 267 286 275 261 272
36 266 256 247 255 241
37 291 267 267 252 262
38 258 245 264 245 281
39 277 267 241 272 244
40 252 267 272 245 252
41 243 241 245 263 248
;
```

The following statements create an $\bar{X}$ chart for the data in Gaps2 using the control limits in Gaplim:

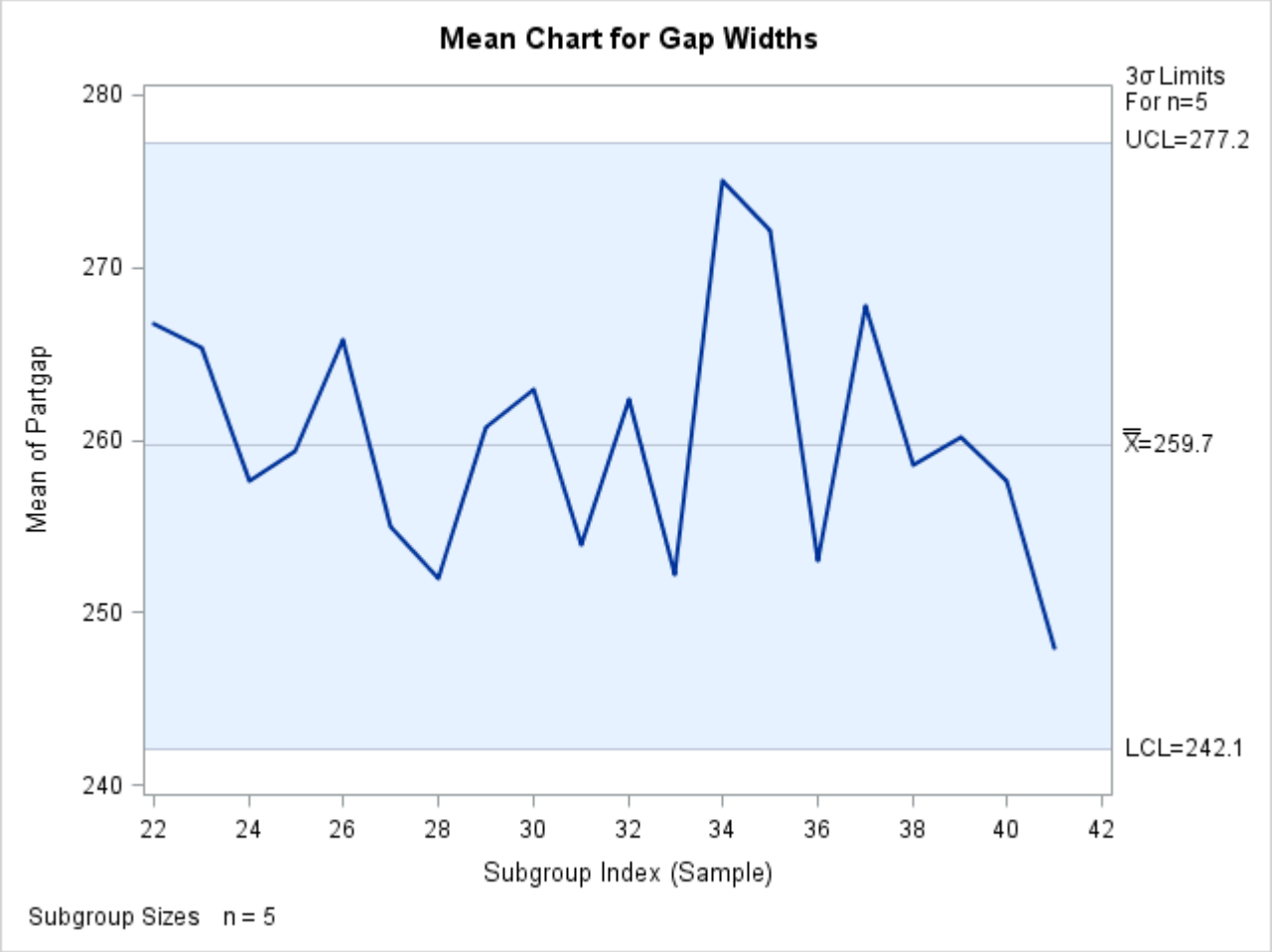
```
ods graphics on;
title 'Mean Chart for Gap Widths';
proc shewhart data=Gaps2 limits=Gaplim;
   xchart Partgap*Sample / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the $\bar{X}$ chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 19.104.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Partgap
- the value of _SUBGRP_ matches the *subgroup-variable* name Sample

Figure 19.104 $\bar{X}$ Chart for Second Set of Gap Width Data (ODS Graphics)



The chart indicates that the process is in control, because all the means lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1868 for details concerning the variables that you must provide.

Syntax: XCHART Statement

The basic syntax for the XCHART statement is as follows:

XCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

XCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of XCHART statements in the SHEWHART procedure. The components of the XCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means from Raw Data" on page 1839.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means from Subgroup Summary Data" on page 1842.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1846.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   xchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See the section "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable
is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See the section "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character
specifies a plotting character for line printer charts. For example, the following statements create an $\bar{X}$ chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   xchart Weight*Day='*';
run;
```

options
enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following table lists the XCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.63 XCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)

Table 19.63 *continued*

Option	Description
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line
NOCTL	Suppresses display of central line
NOLCL	Suppresses display of lower control limit
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit
UCLLABEL=	Specifies label for upper control limit
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line
Process Mean and Standard Deviation Options	
MU0=	Specifies known value of μ_0 for process mean μ
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
STDDEVIATIONS	Specifies that estimate of process standard deviation σ is to be calculated from subgroup standard deviations
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on $\bar{X}$ chart
ALLLABEL2=	Labels every point on trend chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart

Table 19.63 *continued*

Option	Description
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
NOTRENDCONNECT	Suppresses line segments that connect points on trend chart
OUTLABEL=	Labels points outside control limits
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied
ZONELABELS	Adds labels A, B, and C to zone lines
ZONES	Adds lines delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels zone lines with their values

Table 19.63 *continued*

Option	Description
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis

Table 19.63 *continued*

Option	Description
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=	Specifies major tick mark values for vertical axis of trend chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	Specifies number of pages or screens to be used to display chart

Table 19.63 *continued*

Option	Description
TRENDVAR=	Specifies list of trend variables
YPCT1=	Specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and trend charts
ZEROSTD	Displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option

Table 19.63 *continued*

Option	Description
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=	Specifies annotate data set that adds features to trend chart
DESCRIPTION=	Specifies description of $\bar{X}$ chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of $\bar{X}$ chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
WTREND=	Specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency

Table 19.63 *continued*

Option	Description
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed

Table 19.63 *continued*

Option	Description
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend

Table 19.63 *continued*

Option	Description
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays

Table 19.63 *continued*

Option	Description
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines

Table 19.63 *continued*

Option	Description
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: XCHART Statement

The following sections provide details that are specific to the XCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Means

The following notation is used in this section:

μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	Mean of measurements in ith subgroup
R_i	Range of measurements in ith subgroup
n_i	Sample size of ith subgroup
N	Number of subgroups
$\overline{\overline{X}}$	Weighted average of subgroup means
z_p	$100p$th percentile of the standard normal distribution

Plotted Points

Each point on an $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 14}{5} = 15.2$$

Central Line

By default, the central line on an $\bar{X}$ chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1 \bar{X}_1 + \cdots + n_N \bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of $\bar{X}_i$ above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ exceeds the limits

Table 19.65 provides the formulas for the limits.

Table 19.65 Limits for $\bar{X}$ Charts

Control Limits
LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$
UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
Probability Limits
LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$

Note that the limits vary with n_i. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 19.65.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.66 lists the variables that can be saved.

Table 19.66 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_CP_	Capability index C_p
CPK	Capability index C_{pk}

Table 19.66 *continued*

Variable	Description
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLR	Lower control limit for subgroup range
LCLS	Lower control limit for subgroup standard deviation
LCLX	Lower control limit for subgroup mean
LIMITN	Sample size associated with the control limits
LSL	Lower specification limit
MEAN	Process mean ($\overline{\overline{X}}$ or μ_0)
_R_	Value of central line on R chart
_S_	Value of central line on s chart
SIGMAS	Multiple (k) of standard error of $\bar{X}_i$
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the XCHART statement
TARGET	Target value
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	Upper control limit for subgroup range
UCLS	Upper control limit for subgroup standard deviation
UCLX	Upper control limit for subgroup mean
USL	Upper specification limit
VAR	*Process* specified in the XCHART statement

Notes:

1. The variables _LCLS_, _S_, and _UCLS_ are included if you specify the STDDEVIATIONS option; otherwise, the variables _LCLR_, _R_, and _UCLR_ are included. These variables are not used to create $\bar{X}$ charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value 'V' is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.

3. If the limits are defined in terms of a multiple k of the standard error of $\bar{X}_i$, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XCHART statement. For an example, see "Saving Control Limits" on page 1846.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup standard deviation variable named by *process* suffixed with *S*

A subgroup standard deviation variable is included if you specify the STDDEVIATIONS option; otherwise, a subgroup range variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the XCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   xchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthR, WidthN, DiameterX, DiameterR, and DiameterN. The variables WidthR and DiameterR are included, because the STDDEVIATIONS option is not specified. If you specified the STDDEVIATIONS option, the data set Summary would contain the variables WidthS and DiameterS rather than WidthR and DiameterR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1845.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.67 lists the variables that can be saved.

Table 19.67 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on $\bar{X}$ chart
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample size
SUBX	Subgroup mean
TESTS	Tests for special causes signaled on $\bar{X}$ chart
UCLX	Upper control limit for mean
VAR	*Process* specified in the XCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1846.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see the section "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating Charts for Means from Raw Data" on page 1839.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   xchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 19.66. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, and _UCLX_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 19.65

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1849.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the XCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or subgroup standard deviation variable for each *process*

If you specify the STDDEVIATIONS option, the subgroup standard deviation variable must be included; otherwise, the subgroup range variable must be included.

The names of the subgroup mean, subgroup range or subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters *X*, *R* or *S*, and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   xchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightR, WeightN, YieldstrengthX, YieldstrengthR, and YieldstrengthN. If the STDDEVIATIONS option were specified in the preceding XCHART statement, it would be necessary for Summary to include the variables Batch, WeightX, WeightS, WeightN, YieldstrengthX, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see the section "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means from Subgroup Summary Data" on page 1842.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.68 lists the variables required in a TABLE= data set used with the XCHART statement.

Table 19.68 Variables Required in a TABLE= Data Set

Variable	Description
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
Subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBX	Subgroup mean
UCLX	Upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1846.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available for estimating the process standard deviation σ. The method depends on whether you specify the STDDEVIATIONS option. If you specify this option, σ is estimated using subgroup standard deviations; otherwise, σ is estimated using subgroup ranges.

For an illustration of the methods, see Example 19.35.

Default Method Based on Subgroup Ranges

If you do not specify the STDDEVIATIONS option, the default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

Default Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option, the default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method Based on Subgroup Ranges

If you do not specify the STDDEVIATIONS option and you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVLUE Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option and SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option and SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method Based on Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as $\hat{\sigma} = \bar{R}/d_2(2)$, where $\bar{R}$ is the average of the moving ranges of consecutive measurements taken in pairs. This is the method used to estimate σ for individual measurements and moving range charts. See "Methods for Estimating the Standard Deviation" on page 1550.

Examples: XCHART Statement

This section provides advanced examples of the XCHART statement.

Example 19.34: Applying Tests for Special Causes

NOTE: See *Mean Chart-Tests for Special Causes Applied* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make $\bar{X}$ charts more sensitive to special causes of variation.

The following statements create an $\bar{X}$ chart for the gap width measurements in the data set Parts in "Creating Charts for Means from Subgroup Summary Data" on page 1842 and tabulate the results:

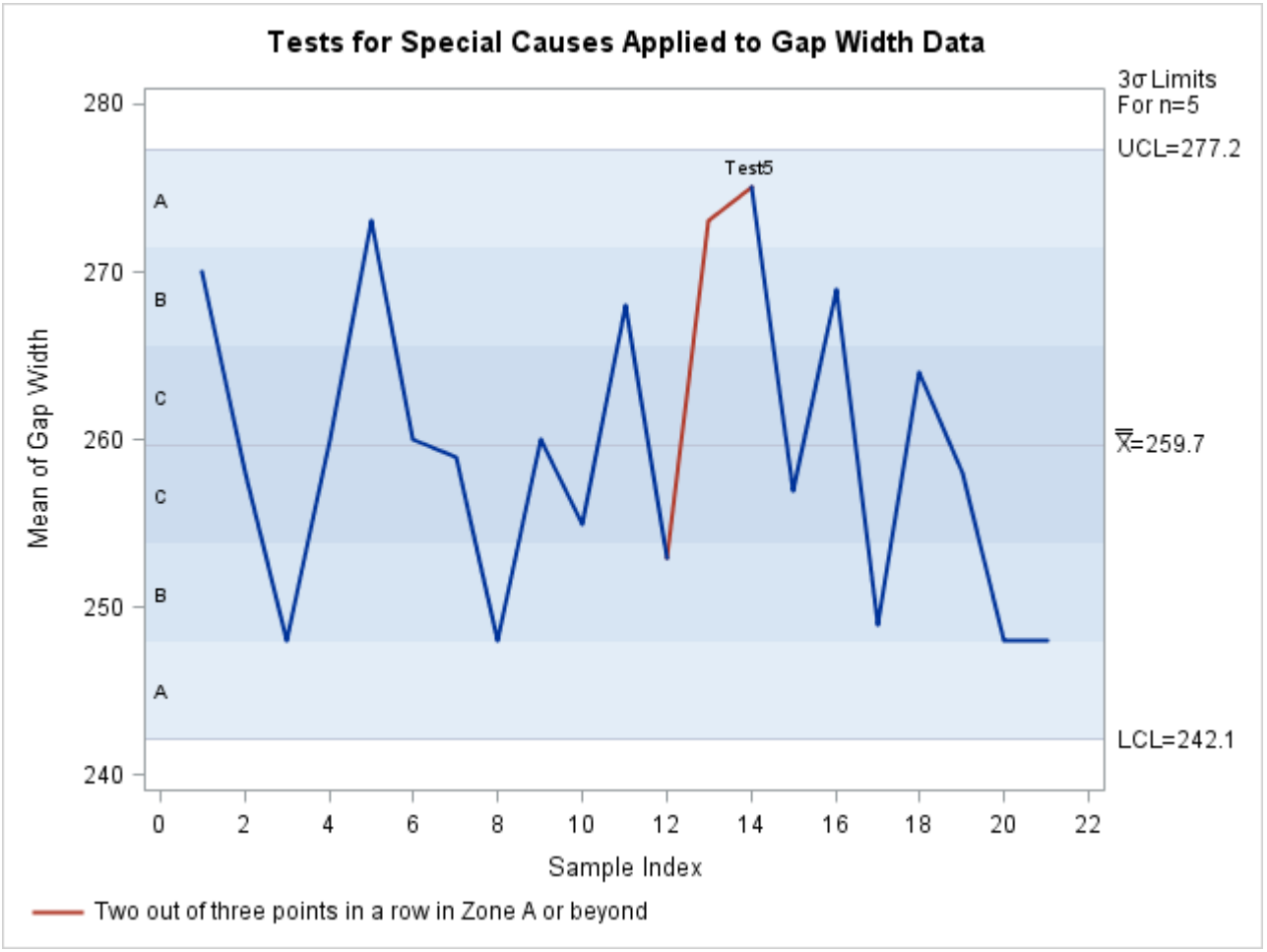
```
ods graphics on;
title 'Tests for Special Causes Applied to Gap Width Data';
proc shewhart history=Parts;
   xchart Partgap*Sample/ tests    = 1 to 5
                          odstitle = title
                          tabletests
                          nolegend
                          tablecentral
                          tablelegend
                          zonelabels;
run;
```

The $\bar{X}$ chart is shown in Output 19.34.1 and the printed output is shown in Output 19.34.2. The TESTS= requests Tests 1, 2, 3, 4, and 5, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. The TABLECENTRAL option requests a table of the subgroup means, control limits, and central line. The TABLETESTS option adds a column indicating which subgroups tested positive for special causes, and the TABLELEGEND option adds a legend describing the tests that were signaled.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The NOLEGEND option suppresses the subgroup sample size legend that is displayed by default in the lower left corner of the chart.

Output 19.34.1 and Output 19.34.2 indicate that Test 5 was positive at sample 14, signaling a possible shift in the mean of the process.

Output 19.34.1 Tests for Special Causes Displayed on an $\bar{X}$ Chart



Output 19.34.2 Tabular Form of $\bar{X}$ Chart

Tests for Special Causes Applied to Gap Width Data

The SHEWHART Procedure

Means Chart Summary for Partgap

3 Sigma Limits with n=5 for Mean

Sample	Subgroup Sample Size	Lower Limit	Subgroup Mean	Average Mean	Upper Limit	Special Tests Signaled
1	5	242.08741	270.00000	259.66667	277.24592	
2	5	242.08741	258.00000	259.66667	277.24592	
3	5	242.08741	248.00000	259.66667	277.24592	
4	5	242.08741	260.00000	259.66667	277.24592	
5	5	242.08741	273.00000	259.66667	277.24592	
6	5	242.08741	260.00000	259.66667	277.24592	
7	5	242.08741	259.00000	259.66667	277.24592	
8	5	242.08741	248.00000	259.66667	277.24592	
9	5	242.08741	260.00000	259.66667	277.24592	
10	5	242.08741	255.00000	259.66667	277.24592	
11	5	242.08741	268.00000	259.66667	277.24592	
12	5	242.08741	253.00000	259.66667	277.24592	
13	5	242.08741	273.00000	259.66667	277.24592	
14	5	242.08741	275.00000	259.66667	277.24592	5
15	5	242.08741	257.00000	259.66667	277.24592	
16	5	242.08741	269.00000	259.66667	277.24592	
17	5	242.08741	249.00000	259.66667	277.24592	
18	5	242.08741	264.00000	259.66667	277.24592	
19	5	242.08741	258.00000	259.66667	277.24592	
20	5	242.08741	248.00000	259.66667	277.24592	
21	5	242.08741	248.00000	259.66667	277.24592	

Test Descriptions	
Test 5	Two out of three points in a row in Zone A or beyond

Example 19.35: Estimating the Process Standard Deviation

NOTE: See *Estimating the Process Standard Deviation* in the SAS/QC Sample Library.

The following data set (Wire) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

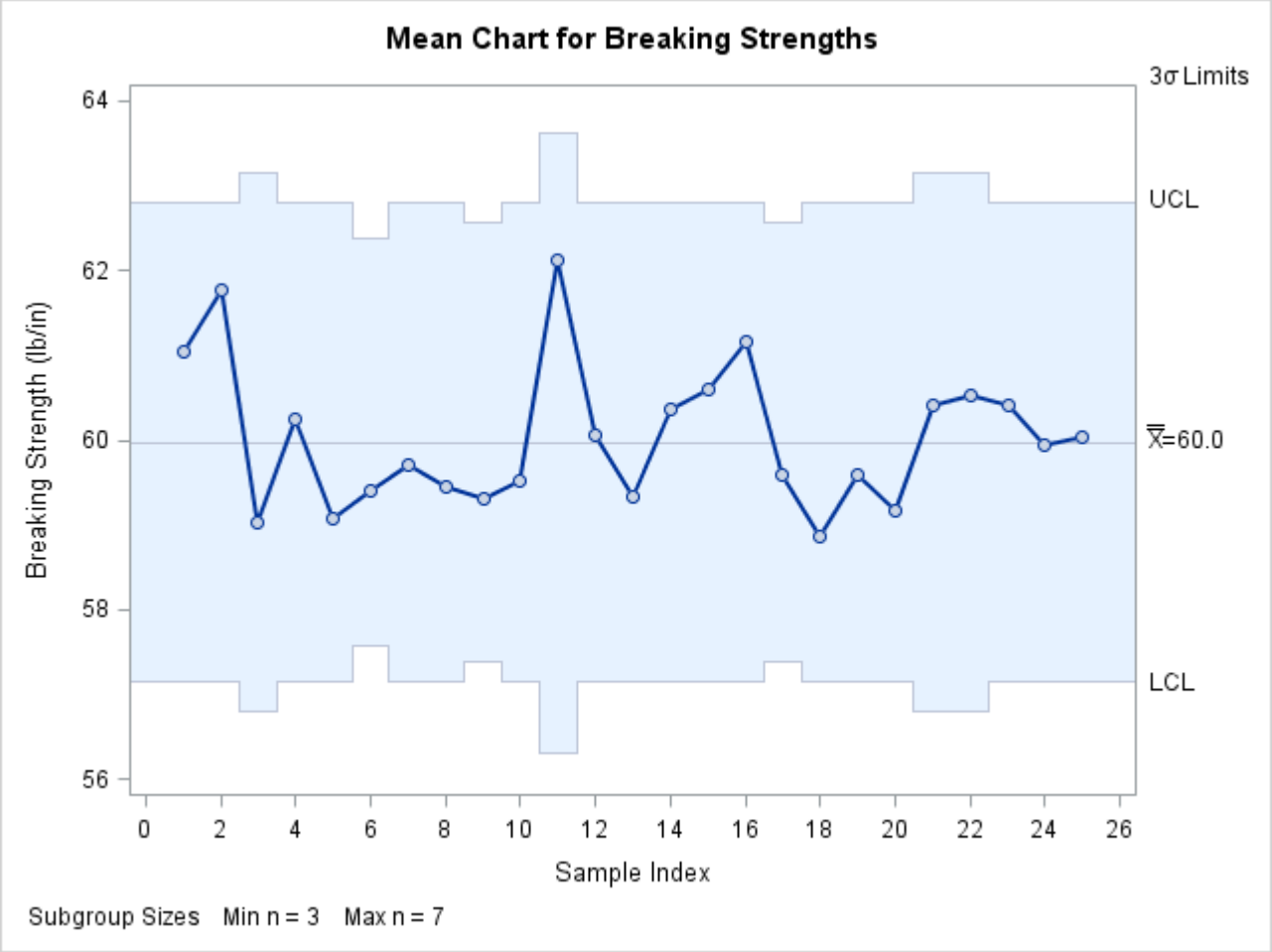
```
data Wire;
   input Sample Size @;
   do i=1 to Size;
      input Breakstrength @@;
      output;
   end;
   drop i Size;
   label Breakstrength  ='Breaking Strength (lb/in)'
```

```
          Sample ='Sample Index';
    datalines;
 1  5 60.6 62.3 62.0 60.4 59.9
 2  5 61.9 62.1 60.6 58.9 65.3
 3  4 57.8 60.5 60.1 57.7
 4  5 56.8 62.5 60.1 62.9 58.9
 5  5 63.0 60.7 57.2 61.0 53.5
 6  7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
 7  5 59.3 61.7 59.1 58.1 60.3
 8  5 61.3 58.5 57.8 61.0 58.6
 9  6 59.5 58.3 57.5 59.4 61.5 59.6
10  5 61.7 60.7 57.2 56.5 61.5
11  3 63.9 61.6 60.9
12  5 58.7 61.4 62.4 57.3 60.5
13  5 56.8 58.5 55.7 63.0 62.7
14  5 62.1 60.6 62.1 58.7 58.3
15  5 59.1 60.4 60.4 59.0 64.1
16  5 59.9 58.8 59.2 63.0 64.9
17  6 58.8 62.4 59.4 57.1 61.2 58.6
18  5 60.3 58.7 60.5 58.6 56.2
19  5 59.2 59.8 59.7 59.3 60.0
20  5 62.3 56.0 57.0 61.8 58.8
21  4 60.5 62.0 61.4 57.7
22  4 59.3 62.4 60.4 60.0
23  5 62.4 61.3 60.5 57.7 60.2
24  5 61.2 55.5 60.2 60.4 62.4
25  5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request an $\bar{X}$ chart, shown in Output 19.35.1, for the breaking strength measurements:

```
ods graphics on;
title 'Mean Chart for Breaking Strengths';
proc shewhart data=Wire;
   xchart Breakstrength*Sample / odstitle = title
                                 markers;
run;
```

Output 19.35.1 $\bar{X}$ Chart with Varying Subgroup Sample Sizes



Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

By default, the control limits are 3σ limits estimated from the data. You can use the STDDEVIATIONS option and the SMETHOD= option to specify how the estimate of the process standard deviation σ is to be computed, as illustrated by the following statements:

```
proc shewhart data=Wire;
   xchart Breakstrength*Sample / outlimits=Wirelim1
                         outindex ='Default-Ranges'
                         nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim2
                         stddeviations
                         outindex ='Default-Stds'
                         nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim3
                         smethod  =mvlue
                         outindex ='MVLUE  -Ranges'
                         nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim4
                         stddeviations
```

```
                          smethod  =mvlue
                          outindex ='MVLUE  -Stds'
                          nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim5
                          stddeviations
                          smethod  =rmsdf
                          outindex ='RMSDF  -Stds'
                          nochart;
run;
```

The STDDEVIATIONS option specifies that the estimate is to be calculated from subgroup standard deviations rather than subgroup ranges, the default. The SMETHOD= option specifies the method for estimating σ. The default method estimates σ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests a minimum variance linear unbiased estimate, and specifying SMETHOD=RMSDF requests a weighted root-mean-square estimate. For details, see "Methods for Estimating the Standard Deviation" on page 1871.

The variable _STDDEV_ in each OUTLIMITS= data set contains the estimate of σ. The OUTINDEX= option specifies the value of the variable _INDEX_ in the OUTLIMITS= data set and is used here to identify the estimation method.

The following statements merge the five OUTLIMITS= data sets into a single data set, which is listed in Output 19.35.2:

```
data Wlimits;
   set Wirelim1 Wirelim2 Wirelim3 Wirelim4 Wirelim5;
   keep _index_ _stddev_;
run;
```

Output 19.35.2 The Data Set WLIMITS

Estimates of the Process Standard Deviation

INDEX	_STDDEV_
Default-Ranges	2.11146
Default-Stds	2.15453
MVLUE -Ranges	2.11240
MVLUE -Stds	2.14790
RMSDF -Stds	2.17479

The $\bar{X}$ chart shown in Output 19.35.1 uses the default estimate listed first in Output 19.35.2 ($\sigma = 2.11146$). In this case, there is very little difference in the five estimates, because the sample sizes do not differ greatly. In general, the MVLUE's are recommended with large sample sizes ($n_i \geq 10$).

Example 19.36: Plotting OC Curves for Mean Charts

NOTE: See *Plotting OC Curves for Mean Charts* in the SAS/QC Sample Library.

This example uses the GPLOT procedure and the DATA step function PROBNORM to plot operating characteristic (OC) curves for $\bar{X}$ charts with 3σ limits. An OC curve is plotted for each of the subgroup samples sizes 1, 2, 3, 4, and 16. Refer to page 226 in Montgomery (1996). Each curve plots the probability β of not detecting a shift of magnitude $\nu\sigma$ in the process mean as a function of ν. The value of β is computed using the following formula:

$$\begin{aligned} \beta &= P\{LCL \leq \bar{X}_i \leq UCL\} \\ &= \Phi(3 - \nu\sqrt{n}) - \Phi(-3 - \nu\sqrt{n}) \end{aligned}$$

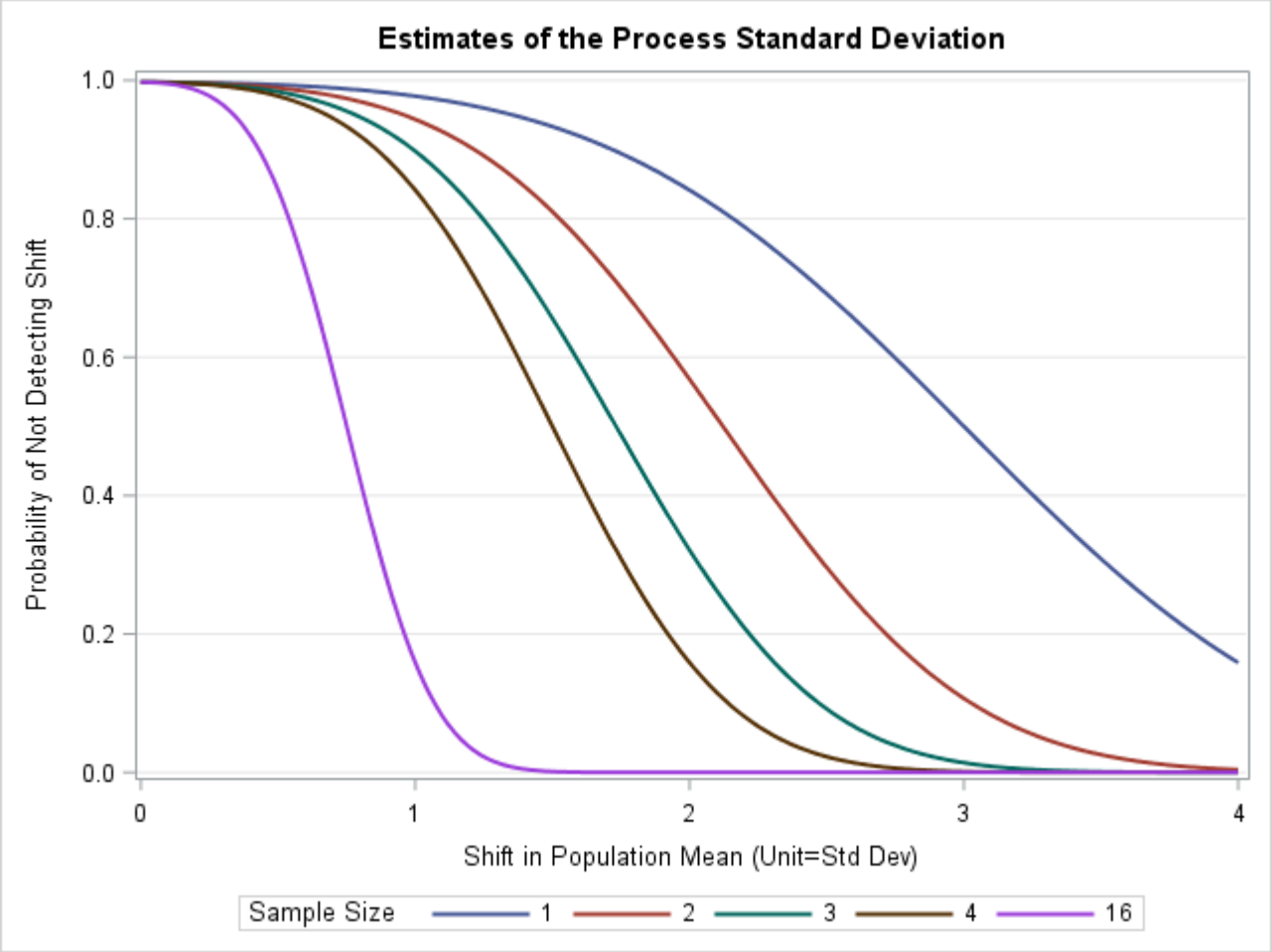
The following statements compute β (the variable prob) as a function of ν (the variable t). The variable nSample contains the sample size.

```
data oc;
   keep prob nSample t plot2;
   plot2=.;
   do nSample=1, 2, 3, 4, 16;
      do j=0 to 400;
         t=j/100;
         prob=probnorm( 3-t*sqrt(nSample)) -
              probnorm(-3-t*sqrt(nSample));
         output;
      end;
   end;
   label t   ='Shift in Population Mean (Unit=Std Dev)'
        prob='Probability of Not Detecting Shift';
run;
```

The following statements use the GPLOT procedure to display the OC curves shown in Output 19.36.1:

```
proc sgplot data=oc;
   series x=t y=prob /
      group=nSample lineattrs=(pattern=solid thickness=2);
   yaxis grid;
   label nSample='Sample Size';
run;
```

Output 19.36.1 OC Curves for Different Subgroup Sample Sizes



Example 19.37: Computing Process Capability Indices

You can save process capability indices in an OUTLIMITS= data set if you provide specification limits with the LSL= and USL= options. This is illustrated by the following statements:

```
title 'Control Limits and Capability Indices';
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outlimits = Gaplim2
                           usl       = 270
                           lsl       = 240
                           nochart;
run;
```

The data set Gaplim2 is listed in Output 19.37.1.

Output 19.37.1 Data Set Gaplim2 Containing Control Limit Information

Control Limits with Capability Indices for Gap Width Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_	_LCLR_
Partgap	Sample	ESTIMATE	5	.002699796	3	242.087	259.667	277.246	0

_R_	_UCLR_	_STDDEV_	_LSL_	_USL_	_CP_	_CPL_	_CPU_	_CPK_
30.4762	64.4419	13.1028	240	270	0.38160	0.50032	0.26288	0.26288

The variables _CP_, _CPL_, _CPU_, and _CPK_ contain the process capability indices. It is reasonable to compute capability indices in this case, because Figure 19.97 indicates that the process is in statistical control. For more information, see the section "OUTLIMITS= Data Set" on page 1864.

XRCHART Statement: SHEWHART Procedure

Overview: XRCHART Statement

The XRCHART statement creates $\bar{X}$ and R charts for subgroup means and ranges, which are used to analyze the central tendency and variability of a process.

You can use options in the XRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means and ranges or as probability limits
- tabulate subgroup sample sizes, subgroup means, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines

- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing $\bar{X}$ and R charts with the XRCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: XRCHART Statement

This section introduces the XRCHART statement with simple examples illustrating commonly used options. Complete syntax for the XRCHART statement is presented in the section "Syntax: XRCHART Statement" on page 1894, and advanced examples are given in the section "Examples: XRCHART Statement" on page 1916.

Creating Charts for Means and Ranges from Raw Data

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

In the manufacture of silicon wafers, batches of five wafers are sampled, and their diameters are measured in millimeters. The following statements create a SAS data set named Wafers, which contains the measurements for 25 batches:

```
data Wafers;
   input Batch @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
 1  35.00 34.99 34.99 34.98 35.00
 2  35.01 34.99 34.99 34.98 35.00
 3  34.99 35.00 35.00 35.00 35.00
 4  35.01 35.00 34.99 34.99 35.00
 5  35.00 34.99 34.98 34.99 35.00
 6  34.99 34.99 35.00 35.00 35.00
 7  35.01 34.98 35.00 35.00 34.99
 8  35.00 35.00 34.99 34.98 34.99
 9  34.99 34.98 34.98 35.01 35.00
10  34.99 35.00 35.01 34.99 35.01
11  35.01 35.00 35.00 34.98 34.99
12  34.99 34.99 35.00 34.98 35.01
```

```
13  35.01 34.99 34.98 34.99 34.99
14  35.00 35.00 34.99 35.01 34.99
15  34.98 34.99 34.99 34.98 35.00
16  34.99 35.00 35.00 35.01 35.00
17  34.98 34.98 34.99 34.99 34.98
18  35.01 35.02 35.00 34.98 35.00
19  34.99 34.98 35.00 34.99 34.98
20  34.99 35.00 35.00 34.99 34.99
21  35.00 34.99 34.99 34.98 35.00
22  35.00 35.00 35.01 35.00 35.00
23  35.02 35.00 34.98 35.02 35.00
24  35.00 35.00 34.99 35.01 34.98
25  34.99 34.99 34.99 35.00 35.00
;
```

The following statements use the PRINT procedure to list the data set Wafers. A portion of this listing is shown in Figure 19.105.

```
title 'The Data Set Wafers';
proc print data=Wafers noobs;
run;
```

Figure 19.105 Partial Listing of the Data Set Wafers

The Data Set Wafers

Batch	Diameter
1	35.00
1	34.99
1	34.99
1	34.98
1	35.00
2	35.01
2	34.99
2	34.99
2	34.98
2	35.00
3	34.99
3	35.00
3	35.00
3	35.00
3	35.00

The data set Wafers is said to be in "strung-out" form because each observation contains the batch number and diameter measurement for a single wafer. The first five observations contain the diameters for the first batch, the second five observations contain the diameters for the second batch, and so on. Because the variable Batch classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Diameter contains the wafer diameter measurements and is referred to as the *process variable* (or *process* for short).

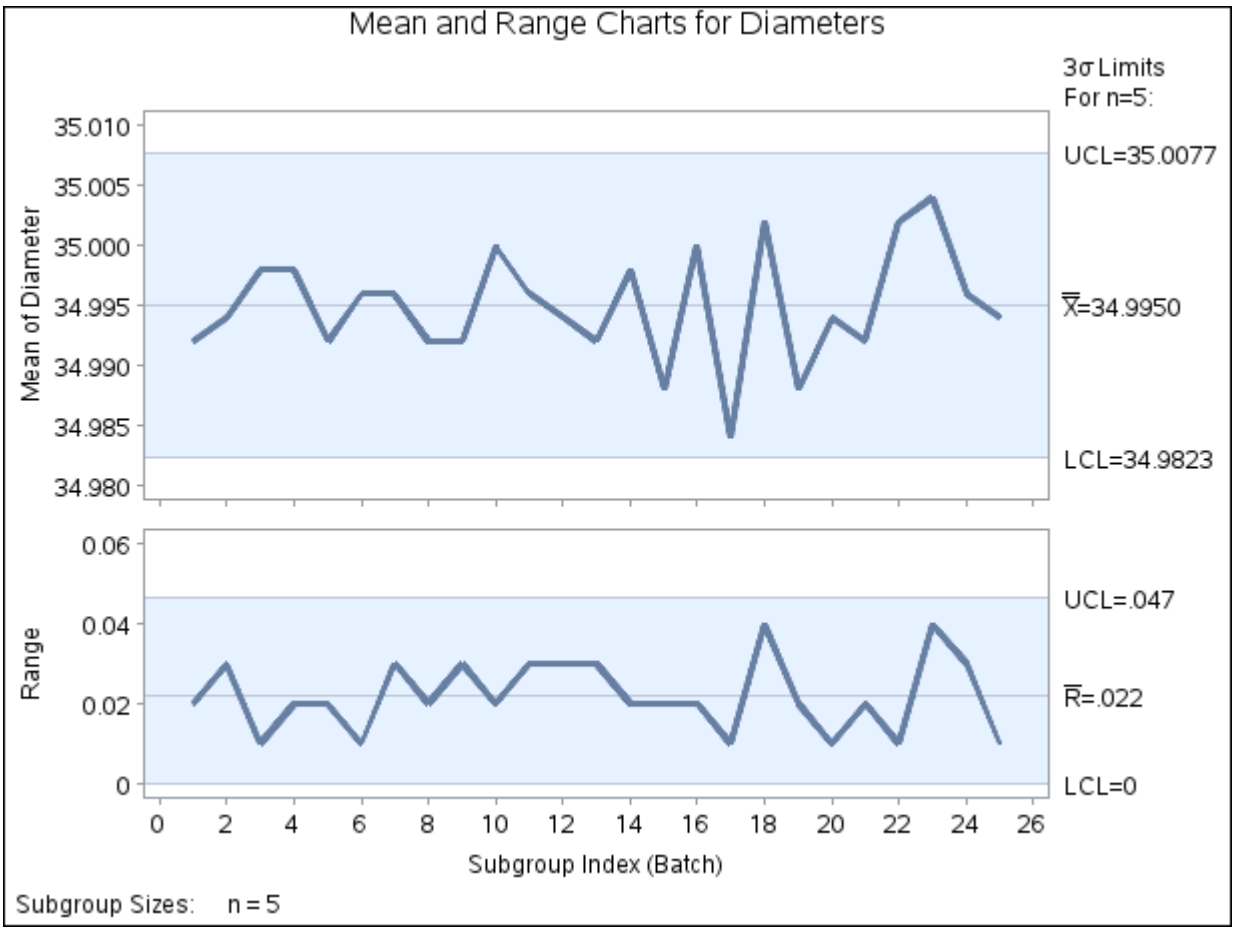
You can use $\bar{X}$ and R charts to determine whether the manufacturing process is in control. The following statements create the $\bar{X}$ and R charts shown in Figure 19.106:

```
ods graphics off;
title 'Mean and Range Charts for Diameters';
proc shewhart data=Wafers;
   xrchart Diameter*Batch;
run;
```

This example illustrates the basic form of the XRCHART statement. After the keyword XRCHART, which specifies the type of control chart to display, you specify the *process* to analyze (in this case, Diameter) followed by an asterisk and the *subgroup-variable* (Batch).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. By default, traditional graphics output is produced, and its appearance is governed by the style in effect for any given ODS destination. See Chapter 4, "SAS/QC Graphics," for a discussion of alternatives for producing graphics with SAS/QC procedures.

Figure 19.106 $\bar{X}$ and R Charts for Wafer Diameter Data (Traditional Graphics)



Each point on the $\bar{X}$ chart represents the average (mean) of the measurements for a particular batch. For instance, the mean plotted for the first batch is

$$\frac{35.00 + 34.99 + 34.99 + 34.98 + 35.00}{5} = 34.992$$

Each point on the *R* chart represents the range of the measurements for a particular batch. For instance, the range plotted for the first batch is 35.00 – 34.98 = 0.02.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.71. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1892.

Because all the points lie within the control limits, it can be concluded that the process is in statistical control. For computational details, see "Constructing Charts for Means and Ranges" on page 1907. For more details on reading raw data, see "DATA= Data Set" on page 1912.

Creating Charts for Means and Ranges from Summary Data

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

The previous example illustrates how you can create $\bar{X}$ and *R* charts based on raw data (process measurements). However, in many applications, the data are provided as subgroup means and ranges. This example illustrates how you can use the XRCHART statement with data of this type.

The following data set (Wafersum) provides the data from the preceding example in summarized form:

```
data Wafersum;
   input Batch DiameterX DiameterR;
   DiameterN = 5;
   datalines;
 1  34.992  0.02
 2  34.994  0.03
 3  34.998  0.01
 4  34.998  0.02
 5  34.992  0.02
 6  34.996  0.01
 7  34.996  0.03
 8  34.992  0.02
 9  34.992  0.03
10  35.000  0.02
11  34.996  0.03
12  34.994  0.03
13  34.992  0.03
14  34.998  0.02
15  34.988  0.02
16  35.000  0.02
17  34.984  0.01
18  35.002  0.04
19  34.988  0.02
20  34.994  0.01
21  34.992  0.02
22  35.002  0.01
23  35.004  0.04
24  34.996  0.03
25  34.994  0.01
;
```

A partial listing of the data set Wafersum is shown in Figure 19.107.

Figure 19.107 Partial Listing of the Summary Data Set Wafersum

Summary Data Set for Wafer Diameters

Batch	DiameterX	DiameterR	DiameterN
1	34.992	0.02	5
2	34.994	0.03	5
3	34.998	0.01	5
4	34.998	0.02	5
5	34.992	0.02	5

In this data set, there is exactly one observation for each subgroup (note that the subgroups are still indexed by Batch). The variable DiameterX contains the subgroup means, the variable DiameterR contains the subgroup ranges, and the variable DiameterN contains the subgroup sample sizes (these are all equal to five).

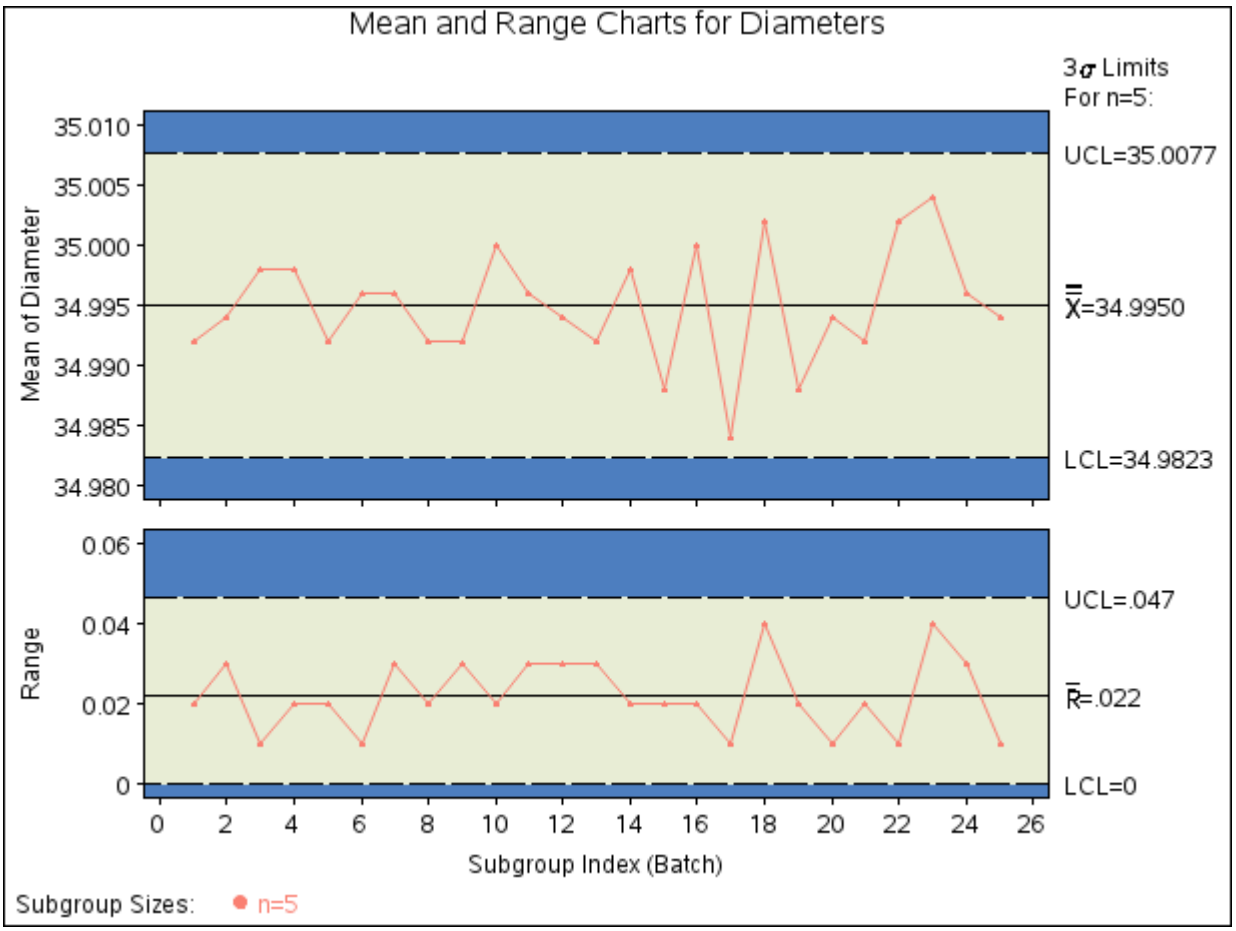
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
title 'Mean and Range Charts for Diameters';
symbol value = dot color = salmon;
proc shewhart history=Wafersum;
   xrchart Diameter*Batch / cframe   = bigb
                            cinfill  = ywh
                            cconnect = salmon
                            coutfill = yellow;
run;
options gstyle;
```

Note that Diameter is *not* the name of a SAS variable in the data set Wafersum but is, instead, the common prefix for the names of the three SAS variables DiameterX, DiameterR, and DiameterN. The suffix characters *X*, *R*, and *N* indicate *mean*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (Diameter), which is referred to as the *process*. The name Batch specified after the asterisk is the name of the *subgroup-variable*.

The NOGSTYLE system option causes the XRCHART statement to ignore ODS styles when producing traditional graphics. Instead, global statements (such as GOPTIONS, SYMBOL, and AXIS statements) and XRCHART options, specified after the slash (/) in the XRCHART statement, control the appearance of the charts. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. A complete list of options is presented in the section "Syntax: XRCHART Statement" on page 1894. For more information about the GOPTIONS and SYMBOL statements, see *SAS/GRAPH: Reference*. The resulting charts are shown in Figure 19.108.

Figure 19.108 $\bar{X}$ and R Charts from Summary Data (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the XRCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, range, and sample size variables must begin with the *process* name specified in the XRCHART statement and end with the special suffix characters *X*, *R*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables DiameterX, DiameterR, and DiameterN, the data set Wafersum contained summary variables named means, ranges, and sizes. The following statements would temporarily rename means, ranges, and sizes to DiameterX, DiameterR, and DiameterN, respectively:

```
proc shewhart
   history=Wafersum (rename=(means  = DiameterX
                             ranges = DiameterR
                             sizes  = DiameterN ));
   xrchart Diameter*Batch=;
run;
```

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read by using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read by using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1913.

Saving Summary Statistics

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

In this example, the XRCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Wafers and create a summary data set named Waferhist:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outhistory = Waferhist
                            nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of charts. Figure 19.109 contains a partial listing of Waferhist.

Figure 19.109 Partial Listing of the Summary Data Set Waferhist

Summary Data Set for Wafer Diameters

Batch	DiameterX	DiameterR	DiameterN
1	34.992	0.02	5
2	34.994	0.03	5
3	34.998	0.01	5
4	34.998	0.02	5
5	34.992	0.02	5

There are four variables in the data set Waferhist:

- Batch contains the subgroup index.
- DiameterX contains the subgroup means.
- DiameterR contains the subgroup ranges.
- DiameterN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* Diameter specified in the XRCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1910.

Saving Control Limits

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

You can save the control limits for $\bar{X}$ and *R* charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1892) or modify the limits with a DATA step program.

The following statements read measurements from the data set Wafers (see "Creating Charts for Means and Ranges from Raw Data" on page 1882) and save the control limits displayed in Figure 19.106 in Waferlim:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outlimits = Waferlim
                            nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Waferlim is listed in Figure 19.110.

Figure 19.110 The Data Set Waferlim Containing Control Limit Information

Control Limits for Wafer Diameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Diameter	Batch	ESTIMATE	5	.002699796	3	34.9823	34.9950	35.0077

LCLR	_R_	_UCLR_	_STDDEV_
0	0.022	0.046519	.009458586

The data set Waferlim contains one observation with the limits for *process* Diameter. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the *R* chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart, and the variable _R_ contains the central line for the *R* chart. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable* names. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

You can save process capability indices in an OUTLIMITS= data set if you provide specification limits with the LSL= and USL= options. This is illustrated by the following statements:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outlimits = Waferlim2
                            usl       = 35.03
                            lsl       = 34.97
                            nochart;
run;
```

The data set Waferlim2 is listed in Figure 19.111.

Figure 19.111 The Data Set Waferlim2 Containing Process Capability Indices

Control Limits and Capability Indices

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_	_LCLR_
Diameter	Batch	ESTIMATE	5	.002699796	3	34.9823	34.9950	35.0077	0

_R_	_UCLR_	_STDDEV_	_LSL_	_USL_	_CP_	_CPL_	_CPU_	_CPK_
0.022	0.046519	.009458586	34.97	35.03	1.05724	0.87962	1.23486	0.87962

The variables _CP_, _CPL_, _CPU_, and _CPK_ contain the process capability indices. It is reasonable to compute capability indices, because Figure 19.106 indicates that the wafer process is in statistical control. However, it is recommended that you also check for normality of the data. You can use the CAPABILITY procedure for this purpose.

For more information, see "OUTLIMITS= Data Set" on page 1909.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outtable=Wafertab
                            nochart;
run;
```

The data set Wafertab is listed in Figure 19.112.

Figure 19.112 The Data Set Wafertab

Summary Statistics and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
Diameter	1	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	2	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586
Diameter	3	3	5	5	34.9823	34.998	34.9950	35.0077	.009458586
Diameter	4	3	5	5	34.9823	34.998	34.9950	35.0077	.009458586
Diameter	5	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	6	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	7	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	8	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	9	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	10	3	5	5	34.9823	35.000	34.9950	35.0077	.009458586
Diameter	11	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	12	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586
Diameter	13	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	14	3	5	5	34.9823	34.998	34.9950	35.0077	.009458586
Diameter	15	3	5	5	34.9823	34.988	34.9950	35.0077	.009458586
Diameter	16	3	5	5	34.9823	35.000	34.9950	35.0077	.009458586
Diameter	17	3	5	5	34.9823	34.984	34.9950	35.0077	.009458586
Diameter	18	3	5	5	34.9823	35.002	34.9950	35.0077	.009458586
Diameter	19	3	5	5	34.9823	34.988	34.9950	35.0077	.009458586
Diameter	20	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586
Diameter	21	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	22	3	5	5	34.9823	35.002	34.9950	35.0077	.009458586

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	0.02	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.04	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	

Figure 19.112 *continued*

Summary Statistics and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
Diameter	23	3	5	5	34.9823	35.004	34.9950	35.0077	.009458586
Diameter	24	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	25	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	0.04	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.01	0.022	0.046519	

This data set contains one observation for each subgroup sample. The variables _SUBX_, _SUBR_, and _SUBN_ contain the subgroup means, subgroup ranges, and subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the *R* chart. The variable _MEAN_ contains the central line of the $\bar{X}$ chart, and the variable _R_ contains the central line of the *R* chart. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1911.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Wafertab and display $\bar{X}$ and *R* charts identical to those in Figure 19.106:

```
title 'Mean and Range Charts for Diameters';
proc shewhart table=Wafertab;
   xrchart Diameter*Batch;
run;
```

Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143).

For more information, see "TABLE= Data Set" on page 1914.

Reading Preestablished Control Limits

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

In a previous example, the OUTLIMITS= data set saved control limits computed from the measurements in Wafers. This example shows how these limits can be applied to new data provided in the following data set:

```
data Wafers2;
   input Batch @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
26 34.99 34.99 35.00 34.99 35.00
27 34.99 35.01 34.98 34.98 34.97
```

```
28 35.00 34.99 34.99 34.99 35.01
29 34.98 34.96 34.98 34.98 34.99
30 34.98 35.00 34.98 34.98 34.99
31 35.00 35.00 34.99 35.01 35.01
32 35.00 34.99 34.98 34.98 35.00
33 34.98 35.00 34.99 35.00 35.01
34 35.00 34.97 35.00 34.99 35.01
35 34.99 34.99 34.98 34.99 34.98
36 35.01 34.98 34.99 34.99 35.00
37 35.01 34.99 34.97 34.98 35.00
38 34.98 34.99 35.00 34.98 35.00
39 34.99 34.99 34.99 34.99 35.01
40 34.99 35.01 35.00 35.01 34.99
41 34.99 35.00 34.99 34.98 34.99
42 35.00 34.99 34.98 34.99 35.00
43 34.99 34.98 34.98 34.99 34.99
44 35.00 35.00 34.98 35.00 34.99
45 34.99 34.99 35.00 34.99 34.99
;
```

The following statements use the control limits in Waferlim to create $\bar{X}$ and R charts for the data in Wafers2:

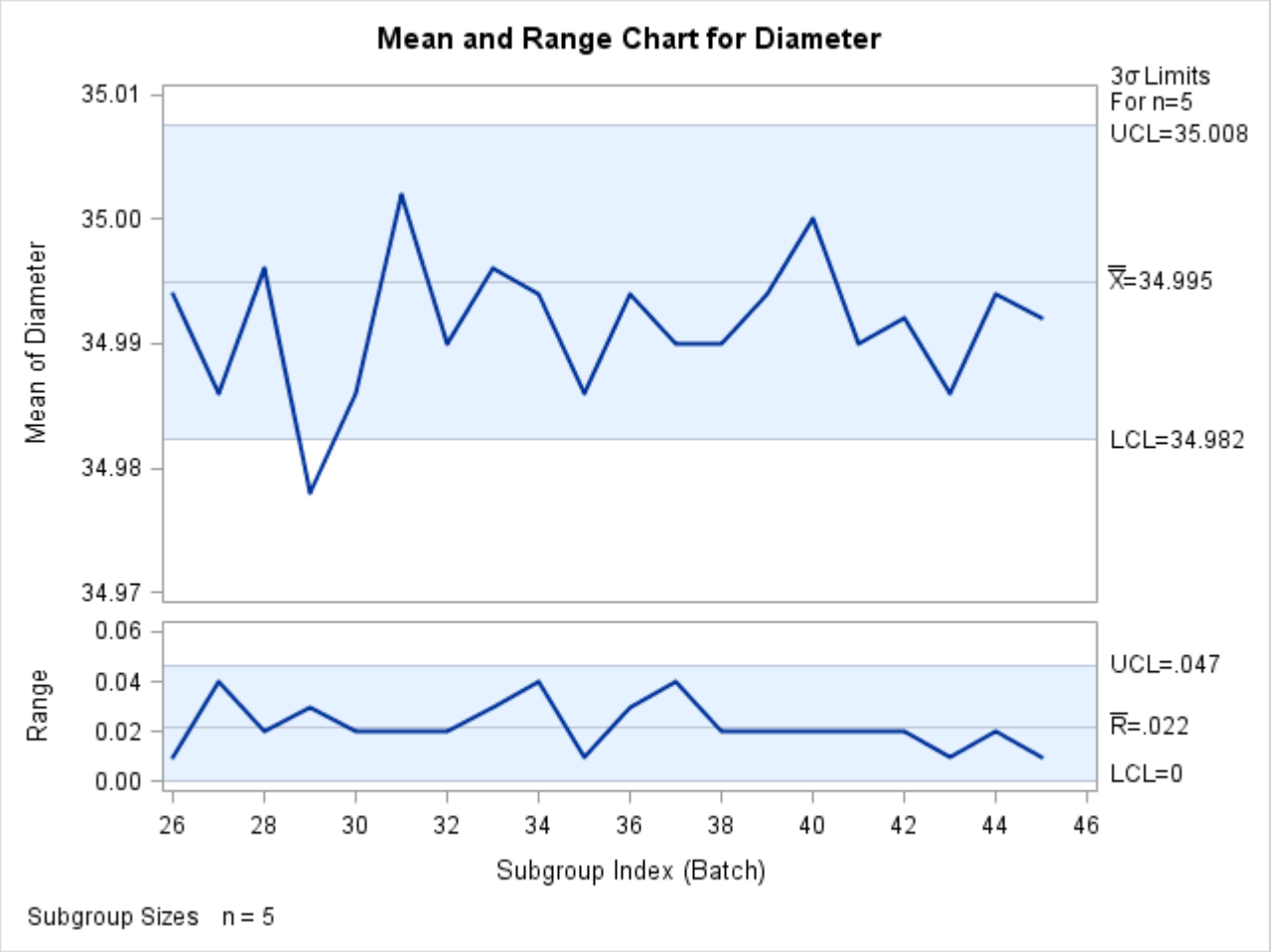
```
ods graphics on;
proc shewhart data=Wafers2 limits=Waferlim;
   xrchart Diameter*Batch;
run;
```

The ODS GRAPHICS ON statement specifies that the $\bar{X}$ and R charts are produced using ODS Graphics. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Diameter
- the value of _SUBGRP_ matches the *subgroup-variable* name Batch

The charts are shown in Figure 19.113.

Figure 19.113 $\bar{X}$ and R Charts for Second Set of Wafer Data (ODS Graphics)



Note that the mean diameter of the 29th batch lies below the lower control limit in the $\bar{X}$ chart, signaling a special cause of variation.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1912 for details concerning the variables that you must provide.

Syntax: XRCHART Statement

The basic syntax for the XRCHART statement is as follows:

XRCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

XRCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of XRCHART statements in the SHEWHART procedure. The components of the XRCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means and Ranges from Raw Data" on page 1882.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means and Ranges from Summary Data" on page 1885.
- If summary data and control limits are read from a TABLE= data set, *process* must be a value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1889.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ and R charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   xrchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XRCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means and ranges.

- Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.
- If you specify the LINEPRINTER option in the PROC SHEWHART statement, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.

character

specifies a plotting character for charts produced with the LINEPRINTER option. For example, the following statements use an asterisk (*) to plot the points on the $\bar{X}$ and R charts:

```
proc shewhart data=Values lineprinter;
   xrchart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options" lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the XRCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.69 XRCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit on $\bar{X}$ chart
LCLLABEL2=	Specifies label for lower control limit on R chart
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on $\bar{X}$ chart

Table 19.69 *continued*

Option	Description
NDECIMAL2=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on *R* chart
NOCTL	Suppresses display of central line on $\bar{X}$ chart
NOCTL2	Suppresses display of central line on *R* chart
NOLCL	Suppresses display of lower control limit on $\bar{X}$ chart
NOLCL2	Suppresses display of lower control limit on *R* chart
NOLIMIT0	Suppresses display of zero lower control limit on *R* chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit on $\bar{X}$ chart
NOUCL2	Suppresses display of upper control limit on *R* chart
RSYMBOL=	Specifies label for central line on *R* chart
UCLLABEL=	Specifies label for upper control limit on $\bar{X}$ chart
UCLLABEL2=	Specifies label for upper control limit on *R* chart
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line on $\bar{X}$ chart
Process Mean and Standard Deviation Options	
MU0=	Specifies known value of μ_0 for process mean μ
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on $\bar{X}$ chart
ALLLABEL2=	Labels every point on *R* chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits

Table 19.69 *continued*

Option	Description
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits on $\bar{X}$ chart
OUTLABEL2=	Labels points outside control limits on *R* chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes for the $\bar{X}$ chart
TESTS2=	Specifies tests for special causes for the *R* chart
TEST2RESET=	Enables tests for special causes to be reset for the *R* chart
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied to the $\bar{X}$ chart
ZONELABELS	Adds labels A, B, and C to zone lines for $\bar{X}$ chart
ZONE2LABELS	Adds labels A, B, and C to zone lines for *R* chart
ZONES	Adds lines to $\bar{X}$ chart delineating zones A, B, and C
ZONES2	Adds lines to *R* chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels $\bar{X}$ chart zone lines with their values
ZONE2VALUES	Labels *R* zone lines with their values

Table 19.69 *continued*

Option	Description
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	Suppresses vertical axis truncation at zero applied by default to *R* chart
NOVANGLE	Requests vertical axis labels that are strung out vertically

Table 19.69 *continued*

Option	Description
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=	Specifies major tick mark values for vertical axis of R chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of charts
NOCHART2	Suppresses creation of R chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel

Table 19.69 *continued*

Option	Description
SEPARATE	Displays $\bar{X}$ and R charts on separate screens or pages
TOTPANELS=	Specifies number of pages or screens to be used to display chart
YPCT1=	Specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and R charts
ZEROSTD	Displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on R chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on R chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on R chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart

Table 19.69 *continued*

Option	Description
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=	Specifies annotate data set that adds features to *R* chart
DESCRIPTION=	Specifies description of $\bar{X}$ chart's GRSEG catalog entry
DESCRIPTION2=	Specifies description of *R* chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of $\bar{X}$ chart's GRSEG catalog entry
NAME2=	Specifies name of *R* chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases

Table 19.69 *continued*

Option	Description
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers in the $\bar{X}$ chart
MARKERDISPLAY2=	Specifies a subset of subgroups to be plotted with markers in the *R* chart
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers in the $\bar{X}$ chart
MARKERLABEL2=	Specifies labels for subgroups that are plotted with markers in the *R* chart
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart

Table 19.69 *continued*

Option	Description
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend

Table 19.69 *continued*

Option	Description
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments

Table 19.69 *continued*

Option	Description
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*

Table 19.69 *continued*

Option	Description
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: XRCHART Statement

The following sections provide details that are specific to the XRCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Means and Ranges

The following notation is used in this section:

μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	Mean of measurements in ith subgroup
R_i	Range of measurements in ith subgroup
n_i	Sample size of ith subgroup
N	Number of subgroups
$\overline{\overline{X}}$	Weighted average of subgroup means
$d_2(n)$	Expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	Standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	$100 \times p$th percentile of the standard normal distribution
$D_p(n)$	$100 \times p$th percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on the $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the mean plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 14}{5} = 15.2$$

Each point on the R chart indicates the value of a subgroup range (R_i). For example, the range plotted for the tenth subgroup is $R_{10} = 19 - 12 = 7$.

Central Lines

On an $\bar{X}$ chart, by default, the central line indicates an estimate of μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

On an R chart, by default, the central line for the ith subgroup indicates an estimate for the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard errors of $\bar{X}_i$ and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ or R_i exceeds the limits

Table 19.71 provides the formulas for the limits.

Table 19.71 Limits for $\bar{X}$ and R Charts

Control Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$
	UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
R Chart	LCL = lower limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$
	UCL = upper limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$
Probability Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
	UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
R Chart	LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$
	UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

The formulas for R charts assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 19.71. Note that the limits vary with n_i and that the probability limits for R_i are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.

- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.72 lists the variables that are saved.

Table 19.72 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_CP_	Capability index C_p
CPK	Capability index C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLR	Lower control limit for subgroup range
LCLX	Lower control limit for subgroup mean
LIMITN	Nominal sample size associated with the control limits
LSL	Lower specification limit
MEAN	Process mean ($\overline{\overline{X}}$ or μ_0)
_R_	Value of central line on R chart
SIGMAS	Multiple (k) of standard error of $\bar{X}_i$ or R_i
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the XRCHART statement
TARGET	Target value
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	Upper control limit for subgroup range
UCLX	Upper control limit for subgroup mean
USL	Upper specification limit
VAR	*Process* specified in the XRCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, and _UCLR_.
2. If the limits are defined in terms of a multiple k of the standard errors of $\bar{X}_i$ and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.

3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.

4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.

5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XRCHART statement. For an example, see "Saving Control Limits" on page 1889.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name containing 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup means, ranges, and sample sizes are created for each *process* specified in the XRCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   xrchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthR, WidthN, DiameterX, DiameterR, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1888.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.73 lists the variables that are saved.

Table 19.73 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on $\bar{X}$ chart
EXLIMR	Control limit exceeded on R chart
LCLR	Lower control limit for range
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
_R_	Average range
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample size
SUBR	Subgroup range
SUBX	Subgroup mean
TESTS	Tests for special causes signaled on $\bar{X}$ chart
TESTS2	Tests for special causes signaled on R chart
UCLR	Upper control limit for range
UCLX	Upper control limit for mean
VAR	*Process* specified in the XRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variable _TESTS2_ is saved if you specify the TESTS2= option.

4. The variables _EXLIM_, _EXLIMR_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1889.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XRCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XRCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating Charts for Means and Ranges from Raw Data" on page 1882.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   xrchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly by using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, _UCLX_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 19.71

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1892.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedure or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the XRCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup range, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *X*, *R*, and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   xrchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightR, WeightN, YieldstrengthX, YieldstrengthR, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means and Ranges from Summary Data" on page 1885.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to read data sets created by other SAS procedures. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.74 lists the variables required in a TABLE= data set used with the XRCHART statement.

Table 19.74 Variables Required in a TABLE= Data Set

Variable	Description
LCLR	Lower control limit for range
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
_R_	Average range
Subgroup-variable	Values of the *subgroup-variable*

Table 19.74 *continued*

Variable	Description
SUBN	Subgroup sample size
SUBR	Subgroup range
SUBX	Subgroup mean
UCLR	Upper control limit for range
UCLX	Upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup means and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup ranges and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1889.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and MVGRANGE) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVGRANGE Method

If you specify SMETHOD=MVGRANGE, σ is estimated by using a moving range of subgroup averages. This is appropriate for constructing control charts for means when the *j*th measurement in the *i*th subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}.

The estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive subgroup averages used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the subgroup averages are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is appropriate for constructing the three-way control chart that is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or *within-group* variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2152. A three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means, and a chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XRCHART statement produces the appropriate charts of subgroup means and subgroup ranges.

Examples: XRCHART Statement

This section provides advanced examples that use the XRCHART statement.

Example 19.38: Applying Tests for Special Causes

NOTE: See *Mean and Range Charts-Tests for Special Causes* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make $\bar{X}$ and R charts more sensitive to special causes of variation.

The weight of a roll of tape is measured before and after an adhesive is applied. The difference in weight represents the amount of adhesive applied to the tape during the coating process. The following data set contains the average and the range of the adhesive amounts for 21 samples of five rolls:

```
data Tape;
   input Sample $ WeightX WeightR;
   WeightN=5;
   label WeightX = 'Average Adhesive Amount'
         Sample  = 'Sample Code';
   datalines;
C9  1270  35
C4  1258  25
A7  1248  24
A1  1260  39
A5  1273  29
D3  1260  21
D6  1259  37
D1  1240  37
R4  1260  28
H7  1255  19
H2  1268  36
H6  1253  36
P4  1273  29
P9  1275  22
J7  1257  24
J2  1269  41
J3  1249  36
B2  1264  31
G4  1258  25
G6  1248  36
G3  1248  30
;
```

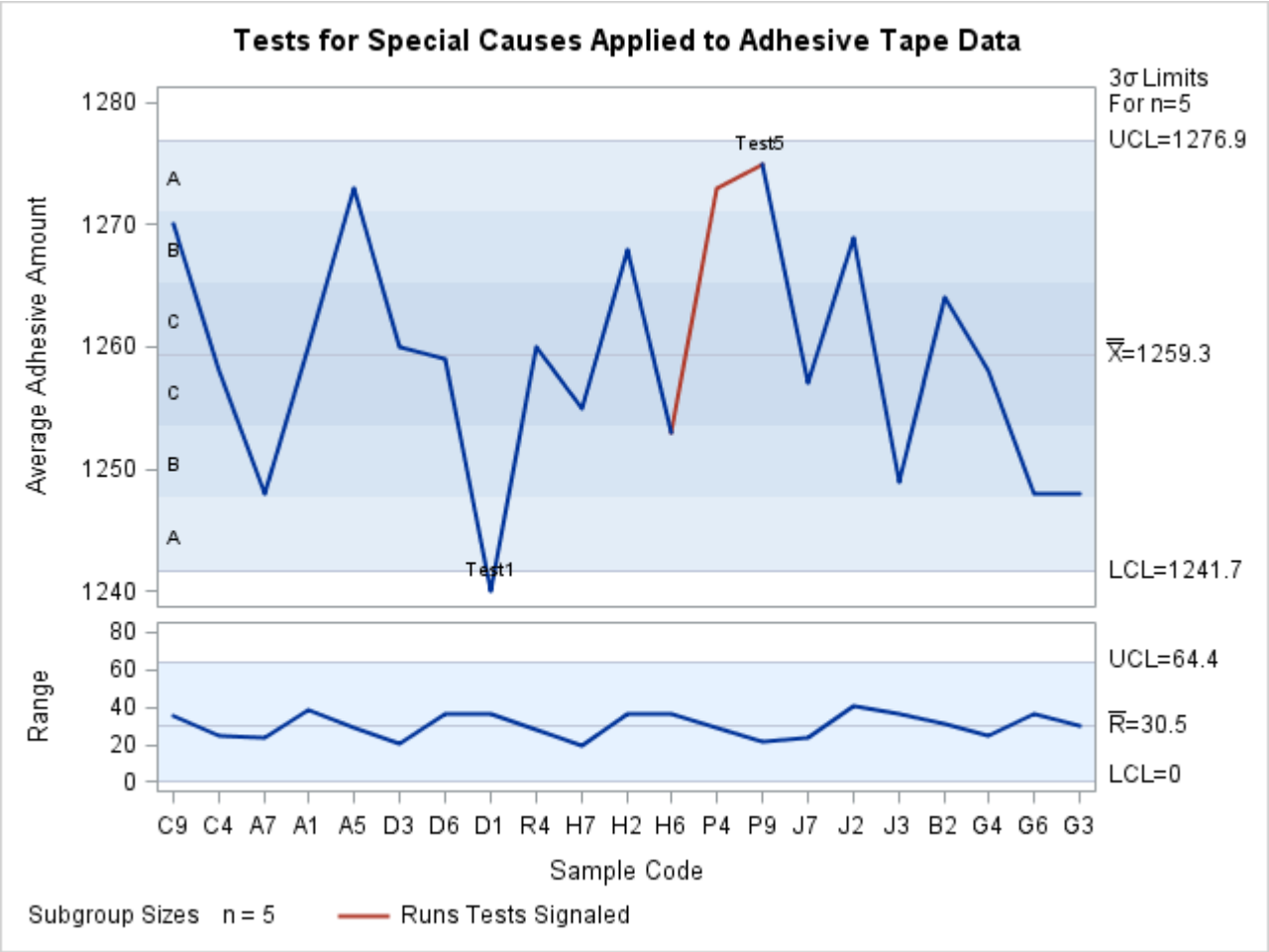
The following statements create $\bar{X}$ and R charts, apply several tests to the $\bar{X}$ chart, and tabulate the results:

```
title 'Tests for Special Causes Applied to Adhesive Tape Data';
ods graphics on;
proc shewhart history=Tape;
   xrchart Weight*Sample / tests    = 1 to 5
                           odstitle = title
                           tabletests
                           zonelabels;
run;
```

The charts are shown in Output 19.38.1, and the table is shown in Output 19.38.2. The TESTS= option requests Tests 1, 2, 3, 4, and 5, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2119. The TABLETESTS option requests a basic table of subgroup statistics and control limits with a column indicating which subgroups tested positive for special causes.

The ZONELABELS option displays zone lines and zone labels on the $\bar{X}$ chart. The zones are used to define the tests.

Output 19.38.1 Tests for Special Causes Displayed on $\bar{X}$ and R Charts



Output 19.38.1 and Output 19.38.2 indicate that Test 1 is positive at sample D1 and Test 5 is positive at sample P9. Test 1 detects one point beyond Zone A (outside the control limits), and Test 5 detects two out of three points in a row in Zone A or beyond.

Output 19.38.2 Tabular Form of $\bar{X}$ and R Charts

Tests for Special Causes Applied to Adhesive Tape Data

The SHEWHART Procedure

Means and Ranges Chart Summary for Weight

		3 Sigma Limits with n=5 for Mean				3 Sigma Limits with n=5 for Range		
Sample	Subgroup Sample Size	Lower Limit	Subgroup Mean	Upper Limit	Special Tests Signaled	Lower Limit	Subgroup Range	Upper Limit
C9	5	1241.7065	1270.0000	1276.8650		0	35.000000	64.441879
C4	5	1241.7065	1258.0000	1276.8650		0	25.000000	64.441879
A7	5	1241.7065	1248.0000	1276.8650		0	24.000000	64.441879
A1	5	1241.7065	1260.0000	1276.8650		0	39.000000	64.441879
A5	5	1241.7065	1273.0000	1276.8650		0	29.000000	64.441879
D3	5	1241.7065	1260.0000	1276.8650		0	21.000000	64.441879
D6	5	1241.7065	1259.0000	1276.8650		0	37.000000	64.441879
D1	5	1241.7065	1240.0000	1276.8650	1	0	37.000000	64.441879
R4	5	1241.7065	1260.0000	1276.8650		0	28.000000	64.441879
H7	5	1241.7065	1255.0000	1276.8650		0	19.000000	64.441879
H2	5	1241.7065	1268.0000	1276.8650		0	36.000000	64.441879
H6	5	1241.7065	1253.0000	1276.8650		0	36.000000	64.441879
P4	5	1241.7065	1273.0000	1276.8650		0	29.000000	64.441879
P9	5	1241.7065	1275.0000	1276.8650	5	0	22.000000	64.441879
J7	5	1241.7065	1257.0000	1276.8650		0	24.000000	64.441879
J2	5	1241.7065	1269.0000	1276.8650		0	41.000000	64.441879
J3	5	1241.7065	1249.0000	1276.8650		0	36.000000	64.441879
B2	5	1241.7065	1264.0000	1276.8650		0	31.000000	64.441879
G4	5	1241.7065	1258.0000	1276.8650		0	25.000000	64.441879
G6	5	1241.7065	1248.0000	1276.8650		0	36.000000	64.441879
G3	5	1241.7065	1248.0000	1276.8650		0	30.000000	64.441879

Example 19.39: Specifying Standard Values for the Process Mean and Standard Deviation

NOTE: See *X-bar and R CHARTS-Specifying Standard Values* in the SAS/QC Sample Library.

By default, the XRCHART statement estimates the process mean (μ) and standard deviation (σ) from the data, as in the previous example. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

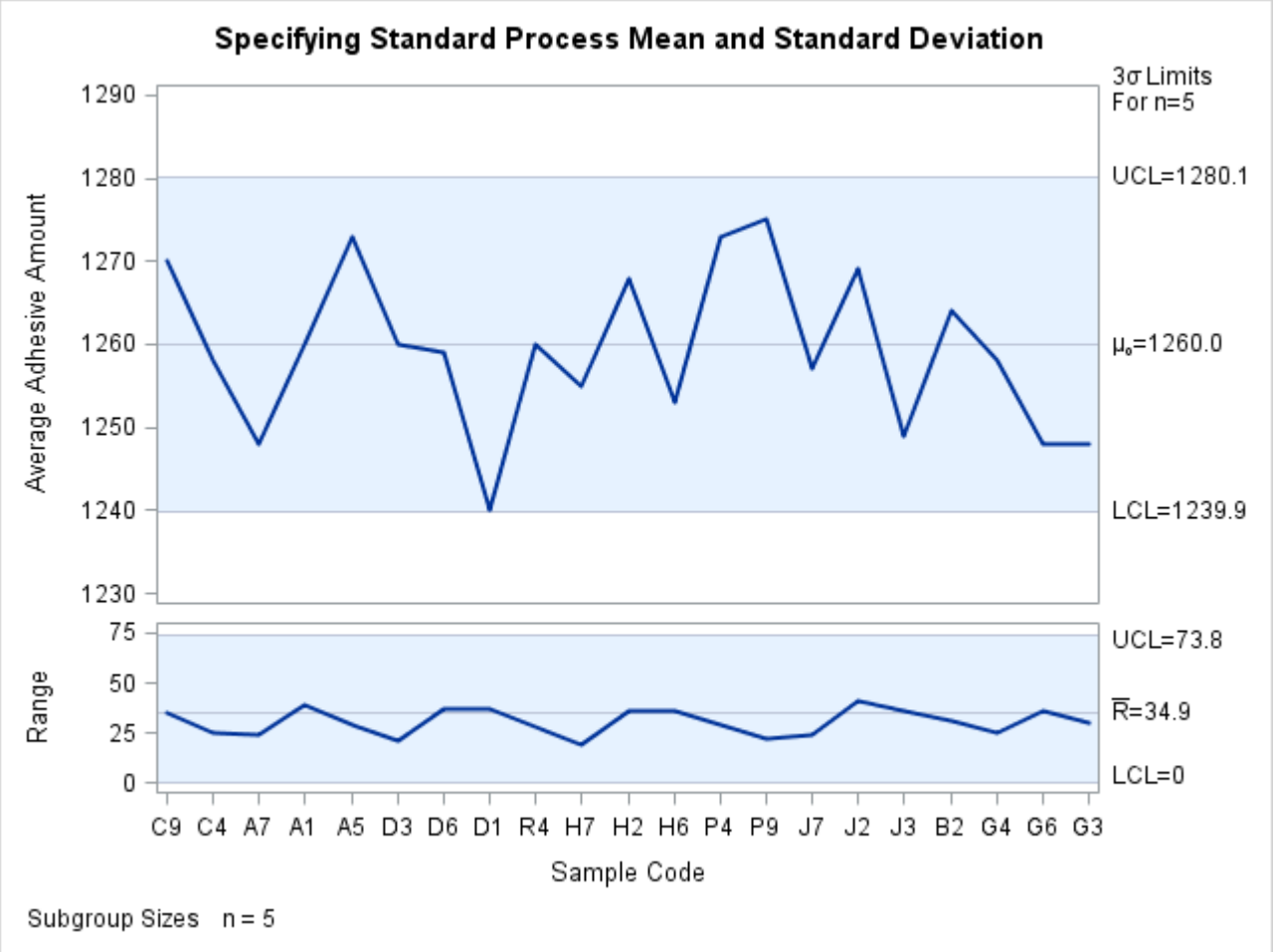
For example, suppose it is known that the adhesive coating process introduced in the previous example has a mean of 1260 and standard deviation of 15. The following statements specify these standard values:

```
ods graphics on;
title 'Specifying Standard Process Mean and Standard Deviation';
proc shewhart history=Tape;
   xrchart Weight*Sample / mu0      = 1260
```

```
                                    sigma0   = 15
                                    xsymbol  = mu0
                                    odstitle = title;
run;
```

The XSYMBOL= option specifies the label for the central line on the $\bar{X}$ chart. The resulting $\bar{X}$ and R charts are shown in Output 19.39.1.

Output 19.39.1 Specifying Standard Values with MU0= and SIGMA0=



The central lines and control limits for both charts are determined by using μ_0 and σ_0 (see the equations in Table 19.71). Output 19.39.1 indicates that the process is in statistical control.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data Tapelim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Weight';
   _subgrp_ = 'Sample';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 1260;
```

```
   _stddev_ = 15;

proc shewhart history=Tape limits=Tapelim;
   xrchart Weight*Sample / xsymbol=mu0;
run;
```

The variables _VAR_ and _SUBGRP_ are required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the XRCHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting charts (not shown here) are identical to those shown in Output 19.39.1.

Example 19.40: Working with Unequal Subgroup Sample Sizes

NOTE: See *X-bar and R Charts with Varying Sample Sizes* in the SAS/QC Sample Library.

The following data set (Wire) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

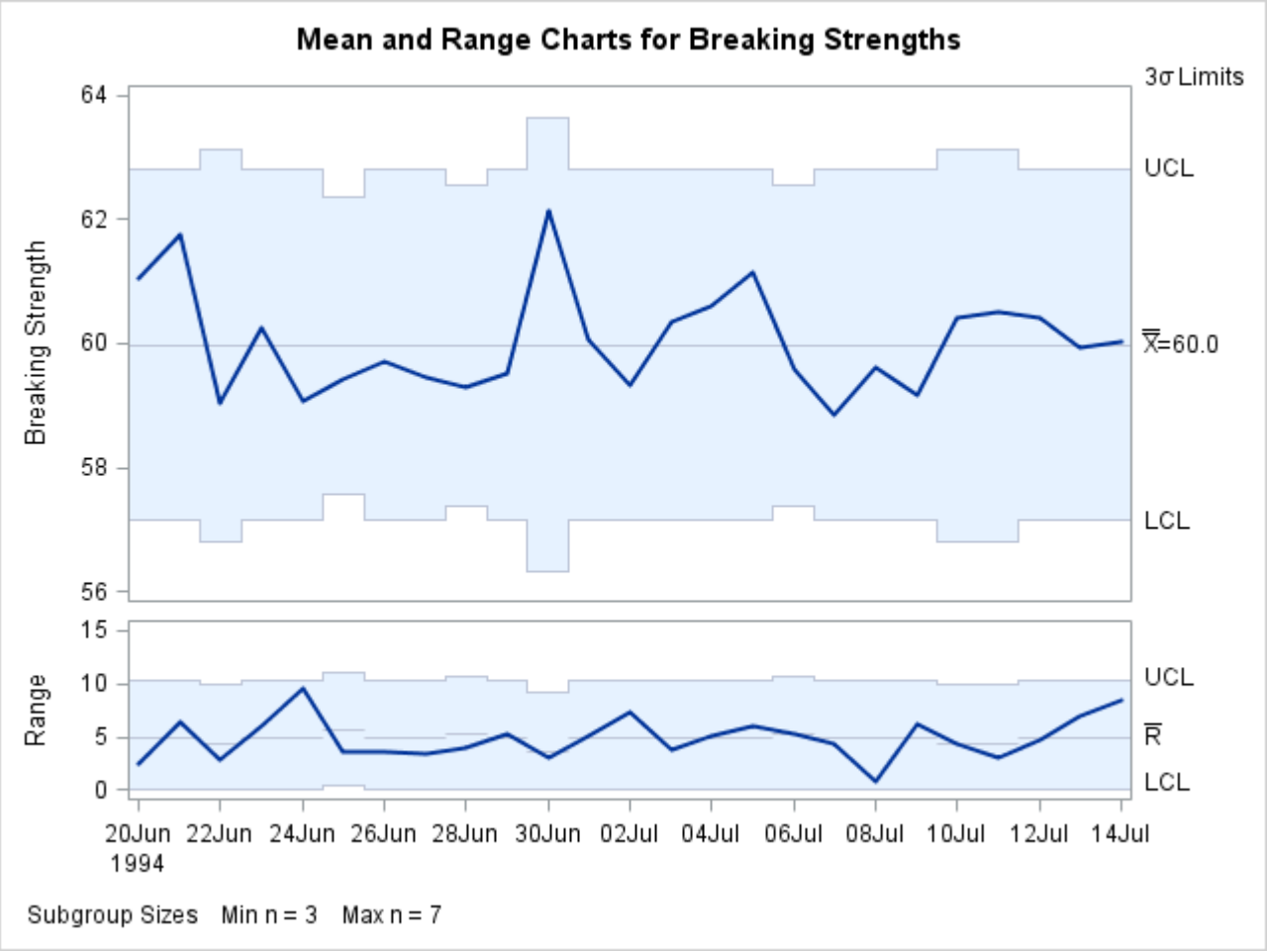
```
data Wire;
   input Day size @;
   informat Day date7.;
   format Day date7.;
   do i=1 to size;
      input Breakstrength @@;
      output;
   end;
   drop i size;
   label Breakstrength = 'Breaking Strength';
   datalines;
20JUN94 5 60.6 62.3 62.0 60.4 59.9
21JUN94 5 61.9 62.1 60.6 58.9 65.3
22JUN94 4 57.8 60.5 60.1 57.7
23JUN94 5 56.8 62.5 60.1 62.9 58.9
24JUN94 5 63.0 60.7 57.2 61.0 53.5
25JUN94 7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
26JUN94 5 59.3 61.7 59.1 58.1 60.3
27JUN94 5 61.3 58.5 57.8 61.0 58.6
28JUN94 6 59.5 58.3 57.5 59.4 61.5 59.6
29JUN94 5 61.7 60.7 57.2 56.5 61.5
30JUN94 3 63.9 61.6 60.9
01JUL94 5 58.7 61.4 62.4 57.3 60.5
02JUL94 5 56.8 58.5 55.7 63.0 62.7
03JUL94 5 62.1 60.6 62.1 58.7 58.3
04JUL94 5 59.1 60.4 60.4 59.0 64.1
05JUL94 5 59.9 58.8 59.2 63.0 64.9
06JUL94 6 58.8 62.4 59.4 57.1 61.2 58.6
07JUL94 5 60.3 58.7 60.5 58.6 56.2
08JUL94 5 59.2 59.8 59.7 59.3 60.0
09JUL94 5 62.3 56.0 57.0 61.8 58.8
10JUL94 4 60.5 62.0 61.4 57.7
11JUL94 4 59.3 62.4 60.4 60.0
```

```
12JUL94 5 62.4 61.3 60.5 57.7 60.2
13JUL94 5 61.2 55.5 60.2 60.4 62.4
14JUL94 5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request $\bar{X}$ and R charts, shown in Output 19.40.1, for the strength measurements:

```
ods graphics on;
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=Wire;
   xrchart Breakstrength*Day / nohlabel
                               odstitle = title;
run;
```

Output 19.40.1 $\bar{X}$ and R Charts with Varying Subgroup Sample Sizes



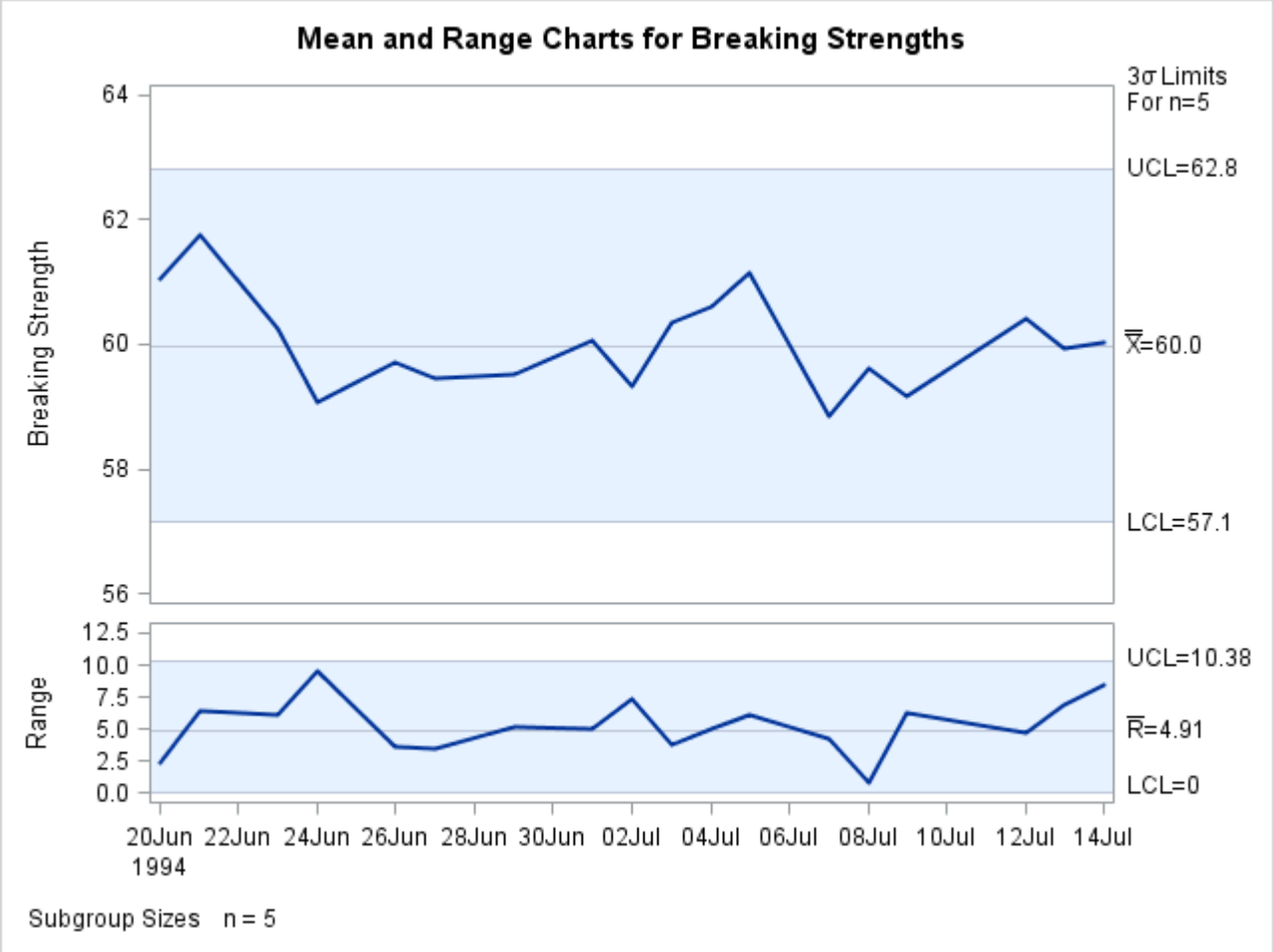
Note that the central line on the R chart and the control limits on both charts vary with the subgroup sample size. The sample size legend in the lower-left corner displays the minimum and maximum subgroup sample sizes.

The XRCHART statement provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing control limits, as illustrated by the following statements:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=Wire;
   xrchart Breakstrength*Day / nohlabel
                               odstitle = title
                               limitn   = 5;
run;
```

The resulting charts are shown in Output 19.40.2.

Output 19.40.2 Control Limits Based on Fixed Sample Size

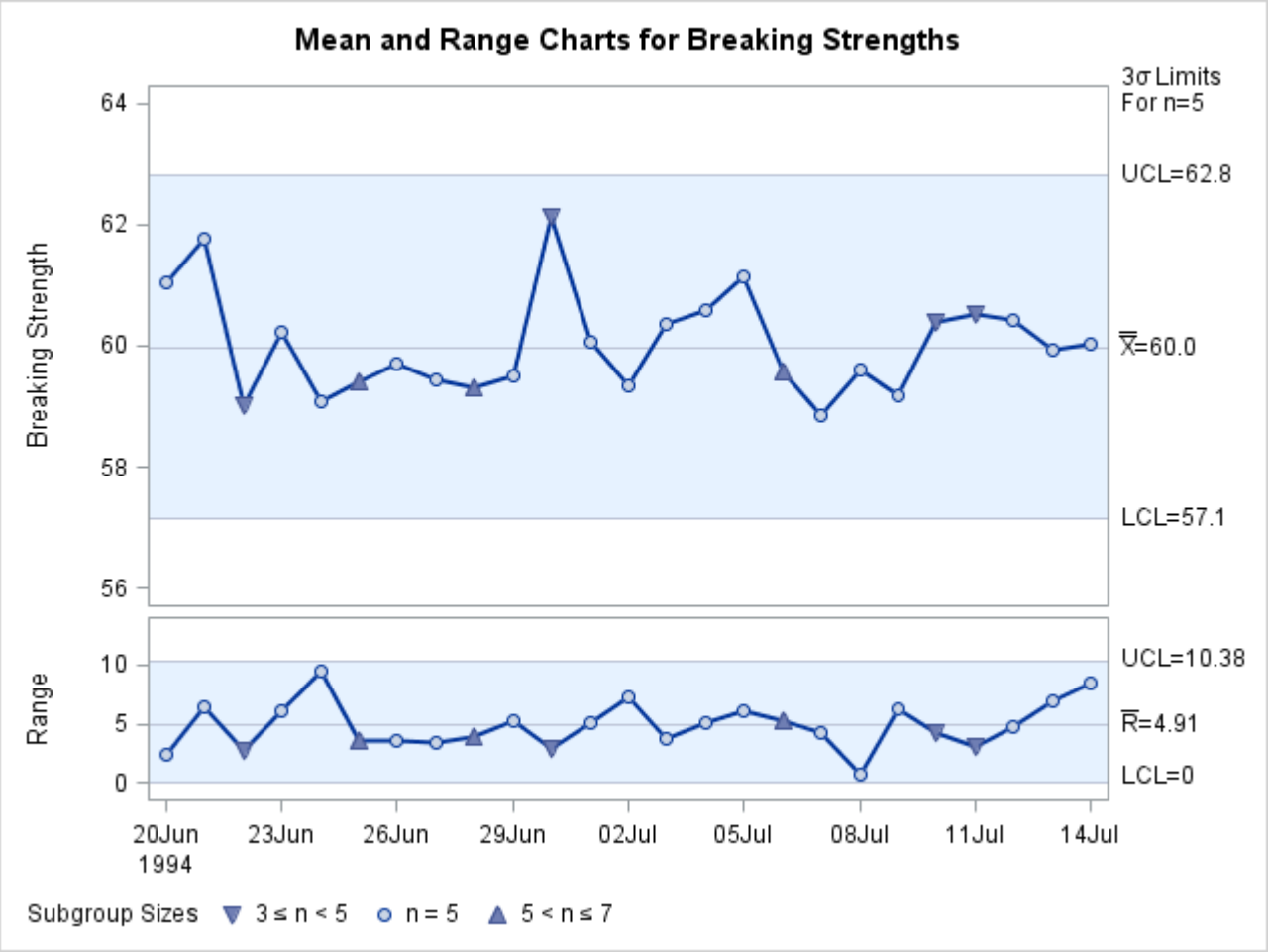


Note that the only points displayed on the chart are those corresponding to subgroups whose sample sizes match the nominal sample size of five. To plot points for all subgroups (regardless of subgroup sample size), you can specify the ALLN option, as follows:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=Wire;
   xrchart Breakstrength*Day / nohlabel
                               odstitle = title
                               limitn   = 5
                               alln
                               nmarkers;
run;
```

The charts are shown in Output 19.40.3. The NMARKERS option requests special symbols to identify points for which the subgroup sample size differs from the nominal sample size.

Output 19.40.3 Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes $\hat{\sigma}$ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests an estimate that assigns greater weight to estimates of σ from subgroups with larger sample sizes. For more information, see "Methods for Estimating the Standard Deviation" on page 1915.

The following statements apply both methods:

```
proc shewhart data=Wire;
   xrchart Breakstrength*Day / outlimits = Wlim1
                               outindex  = 'Default'
                               nochart;
   xrchart Breakstrength*Day / smethod   = mvlue
                               outlimits = Wlim2
                               outindex  = 'MVLUE'
                               nochart;
run;
```

```
data Wlimits;
   set Wlim1 Wlim2;
run;
```

The data set Wlimits is listed in Output 19.40.4.

Output 19.40.4 Listing of the Data Set Wlimits

The WLIMITS Data Set

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
Breakstrength	Day	Default	ESTIMATE	V	.002699796	3	V
Breakstrength	Day	MVLUE	ESTIMATE	V	.002699796	3	V

MEAN	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
59.9766	V	V	V	V	2.11146
59.9766	V	V	V	V	2.11240

The variables in an OUTLIMITS= data set whose values vary with subgroup sample size are assigned the special missing value *V*. Consequently, the control limit variables (_LCLX_, _UCLX_, _LCLR_, and _UCLR_), as well as the variables _R_ and _LIMITN_, have this value.

XSCHART Statement: SHEWHART Procedure

Overview: XSCHART Statement

The XSCHART statement creates $\bar{X}$ and s charts for subgroup means and standard deviations, which are used to analyze the central tendency and variability of a process.

You can use options in the XSCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means and standard deviations or as probability limits
- tabulate subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits

- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing $\bar{X}$ and s charts with the XSCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 4, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: XSCHART Statement

This section introduces the XSCHART statement with simple examples that illustrate commonly used options. Complete syntax for the XSCHART statement is presented in the section "Syntax: XSCHART Statement" on page 1936, and advanced examples are given in the section "Examples: XSCHART Statement" on page 1959.

Creating Charts for Means and Standard Deviations from Raw Data

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

A petroleum company uses a turbine to heat water into steam, which is then pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set named Turbine, which contains the power output measurements for 20 days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
   end;
   drop i;
```

```
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711

   ... more lines ...

23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
```

A partial listing of Turbine is shown in Figure 19.114.

Figure 19.114 Partial Listing of the Data Set Turbine

Kilowatt Power Output Data

Obs	Day	KWatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
5	04JUL	3583
6	04JUL	3617
7	04JUL	3789
8	04JUL	3180
9	04JUL	3505
10	04JUL	3454
11	04JUL	3417
12	04JUL	3199
13	04JUL	3613
14	04JUL	3384
15	04JUL	3475
16	04JUL	3316
17	04JUL	3556
18	04JUL	3607
19	04JUL	3364
20	04JUL	3721
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
24	05JUL	3193
25	05JUL	3635

The data set is said to be in "strung-out" form because each observation contains the day and power output for a single heating. The first 20 observations contain the power outputs for the first day, the second 20 observations contain the power outputs for the second day, and so on. Because the variable Day classifies the

observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWatts contains the power output measurements and is referred to as the *process variable* (or *process* for short).

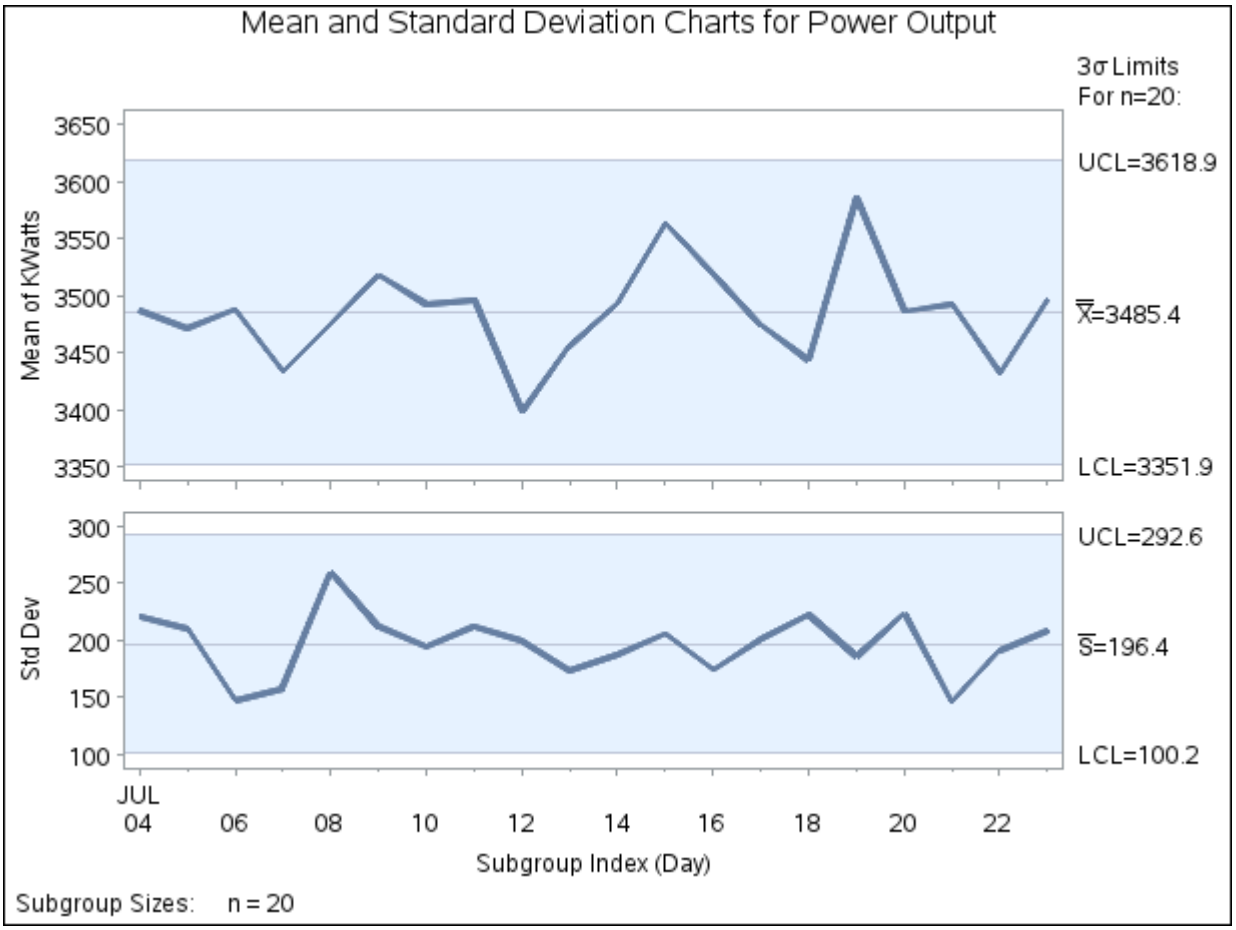
You can use $\bar{X}$ and s charts to determine whether the heating process is in control. The following statements create the $\bar{X}$ and s charts shown in Figure 19.115:

```
ods graphics off;
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart data=Turbine;
   xschart KWatts*Day ;
run;
```

This example illustrates the basic form of the XSCHART statement. After the keyword XSCHART, you specify the *process* to analyze (in this case KWatts), followed by an asterisk and the *subgroup-variable* (Day).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 19.115 $\bar{X}$ and s Charts for Power Output Data (Traditional Graphics)



Each point on the $\bar{X}$ chart represents the mean of the measurements for a particular day. For instance, the mean plotted for the first day is $(3196 + 3507 + \cdots + 3721)/20 = 3487.4$.

Each point on the *s* chart represents the standard deviation of the measurements for a particular day. For instance, the standard deviation plotted for the first day is

$$\sqrt{\frac{(3196-3487.4)^2+(3507-3487.4)^2+\cdots+(3721-3487.4)^2}{19}}=220.26$$

Because all the points lie within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 19.77. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1935.

For computational details, see "Constructing Charts for Means and Standard Deviations" on page 1949. For more details on reading raw data, see "DATA= Data Set" on page 1954.

Creating Charts for Means and Standard Deviations from Summary Data

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create $\bar{X}$ and *s* charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the XSCHART statement with data of this type.

The following data set (Oilsum) provides the data from the preceding example in summarized form:

```
data Oilsum;
   input Day KWattsX KWattsS KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day='Date of Measurement';
   datalines;
04JUL94 3487.40 220.260 20
05JUL94 3471.65 210.427 20
06JUL94 3488.30 147.025 20
07JUL94 3434.20 157.637 20
08JUL94 3475.80 258.949 20
09JUL94 3518.10 211.566 20
10JUL94 3492.65 193.779 20
11JUL94 3496.40 212.024 20
12JUL94 3398.50 199.201 20
13JUL94 3456.05 173.455 20
14JUL94 3493.60 187.465 20
15JUL94 3563.30 205.472 20
16JUL94 3519.05 173.676 20
17JUL94 3474.20 200.576 20
18JUL94 3443.60 222.084 20
19JUL94 3586.35 185.724 20
20JUL94 3486.45 223.474 20
21JUL94 3492.90 145.267 20
22JUL94 3432.80 190.994 20
23JUL94 3496.90 208.858 20
;
```

A partial listing of Oilsum is shown in Figure 19.116.

Figure 19.116 The Summary Data Set Oilsum

Summary Data Set for Power Output

Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

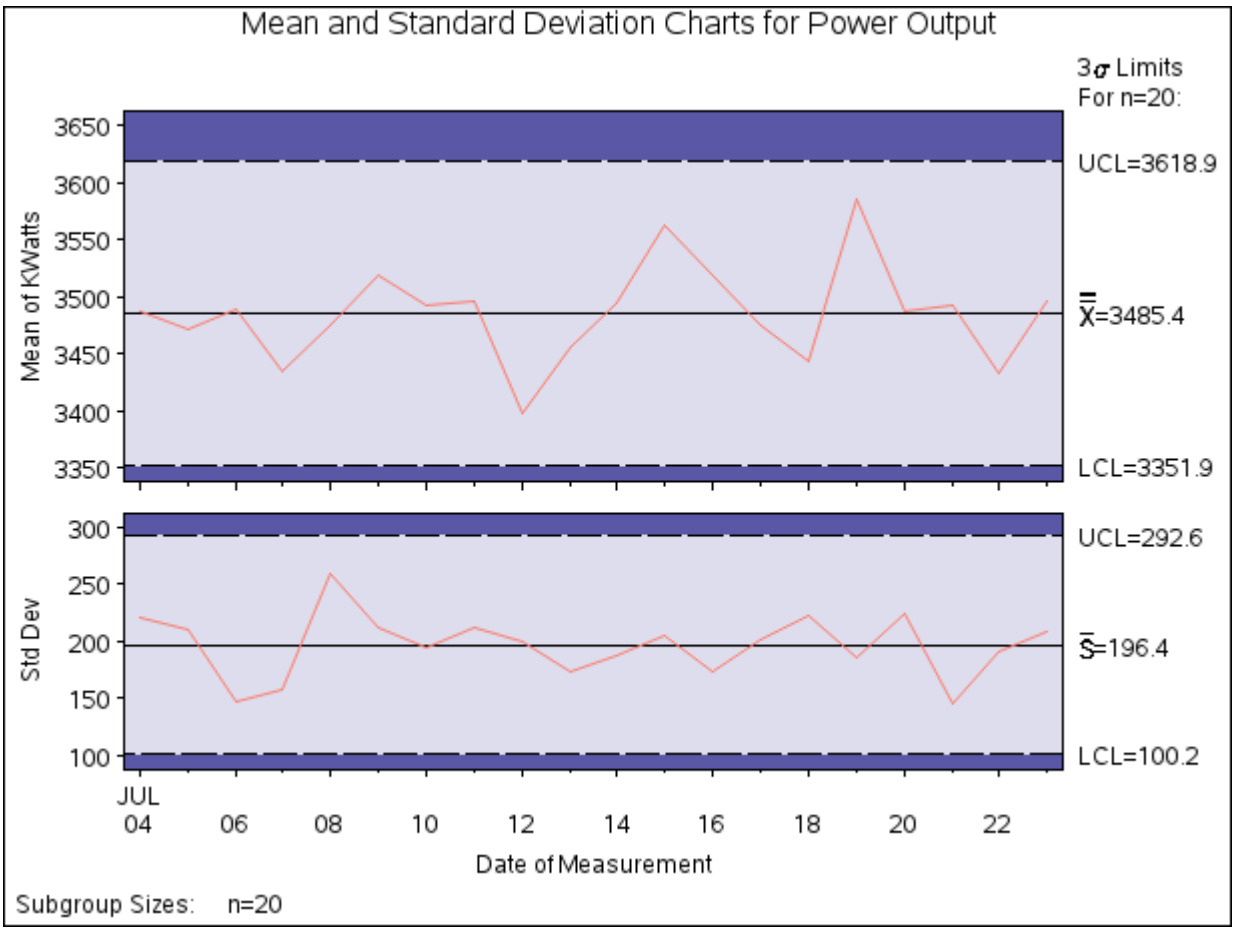
There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day). The variable KWattsX contains the subgroup means, the variable KWattsS contains the subgroup standard deviations, and the variable KWattsN contains the subgroup sample sizes (which are all 20). You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext='albany amt';
symbol color = salmon h = .8;
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart history=Oilsum;
   xschart KWatts*Day / cframe  = lib
                        cinfill = bwh
                        cconnect = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and XSCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting $\bar{X}$ and s charts are shown in Figure 19.117.

Note that KWatts is *not* the name of a SAS variable in the data set Oilsum but is, instead, the common prefix for the names of the three SAS variables KWattsX, KWattsS, and KWattsN. The suffix characters *X*, *S*, and *N* indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (KWatts), which is referred to as the *process*. The name Day specified after the asterisk is the name of the *subgroup-variable*.

Figure 19.117 $\bar{X}$ and s Charts for Power Output Data (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the XSCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the XSCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. For an illustration, see Example 19.42.

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.

- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1955.

Saving Summary Statistics

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

In this example, the XSCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1926) and create a summary data set named Turbhist:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / outhistory = Turbhist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to those in Figure 19.115. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the XSCHART statement. A complete list of options is presented in the section "Syntax: XSCHART Statement" on page 1936.

Figure 19.118 contains a partial listing of Turbhist.

Figure 19.118 The Summary Data Set Turbhist

Summary Data Set for Power Output

Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

There are four variables in the data set Turbhist.

- Day contains the subgroup index.
- KWattsX contains the subgroup means.
- KWattsS contains the subgroup standard deviations.
- KWattsN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *process* KWatts specified in the XSCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1952.

Saving Control Limits

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

You can save the control limits for $\bar{X}$ and s charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1935) or modify the limits with a DATA step program.

The following statements read measurements from the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1926) and save the control limits displayed in Figure 19.115 in a data set named Turblim:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / outlimits=Turblim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Turblim is listed in Figure 19.119.

Figure 19.119 The Data Set Turblim Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	.002699796	3	3351.92	3485.41	3618.90

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

The data set Turblim contains one observation with the limits for *process* KWatts. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart, and the variables _LCLS_ and _UCLS_ contain the lower and upper control limits for the s chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart, and the variable _S_ contains the central line for the s chart. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1951.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / outtable=Turbtab
                        nochart;
run;
```

The data set Turbtab is listed in Figure 19.120.

Figure 19.120 The OUTTABLE= Data Set Turbtab

Summary Statistics and Control Limit Information

VAR	Day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
KWatts	04JUL	3	20	20	3351.92	3487.40	3485.41	3618.90	198.996
KWatts	05JUL	3	20	20	3351.92	3471.65	3485.41	3618.90	198.996
KWatts	06JUL	3	20	20	3351.92	3488.30	3485.41	3618.90	198.996
KWatts	07JUL	3	20	20	3351.92	3434.20	3485.41	3618.90	198.996
KWatts	08JUL	3	20	20	3351.92	3475.80	3485.41	3618.90	198.996
KWatts	09JUL	3	20	20	3351.92	3518.10	3485.41	3618.90	198.996
KWatts	10JUL	3	20	20	3351.92	3492.65	3485.41	3618.90	198.996
KWatts	11JUL	3	20	20	3351.92	3496.40	3485.41	3618.90	198.996
KWatts	12JUL	3	20	20	3351.92	3398.50	3485.41	3618.90	198.996
KWatts	13JUL	3	20	20	3351.92	3456.05	3485.41	3618.90	198.996
KWatts	14JUL	3	20	20	3351.92	3493.60	3485.41	3618.90	198.996
KWatts	15JUL	3	20	20	3351.92	3563.30	3485.41	3618.90	198.996
KWatts	16JUL	3	20	20	3351.92	3519.05	3485.41	3618.90	198.996
KWatts	17JUL	3	20	20	3351.92	3474.20	3485.41	3618.90	198.996
KWatts	18JUL	3	20	20	3351.92	3443.60	3485.41	3618.90	198.996
KWatts	19JUL	3	20	20	3351.92	3586.35	3485.41	3618.90	198.996
KWatts	20JUL	3	20	20	3351.92	3486.45	3485.41	3618.90	198.996
KWatts	21JUL	3	20	20	3351.92	3492.90	3485.41	3618.90	198.996
KWatts	22JUL	3	20	20	3351.92	3432.80	3485.41	3618.90	198.996
KWatts	23JUL	3	20	20	3351.92	3496.90	3485.41	3618.90	198.996

EXLIM	_LCLS_	_SUBS_	_S_	_UCLS_	_EXLIMS_
	100.207	220.260	196.396	292.584	
	100.207	210.427	196.396	292.584	
	100.207	147.025	196.396	292.584	
	100.207	157.637	196.396	292.584	
	100.207	258.949	196.396	292.584	
	100.207	211.566	196.396	292.584	
	100.207	193.779	196.396	292.584	
	100.207	212.024	196.396	292.584	
	100.207	199.201	196.396	292.584	
	100.207	173.455	196.396	292.584	
	100.207	187.465	196.396	292.584	
	100.207	205.472	196.396	292.584	
	100.207	173.676	196.396	292.584	
	100.207	200.576	196.396	292.584	
	100.207	222.084	196.396	292.584	
	100.207	185.724	196.396	292.584	
	100.207	223.474	196.396	292.584	
	100.207	145.267	196.396	292.584	
	100.207	190.994	196.396	292.584	
	100.207	208.858	196.396	292.584	

The data set Turbtab contains one observation for each subgroup sample. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits for the *s* chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart. The variable _S_ contains the central line for the *s* chart. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1953.

A data set created with the OUTTABLE= option can be read later as a TABLE= data set. For example, the following statements read Turbtab and display charts (not shown here) identical to those in Figure 19.115:

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart table=Turbtab;
   xschart KWatts*Day;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2143). For more information, see "TABLE= Data Set" on page 1956.

Reading Preestablished Control Limits

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Turblim saved control limits computed from the measurements in Turbine. This example shows how these limits can be applied to new data. The following statements create $\bar{X}$ and *s* charts for new measurements in a data set named Turbine2 (not listed here) using the control limits in Turblim:

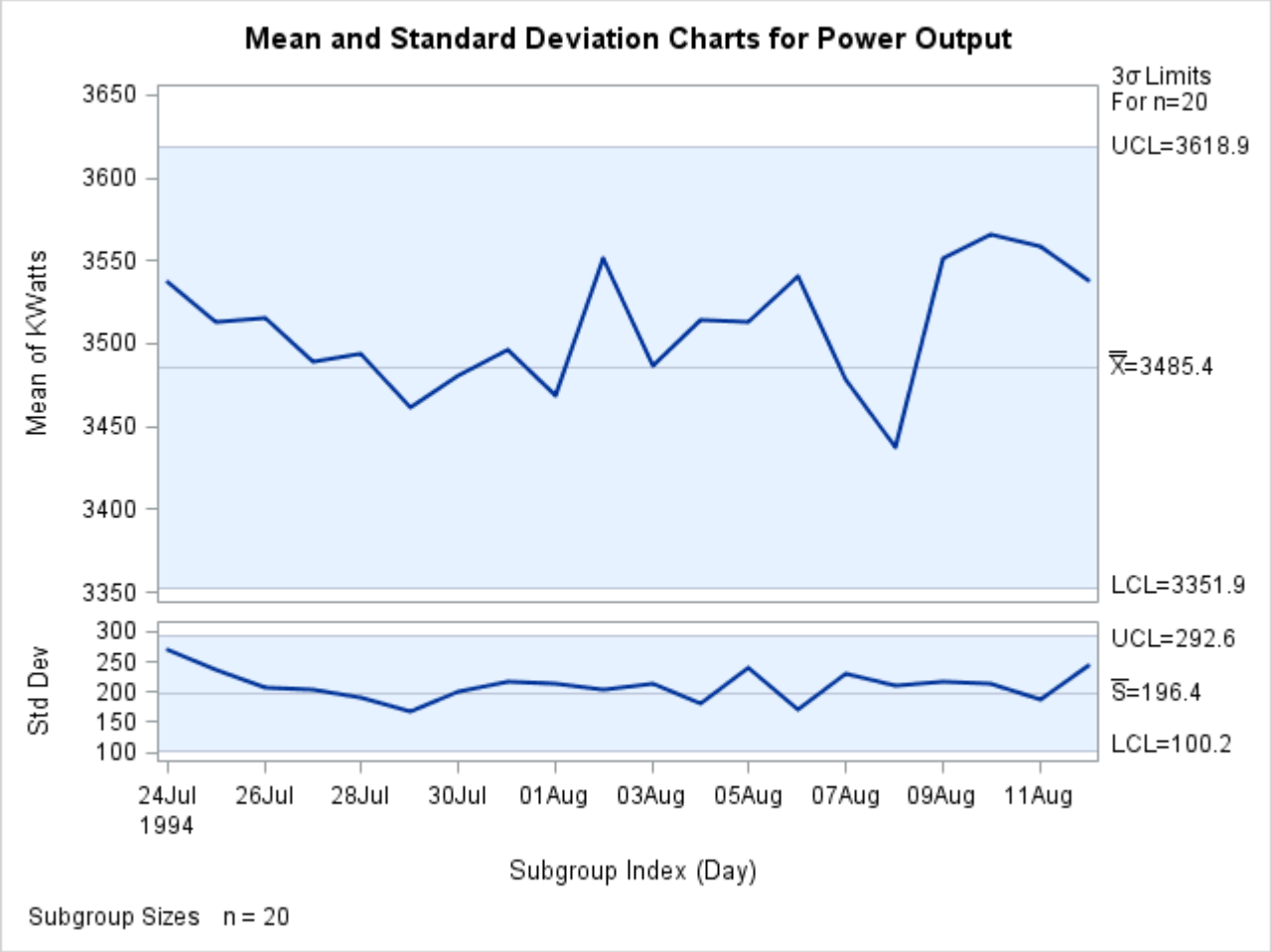
```
ods graphics on;
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart data=Turbine2 limits=Turblim;
   xschart KWatts*Day / odstitle = title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the $\bar{X}$ and *s* charts are created by using ODS Graphics instead of traditional graphics. The charts are shown in Figure 19.121.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing preestablished control limit information. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWatts
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

Figure 19.121 $\bar{X}$ and s Charts for Second Set of Power Outputs (ODS Graphics)



The means and standard deviations lie within the control limits, indicating that the heating process is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1955 for details concerning the variables that you must provide.

Syntax: XSCHART Statement

The basic syntax for the XSCHART statement is as follows:

XSCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

XSCHART *processes* ∗ *subgroup-variable* < **(***block-variables* | **=***'character'*> / < *options* > **;**

You can use any number of XSCHART statements in the SHEWHART procedure. The components of the XSCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If the raw data are read using a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1926.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means and Standard Deviations from Summary Data" on page 1929.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1933.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ and s charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   xschart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XSCHART statement, Day is the subgroup variable. For details, see the section "Subgroup Variables" on page 1970.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2074 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means and standard deviations.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2073 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create $\bar{X}$ and s charts using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   xschart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options" on page 1938 lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1993 describes each option in detail.

Summary of Options

The following tables list the XSCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1993.

Table 19.75 XSCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	Requests probability limits for chart
LIMITN=	Specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	Computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	Reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	Reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	Specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	Specifies color for area inside control limits
CLIMITS=	Specifies color of control limits, central line, and related labels
LCLLABEL=	Specifies label for lower control limit on $\bar{X}$ chart
LCLLABEL2=	Specifies label for lower control limit on s chart
LIMLABSUBCHAR=	Specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	Specifies line type for control limits

Table 19.75 *continued*

Option	Description
NDECIMAL=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on $\bar{X}$ chart
NDECIMAL2=	Specifies number of digits to right of decimal place in default Labels for control limits and central line on s chart
NOCTL	Suppresses display of central line on $\bar{X}$ chart
NOCTL2	Suppresses display of central line on s chart
NOLCL	Suppresses display of lower control limit on $\bar{X}$ chart
NOLCL2	Suppresses display of lower control limit on s chart
NOLIMIT0	Suppresses display of zero lower control limit on s chart
NOLIMITLABEL	Suppresses labels for control limits and central line
NOLIMITS	Suppresses display of control limits
NOLIMITSFRAME	Suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	Suppresses legend for control limits
NOUCL	Suppresses display of upper control limit on $\bar{X}$ chart
NOUCL2	Suppresses display of upper control limit on s chart
SSYMBOL=	Specifies label for central line on s chart
UCLLABEL=	Specifies label for upper control limit on $\bar{X}$ chart
UCLLABEL2=	Specifies label for upper control limit on s chart
WLIMITS=	Specifies width for control limits and central line
XSYMBOL=	Specifies label for central line on $\bar{X}$ chart
Process Mean and Standard Deviation Options	
MU0=	Specifies known value of μ_0 for process mean μ
SIGMA0=	Specifies known value σ_0 for process standard deviation σ
SMETHOD=	Specifies method for estimating process standard deviation σ
TYPE=	Identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	Labels every point on $\bar{X}$ chart
ALLLABEL2=	Labels every point on s chart
CLABEL=	Specifies color for labels
CCONNECT=	Specifies color for line segments that connect points on chart
CFRAMELAB=	Specifies fill color for frame around labeled points
CNEEDLES=	Specifies color for needles that connect points to central line

Table 19.75 *continued*

Option	Description
COUT=	Specifies color for portions of line segments that connect points outside control limits
COUTFILL=	Specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	Specifies angle at which labels are drawn
LABELFONT=	Specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	Specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	Connects points to central line with vertical needles
NOCONNECT	Suppresses line segments that connect points on chart
OUTLABEL=	Labels points outside control limits on $\bar{X}$ chart
OUTLABEL2=	Labels points outside control limits on s chart
SYMBOLLEGEND=	Specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	Specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	Turns point labels so that they are strung out vertically
WNEEDLES=	Specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	Computes zone widths independently above and below center line
NO3SIGMACHECK	Enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	Suppresses tests across *phase* boundaries
TESTS=	Specifies tests for special causes for the $\bar{X}$ chart
TESTS2=	Specifies tests for special causes for the s chart
TEST2RESET=	Enables tests for special causes to be reset for the s chart
TEST2RUN=	Specifies length of pattern for Test 2
TEST3RUN=	Specifies length of pattern for Test 3
TESTACROSS	Applies tests across *phase* boundaries
TESTLABEL=	Provides labels for points where test is positive
TESTLABEL*n*=	Specifies label for *n*th test for special causes
TESTNMETHOD=	Applies tests to standardized chart statistics
TESTOVERLAP	Performs tests on overlapping patterns of points
TESTRESET=	Enables tests for special causes to be reset
WESTGARD=	Requests that Westgard rules be applied to the $\bar{X}$ chart
ZONELABELS	Adds labels A, B, and C to zone lines for $\bar{X}$ chart
ZONE2LABELS	Adds labels A, B, and C to zone lines for s chart
ZONES	Adds lines to $\bar{X}$ chart delineating zones A, B, and C
ZONES2	Adds lines to s chart delineating zones A, B, and C
ZONEVALPOS=	Specifies position of ZONEVALUES labels
ZONEVALUES	Labels $\bar{X}$ chart zone lines with their values

Table 19.75 *continued*

Option	Description
ZONE2VALUES	Labels *s* zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	Specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	Specifies color for labels indicating points where test is positive
CTESTSYMBOL=	Specifies color for symbol used to plot points where test is positive
CZONES=	Specifies color for lines and labels delineating zones A, B, and C
LTESTS=	Specifies type of line connecting points where test is positive
LZONES=	Specifies line type for lines delineating zones A, B, and C
TESTFONT=	Specifies software font for labels at points where test is positive
TESTHEIGHT=	Specifies height of labels at points where test is positive
TESTLABBOX	Requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	Specifies plot symbol for points where test is positive
TESTSYMBOLHT=	Specifies symbol height for points where test is positive
WTESTS=	Specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	Specifies color for axis lines and tick marks
CFRAME=	Specifies fill colors for frame for plot area
CTEXT=	Specifies color for tick mark values and axis labels
DISCRETE	Produces horizontal axis for discrete numeric group values
HAXIS=	Specifies major tick mark values for horizontal axis
HEIGHT=	Specifies height of axis label and axis legend text
HMINOR=	Specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	Specifies length of offset at both ends of horizontal axis
INTSTART=	Specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	Suppresses label for horizontal axis
NOTICKREP	Specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis

Table 19.75 *continued*

Option	Description
NOTRUNC	Suppresses vertical axis truncation at zero applied by default to *s* chart
NOVANGLE	Requests vertical axis labels that are strung out vertically
NOVLABEL	Suppresses label for primary vertical axis
NOV2LABEL	Suppresses label for secondary vertical axis
SKIPHLABELS=	Specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	Specifies splitting character for axis labels
TURNHLABELS	Requests horizontal axis labels that are strung out vertically
VAXIS=	Specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=	Specifies major tick mark values for vertical axis of *s* chart
VFORMAT=	Specifies format for primary vertical axis tick mark labels
VFORMAT2=	Specifies format for secondary vertical axis tick mark labels
VMINOR=	Specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	Specifies length of offset at both ends of vertical axis
VZERO	Forces origin to be included in vertical axis for primary chart
VZERO2	Forces origin to be included in vertical axis for secondary chart
WAXIS=	Specifies width of axis lines
Plot Layout Options	
ALLN	Plots means for all subgroups
BILEVEL	Creates control charts using half-screens and half-pages
EXCHART	Creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	Requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	Suppresses creation of charts
NOCHART2	Suppresses creation of *s* chart
NOFRAME	Suppresses frame for plot area
NOLEGEND	Suppresses legend for subgroup sample sizes

Table 19.75 *continued*

Option	Description
NPANELPOS=	Specifies number of subgroup positions per panel on each chart
REPEAT	Repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	Displays $\bar{X}$ and s charts on separate screens or pages
TOTPANELS=	Specifies number of pages or screens to be used to display chart
YPCT1=	Specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and s charts
ZEROSTD	Displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	Specifies color for lines requested by HREF= and HREF2= options
CVREF=	Specifies color for lines requested by VREF= and VREF2= options
HREF=	Specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=	Specifies position of reference lines perpendicular to horizontal axis on s chart
HREFDATA=	Specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA=	Specifies position of reference lines perpendicular to horizontal axis on s chart
HREFLABELS=	Specifies labels for HREF= lines
HREF2LABELS=	Specifies labels for HREF2= lines
HREFLABPOS=	Specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	Specifies line type for HREF= and HREF2= lines
LVREF=	Specifies line type for VREF= and VREF2= lines
NOBYREF	Specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	Specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=	Specifies position of reference lines perpendicular to vertical axis on s chart
VREFLABELS=	Specifies labels for VREF= lines
VREF2LABELS=	Specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels

Table 19.75 *continued*

Option	Description
Grid Options	
CGRID=	Specifies color for grid requested with GRID or ENDGRID option
ENDGRID	Adds grid after last plotted point
GRID	Adds grid to control chart
LENDGRID=	Specifies line type for grid requested with the ENDGRID option
LGRID=	Specifies line type for grid requested with the GRID option
WGRID=	Specifies width of grid lines
Clipping Options	
CCLIP=	Specifies color for plot symbol for clipped points
CLIPFACTOR=	Determines extent to which extreme points are clipped
CLIPLEGEND=	Specifies text for clipping legend
CLIPLEGPOS=	Specifies position of clipping legend
CLIPSUBCHAR=	Specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	Specifies plot symbol for clipped points
CLIPSYMBOLHT=	Specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	Specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=	Specifies annotate data set that adds features to s chart
DESCRIPTION=	Specifies description of $\bar{X}$ chart's GRSEG catalog entry
DESCRIPTION2=	Specifies description of s chart's GRSEG catalog entry
FONT=	Specifies software font for labels and legends on charts
NAME=	Specifies name of $\bar{X}$ chart's GRSEG catalog entry
NAME2=	Specifies name of s chart's GRSEG catalog entry
PAGENUM=	Specifies the form of the label used in pagination
PAGENUMPOS=	Specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	Specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	Draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	Specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	Shades areas between control limits and connected points lying outside the limits
STARFILL=	Specifies a variable identfying groups of stars filled with different colors

Table 19.75 *continued*

Option	Description
STARS=	Specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	Specifies the control limit infill transparency
MARKERDISPLAY=	Specifies a subset of subgroups to be plotted with markers in the $\bar{X}$ chart
MARKERDISPLAY2=	Specifies a subset of subgroups to be plotted with markers in the *s* chart
MARKERLABEL=	Specifies labels for subgroups that are plotted with markers in the $\bar{X}$ chart
MARKERLABEL2=	Specifies labels for subgroups that are plotted with markers in the *s* chart
MARKERMISSINGGROUP=	Specifies whether subgroups that have missing *symbol-variable* values are plotted with markers
MARKERS	Plots subgroup points with markers
NOBLOCKREF	Suppresses block and phase reference lines
NOBLOCKREFFILL	Suppresses block and phase wall fills
NOFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOPHASEREF	Suppresses block and phase reference lines
NOPHASEREFFILL	Suppresses block and phase wall fills
NOREF	Suppresses block and phase reference lines
NOREFFILL	Suppresses block and phase wall fills
NOSTARFILLLEGEND	Suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	Disables transparency in ODS Graphics output
ODSFOOTNOTE=	Specifies a graph footnote
ODSFOOTNOTE2=	Specifies a secondary graph footnote
ODSLEGENDEXPAND	Specifies that legend entries contain all levels observed in the data
ODSTITLE=	Specifies a graph title
ODSTITLE2=	Specifies a secondary graph title
OUTFILLTRANSPARENCY=	Specifies control limit outfill transparency
OVERLAYURL=	Specifies URLs to associate with overlay points
OVERLAY2URL=	Specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	Specifies vertical position of phase legend
PHASEREFLEVEL=	Associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	Specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	Draws central line labels using a simulated software font
STARTRANSPARENCY=	Specifies star fill transparency

Table 19.75 *continued*

Option	Description
URL=	Specifies a variable whose values are URLs to be associated with subgroups
URL2=	Specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	Specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	Creates output data set containing subgroup summary statistics
OUTINDEX=	Specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	Creates output data set containing control limits
OUTTABLE=	Creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	Creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	Augments basic table with values of central lines
TABLEID	Augments basic table with columns for ID variables
TABLELEGEND	Augments basic table with legend for tests for special causes
TABLEOUTLIM	Augments basic table with columns indicating control limits exceeded
TABLETESTS	Augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	Specifies α value and type for computing capability index confidence limits
LSL=	Specifies list of lower specification limits
TARGET=	Specifies list of target values
USL=	Specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	Specifies position of label for *block-variable* legend
BLOCKLABTYPE=	Specifies text size of *block-variable* legend

Table 19.75 *continued*

Option	Description
BLOCKPOS=	Specifies vertical position of *block-variable* legend
BLOCKREP	Repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	Specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	Specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	Specifies text color for *phase* legend
NOPHASEFRAME	Suppresses default frame for *phase* legend
OUTPHASE=	Specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	Disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	Specifies text size of *phase* legend
PHASELEGEND	Displays *phase* labels in a legend across top of chart
PHASELIMITS	Labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	Specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	Specifies color for STARCIRCLES= circles
CSTARFILL=	Specifies color for filling stars
CSTAROUT=	Specifies outline color for stars exceeding inner or outer circles
CSTARS=	Specifies color for outlines of stars
LSTARCIRCLES=	Specifies line types for STARCIRCLES= circles
LSTARS=	Specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	Specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	Specifies reference circles for stars
STARINRADIUS=	Specifies inner radius of stars
STARLABEL=	Specifies vertices to be labeled
STARLEGEND=	Specifies style of legend for star vertices
STARLEGENDLAB=	Specifies label for STARLEGEND= legend
STAROUTRADIUS=	Specifies outer radius of stars
STARSPECS=	Specifies method used to standardize vertex variables
STARSTART=	Specifies angle for first vertex
STARTYPE=	Specifies graphical style of star
STARVERTICES=	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=	Specifies width of STARCIRCLES= circles
WSTARS=	Specifies width of STARVERTICES= stars

Table 19.75 *continued*

Option	Description
Overlay Options	
CCOVERLAY=	Specifies colors for primary chart overlay line segments
CCOVERLAY2=	Specifies colors for secondary chart overlay line segments
COVERLAY=	Specifies colors for primary chart overlay plots
COVERLAY2=	Specifies colors for secondary chart overlay plots
COVERLAYCLIP=	Specifies color for clipped points on overlays
LOVERLAY=	Specifies line types for primary chart overlay line segments
LOVERLAY2=	Specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	Suppresses legend for overlay plots
OVERLAY=	Specifies variables to overlay on primary chart
OVERLAY2=	Specifies variables to overlay on secondary chart
OVERLAY2HTML=	Specifies links to associate with secondary chart overlay points
OVERLAY2ID=	Specifies labels for secondary chart overlay points
OVERLAY2SYM=	Specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	Specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	Specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	Specifies symbol height for clipped points on overlays
OVERLAYHTML=	Specifies links to associate with primary chart overlay points
OVERLAYID=	Specifies labels for primary chart overlay points
OVERLAYLEGLAB=	Specifies label for overlay legend
OVERLAYSYM=	Specifies symbols for primary chart overlays
OVERLAYSYMHT=	Specifies symbol heights for primary chart overlays
WOVERLAY=	Specifies widths of primary chart overlay line segments
WOVERLAY2=	Specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	Specifies a variable whose values create links to be associated with subgroups
HTML2=	Specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	Specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	Creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	Specifies plot character for clipped points

Table 19.75 *continued*

Option	Description
CONNECTCHAR=	Specifies character used to form line segments that connect points on chart
HREFCHAR=	Specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	Specifies characters indicating *symbol-variable*
TESTCHAR=	Specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	Specifies line character for VREF= and VREF2= lines
ZONECHAR=	Specifies character for lines that delineate zones for tests for special causes

Details: XSCHART Statement

The following sections provide details that are specific to the XSCHART statement. See the section "Chart Statement Details: SHEWHART Procedure" on page 1966 for details that apply to all the SHEWHART procedure chart statements.

Constructing Charts for Means and Standard Deviations

The following notation is used in this section:

μ	Process mean (expected value of the population of measurements)
σ	Process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	Mean of measurements in ith subgroup
s_i	Standard deviation of the measurements $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup

$$s_i = \sqrt{((x_{i1} - \bar{X}_i)^2 + \cdots + (x_{in_i} - \bar{X}_i)^2)/(n_i - 1)}$$

n_i	Sample size of ith subgroup
N	Number of subgroups
$\overline{\overline{X}}$	Weighted average of subgroup means
z_p	$100p$th percentile of the standard normal distribution
$c_4(n)$	Expected value of the standard deviation of n independent normally distributed variables with unit standard deviation
$c_5(n)$	Standard error of the standard deviation of n independent observations from a normal population with unit standard deviation
$\chi^2_p(n)$	$100p$th percentile ($0 < p < 1$) of the χ^2 distribution with n degrees of freedom

Plotted Points

Each point on an $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 13, the mean plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 13}{5} = 15$$

Each point on an s chart indicates the value of a subgroup standard deviation (s_i). For example, the standard deviation plotted for the tenth subgroup is

$$s_{10} = \sqrt{((12-15)^2 + (15-15)^2 + (19-15)^2 + (16-15)^2 + (13-15)^2)/4} = 2.739$$

Central Lines

On an $\bar{X}$ chart, by default, the central line indicates an estimate of μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1\bar{X}_1 + \cdots + n_N\bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

On the s chart, by default, the central line for the ith subgroup indicates an estimate for the expected value of s_i, which is computed as $c_4(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $c_4(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard errors of $\bar{X}_i$ and s_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ or s_i exceeds the limits

The following table provides the formulas for the limits:

Table 19.77 Limits for $\bar{X}$ and s Charts

Control Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$
	UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
s Chart	LCL = lower limit = $\max(c_4(n_i)\hat{\sigma} - kc_5(n_i)\hat{\sigma}, 0)$
	UCL = upper limit = $c_4(n_i)\hat{\sigma} + kc_5(n_i)\hat{\sigma}$
Probability Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
	UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
s Chart	LCL = lower limit = $\hat{\sigma}\sqrt{\chi^2_{\alpha/2}(n_i - 1)/(n_i - 1)}$
	UCL = upper limit = $\hat{\sigma}\sqrt{\chi^2_{1-\alpha/2}(n_i - 1)/(n_i - 1)}$

The formulas for s charts assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 19.77. Note that the limits vary with n_i and that the probability limits for s_i are asymmetric about the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. Table 19.78 lists the variables that are saved.

Table 19.78 OUTLIMITS= Data Set

Variable	Description
ALPHA	Probability (α) of exceeding limits
_CP_	Capability index C_p
CPK	Capability index C_{pk}
CPL	Capability index CPL
CPM	Capability index C_{pm}
CPU	Capability index CPU
INDEX	Optional identifier for the control limits specified with the OUTINDEX= option
LCLS	Lower control limit for subgroup standard deviation
LCLX	Lower control limit for subgroup mean
LIMITN	Nominal sample size associated with the control limits
LSL	Lower specification limit
MEAN	Process mean ($\overline{\overline{X}}$ or μ_0)
_S_	Value of central line on s chart
SIGMAS	Multiple (k) of standard error of $\bar{X}_i$ or s_i
STDDEV	Process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*Subgroup-variable* specified in the XSCHART statement
TARGET	Target value
TYPE	Type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLS	Upper control limit for subgroup standard deviation
UCLX	Upper control limit for subgroup mean
USL	Upper specification limit
VAR	*Process* specified in the XSCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value 'V' is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLS_, _S_, and _UCLS_.
2. If the limits are defined in terms of a multiple k of the standard errors of $\bar{X}_i$ and s_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1971 for computational details.
5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XSCHART statement. For an example, see "Saving Control Limits" on page 1933.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the XSCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   xschart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthS, WidthN, DiameterX, DiameterS, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*

- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1932.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. Table 19.79 lists the variables that are saved.

Table 19.79 OUTTABLE= Data Set Variables

Variable	Description
ALPHA	Probability (α) of exceeding control limits
EXLIM	Control limit exceeded on $\bar{X}$ chart
EXLIMS	Control limit exceeded on s chart
LCLS	Lower control limit for standard deviation
LCLX	Lower control limit for mean
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
_S_	Average standard deviation
SIGMAS	Multiple (k) of the standard error associated with control limits
Subgroup	Values of the subgroup variable
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
SUBX	Subgroup mean
TESTS	Tests for special causes signaled on $\bar{X}$ chart
TESTS2	Tests for special causes signaled on s chart
UCLS	Upper control limit for standard deviation
UCLX	Upper control limit for mean
VAR	*Process* specified in the XSCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variable _TESTS2_ is saved if you specify the TESTS2= option.
4. The variables _EXLIM_, _EXLIMS_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1933.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XSCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XSCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2079).

For an example of a DATA= data set, see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1926.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   xschart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, _UCLX_, _LCLS_, _S_, and _UCLS_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 19.77

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1935.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the XSCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup standard deviation variable for each *process*

- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *X*, *S*, and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=Summary;
   xschart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightS, WeightN, YieldstrengthX, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2079 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means and Standard Deviations from Summary Data" on page 1929.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2143.

Table 19.80 lists the variables required in a TABLE= data set used with the XSCHART statement:

Table 19.80 Variables Required in a TABLE= Data Set

Variable	Description
LCLS	Lower control limit for standard deviation
LCLX	Lower control limit for mean

Table 19.80 *continued*

Variable	Description
LIMITN	Nominal sample size associated with the control limits
MEAN	Process mean
_S_	Average standard deviation
Subgroup-variable	Values of the *subgroup-variable*
SUBN	Subgroup sample size
SUBS	Subgroup standard deviation
SUBX	Subgroup mean
UCLS	Upper control limit for standard deviation
UCLX	Upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup means and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup standard deviations and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1933.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, four methods (referred to as default, MVLUE, MVGRANGE, and RMSDF) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here, $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVGRANGE Method

If you specify SMETHOD=MVGRANGE, σ is estimated using a moving range of subgroup averages. This is appropriate for constructing control charts for means when the jth measurement in the ith subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}.

The estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive subgroup averages used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the subgroup averages are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is appropriate for constructing a variation on the three-way control chart that is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or *within-group* variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2152. Wheeler's three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means,

and a chart of subgroup ranges. This variation substitutes a chart of subgroup standard deviations for the chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XSCHART statement produces the appropriate charts of subgroup means and subgroup standard deviations.

RMSDF Method

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Examples: XSCHART Statement

This section provides advanced examples of the XSCHART statement.

Example 19.41: Specifying Probability Limits

NOTE: See *X-Bar and s Charts with Probability Limits* in the SAS/QC Sample Library.

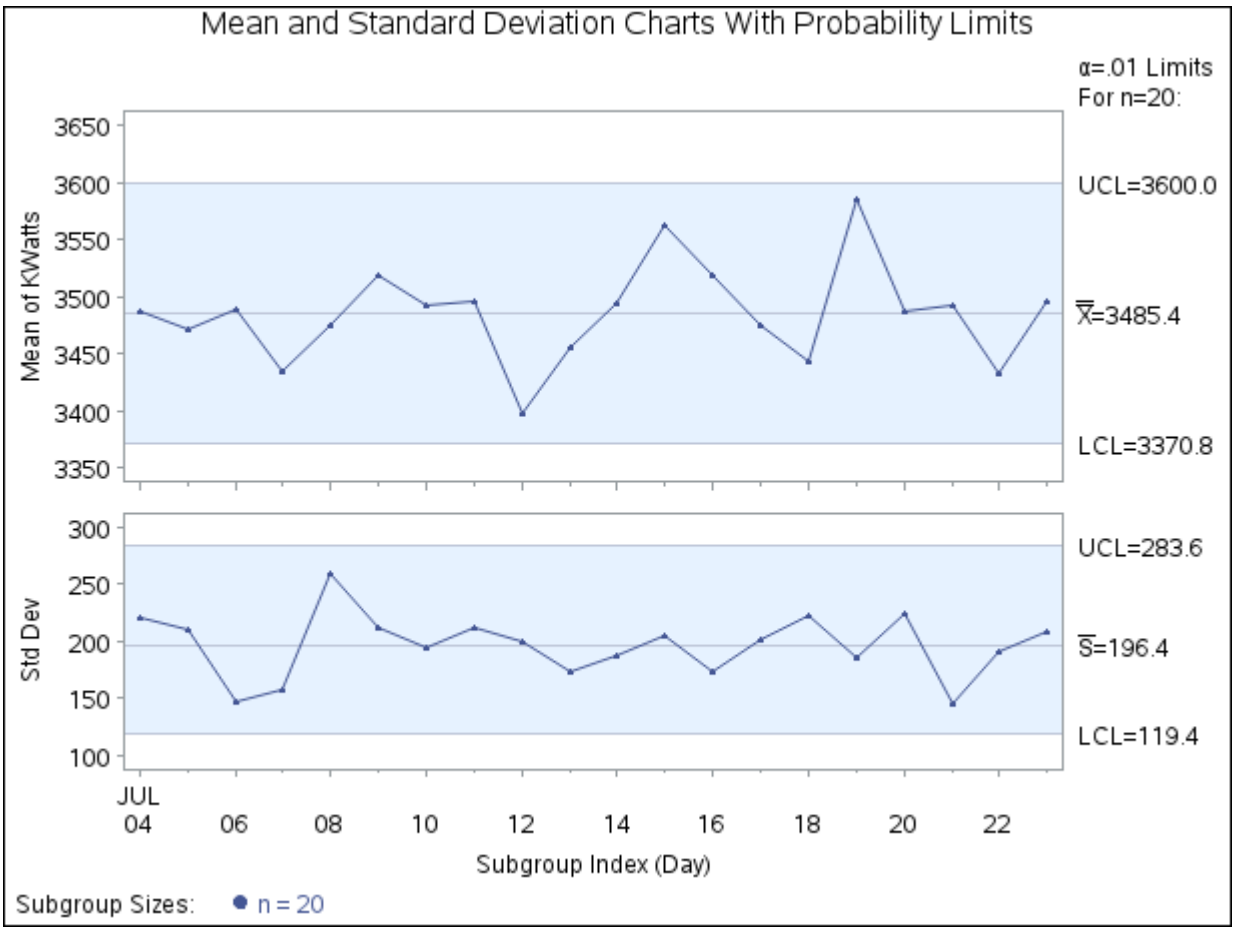
This example illustrates how to create $\bar{X}$ and s charts with probability limits. The following statements read the kilowatt power output measurements from the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1926) and create the $\bar{X}$ and s charts shown in Output 19.41.1:

```
ods graphics off;
symbol v=dot h=.8;
title 'Mean and Standard Deviation Charts With Probability Limits';
proc shewhart data=Turbine;
   xschart KWatts*Day  / alpha     = 0.01
                         outlimits = Oillim;
run;
```

The ALPHA= option specifies the probability (α) that a subgroup summary statistic is outside the limits. Here, the limits are computed so that the probability that a subgroup mean or standard deviation is less than its lower limit is $\alpha/2 = 0.005$, and the probability that a subgroup mean or standard deviation is greater than its upper limit is $\alpha/2 = 0.005$. This assumes that the measurements are normally distributed.

The OUTLIMITS= option names an output data set (Oillim) that saves the probability limits. The data set Oillim is shown in Output 19.41.2.

Output 19.41.1 Probability Limits on $\bar{X}$ and s Charts



Output 19.41.2 Probability Limit Information

Mean and Standard Deviation Charts with Probability Limits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	0.01	2.57583	3370.79	3485.41	3600.03

LCLS	_S_	_UCLS_	_STDDEV_
119.432	196.396	283.570	198.996

The variable _ALPHA_ saves the value of α. The value of the variable _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function. Note that, in this case, the probability limits for the mean are equivalent to 2.58σ limits.

Because all the points fall within the probability limits, it can be concluded that the process is in statistical control.

Example 19.42: Computing Subgroup Summary Statistics

NOTE: See *Reading Subgroup Summary Data* in the SAS/QC Sample Library.

You can use output data sets from a number of SAS procedures as input data sets for the SHEWHART procedure. In this example, the MEANS procedure is used to create a data set containing subgroup summary statistics, which can be read by the SHEWHART procedure as a HISTORY= data set. The following statements create an output data set named Oilsummeans, which contains subgroup means, standard deviations, and sample sizes for the variable KWatts in the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1926):

```
proc means data=Turbine noprint;
   var KWatts;
   by Day;
   output out=Oilsummeans mean=means std=stds n=sizes;
run;
```

A listing of Oilsummeans is shown in Output 19.42.1.

Output 19.42.1 The Data Set Oilsummeans

Summary Statistics for Power Output Data

Day	_TYPE_	_FREQ_	means	stds	sizes
04JUL	0	20	3487.40	220.260	20
05JUL	0	20	3471.65	210.427	20
06JUL	0	20	3488.30	147.025	20
07JUL	0	20	3434.20	157.637	20
08JUL	0	20	3475.80	258.949	20
09JUL	0	20	3518.10	211.566	20
10JUL	0	20	3492.65	193.779	20
11JUL	0	20	3496.40	212.024	20
12JUL	0	20	3398.50	199.201	20
13JUL	0	20	3456.05	173.455	20
14JUL	0	20	3493.60	187.465	20
15JUL	0	20	3563.30	205.472	20
16JUL	0	20	3519.05	173.676	20
17JUL	0	20	3474.20	200.576	20
18JUL	0	20	3443.60	222.084	20
19JUL	0	20	3586.35	185.724	20
20JUL	0	20	3486.45	223.474	20
21JUL	0	20	3492.90	145.267	20
22JUL	0	20	3432.80	190.994	20
23JUL	0	20	3496.90	208.858	20

The variables MEANS, STDS, and SIZES do not follow the naming convention required for HISTORY= data sets (see "HISTORY= Data Set" on page 1955. The following statements temporarily rename these variables to KWattsX, KWattsS, and KWattsN, respectively (the names required when the *process* KWatts is specified in the XSCHART statement):

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart
   history=Oilsummeans (rename=(means  = KWattsX
                                stds   = KWattsS
                                sizes  = KWattsN ));
   xschart KWatts*Day;
run;
```

The resulting charts are identical to the charts in Figure 19.115.

Example 19.43: Analyzing Nonnormal Process Data

NOTE: See *Analyzing Nonnormal Process Data* in the SAS/QC Sample Library.

The standard control limits for *s* charts (see Table 19.77) are calculated under the assumption that the data are normally distributed. This example illustrates how a transformation to normality can be used in conjunction with $\bar{X}$ and *s* charts.

The length of a metal brace is measured in centimeters for each of 20 braces sampled daily. Subgroup samples are collected for nineteen days, and the data are analyzed to determine if the manufacturing process is in statistical control.

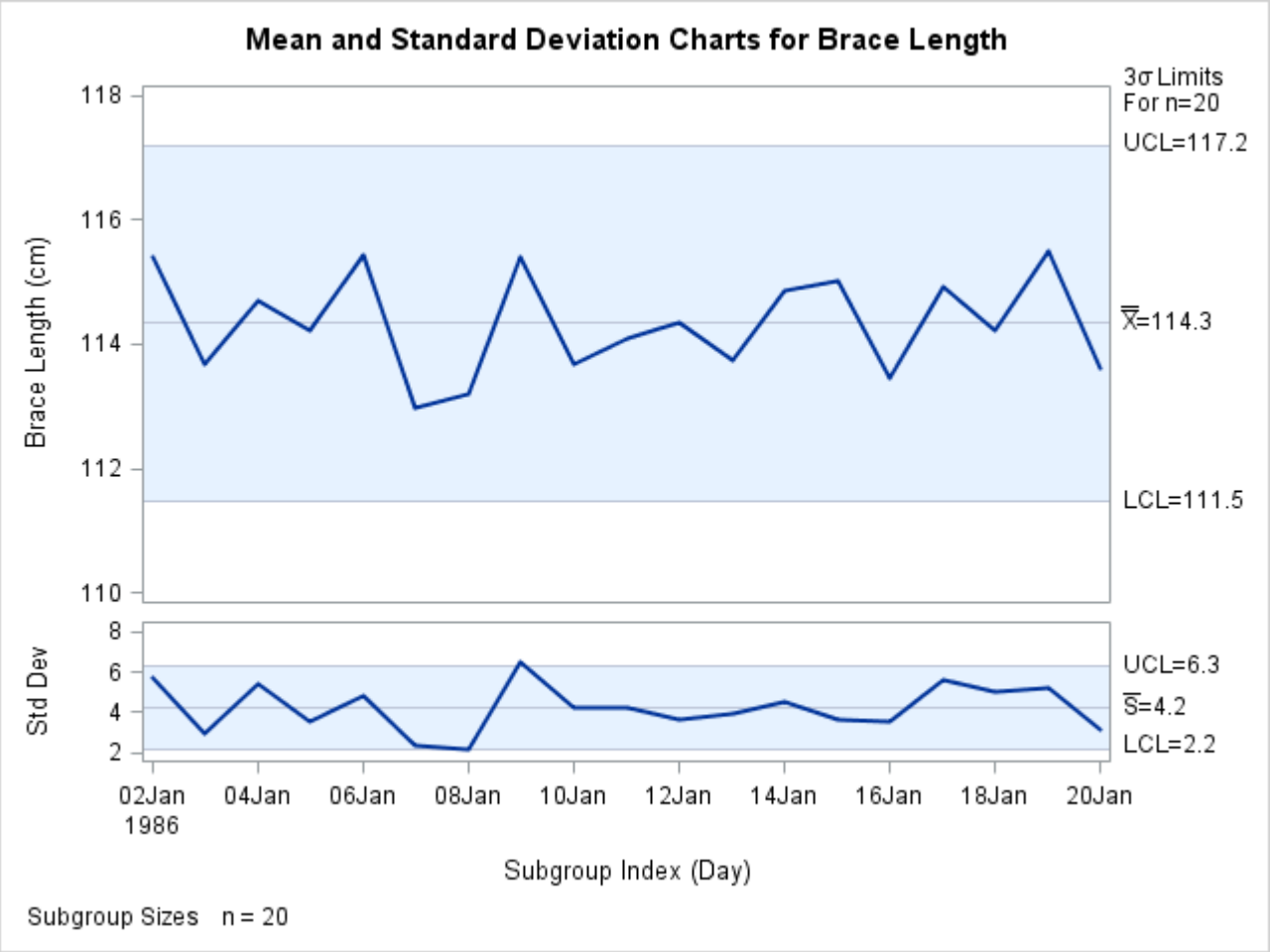
```
data LengthData;
   informat Day date7.;
   format Day date5.;
   label Length='Brace Length (cm)';
   input Day @;
   do i=1 to 5;
      input Length @;
      output;
   end;
   drop i;
   datalines;
02JAN86  113.64  119.60  111.66  111.88  125.29
02JAN86  114.08  115.28  127.84  109.97  109.34
02JAN86  109.65  121.76  112.17  116.01  111.64
02JAN86  112.70  114.43  110.27  114.76  125.89
03JAN86  115.92  113.62  117.52  114.44  118.08
03JAN86  111.13  118.42  112.16  112.25  107.71
03JAN86  110.46  113.78  109.89  114.59  116.98

   ... more lines ...

20JAN86  115.15  112.34  114.99  109.70  111.20
20JAN86  117.81  119.51  109.03  111.61  118.01
20JAN86  113.55  114.78  112.91  111.87  118.54
;
```

The following statements create preliminary $\bar{X}$ and *s* charts for the lengths:

```
ods graphics on;
title 'Mean and Standard Deviation Charts for Brace Length';
proc shewhart data=LengthData;
   xschart Length*Day / odstitle = title;
run;
```

The charts are shown in Output 19.43.1.

Output 19.43.1 $\bar{X}$ and s Charts



The s chart suggests that the process is not in control, because the standard deviation of the measurements recorded on January 9 exceeds its upper control limit. In addition, a number of other points on the s chart are close to the control limits.

The following statements create a box chart for the lengths (for more information about box charts, see "BOXCHART Statement: SHEWHART Procedure" on page 1417).

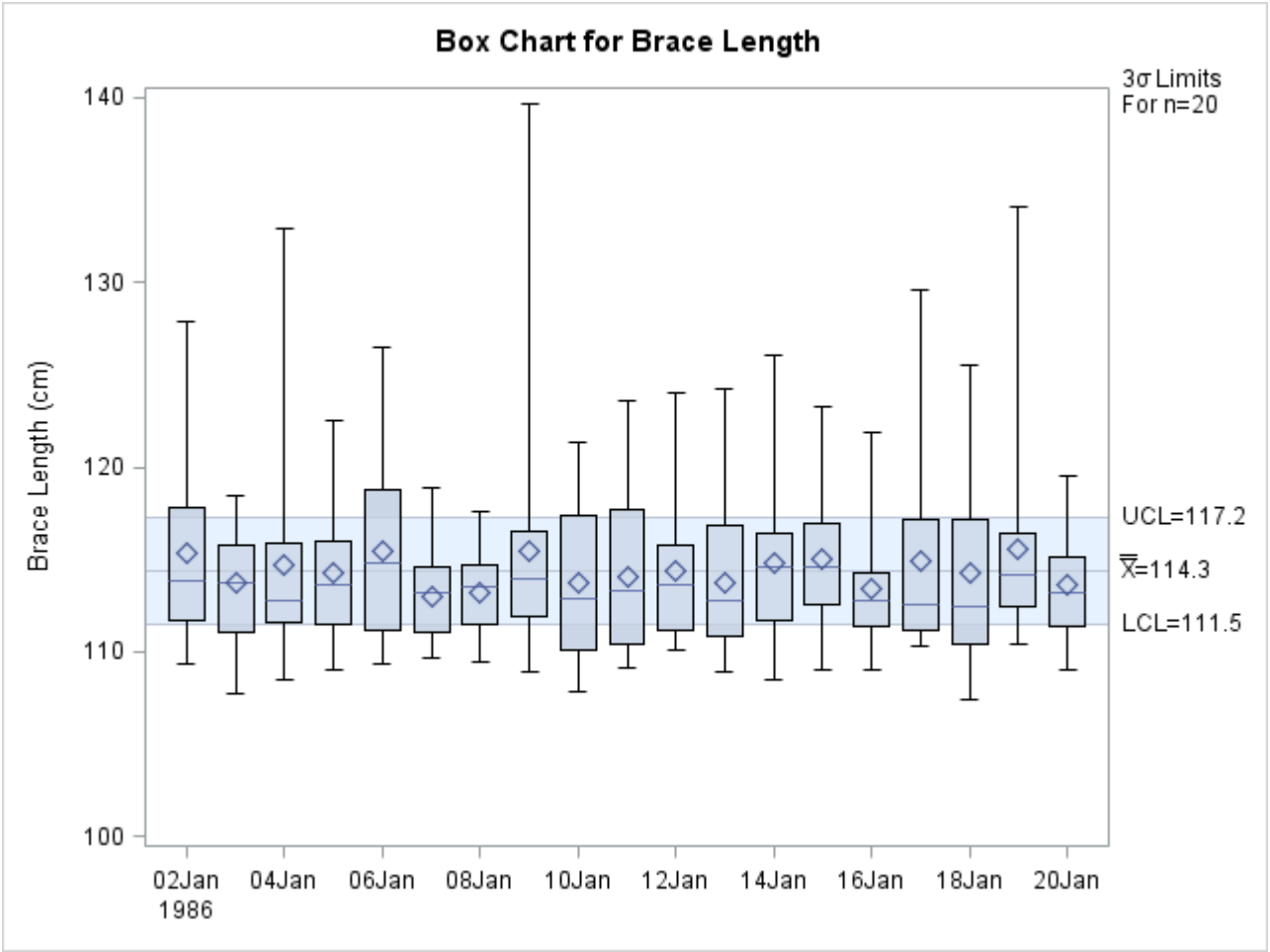
```
title 'Box Chart for Brace Length';
proc shewhart data=LengthData;
   boxchart Length*Day / serifs
                         ranges
                         nohlabel
```

```
                              nolegend
                              odstitle = title;
run;
```

The chart, shown in Output 19.43.2, reveals that most of the subgroup distributions are skewed to the right. Consequently, the *s* chart shown in Output 19.43.1 should be interpreted with caution, because control limits for *s* charts are based on the assumption that the data are normally distributed.

No special cause for the skewness of the subgroup distributions is discovered. This indicates that the process is in statistical control and that the length distribution is naturally skewed.

Output 19.43.2 Box Chart



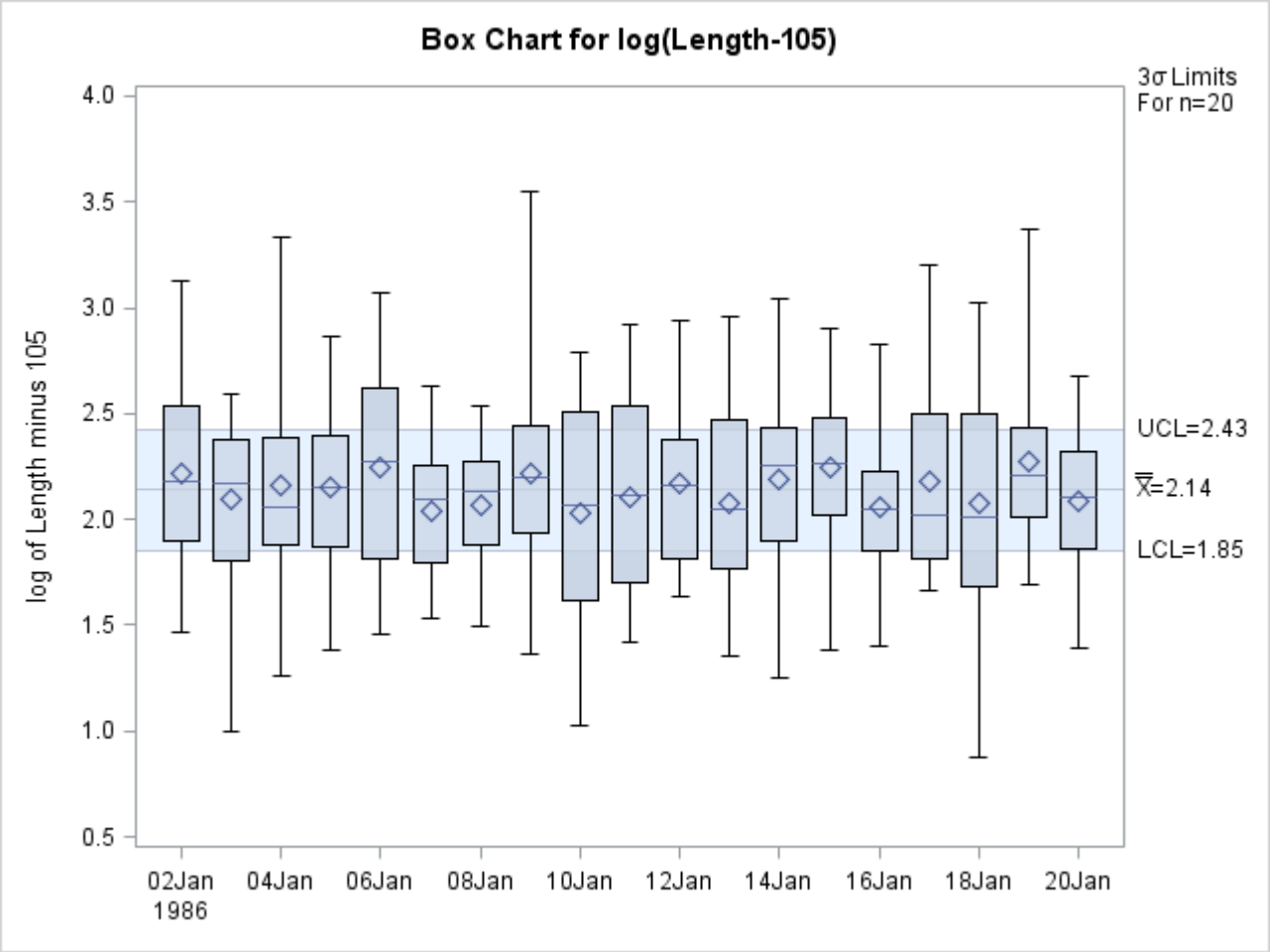
The following statements apply a lognormal transformation to the length measurements and display a box chart for the transformed data:

```
data LengthData;
   set LengthData;
   LogLength=log(Length-105);
label LogLength='log of Length minus 105';
run;
```

```
title 'Box Chart for log(Length-105)';
proc shewhart data=LengthData;
   boxchart LogLength*Day / serifs
                            ranges
                            nohlabel
                            nolegend
                            odstitle = title;
run;
```

The chart, shown in Output 19.43.3, indicates that the subgroup distributions of LogLength are approximately normal (this can be verified with goodness-of-fit tests by using the CAPABILITY procedure).

Output 19.43.3 Box Chart for Transformed Data



Finally, $\bar{X}$ and s charts, shown in Output 19.43.4, are created for LogLength. They indicate that the variability and mean level of the transformed lengths are in control.

```
title 'Mean and Standard Deviation Charts for log(Length)';
proc shewhart data=LengthData;
   xschart LogLength*Day / split    = '/'
                           odstitle = title;
   label LogLength='Avg log of Length/Std Dev';
run;
```

Output 19.43.4 $\bar{X}$ and s Charts for Transformed Length

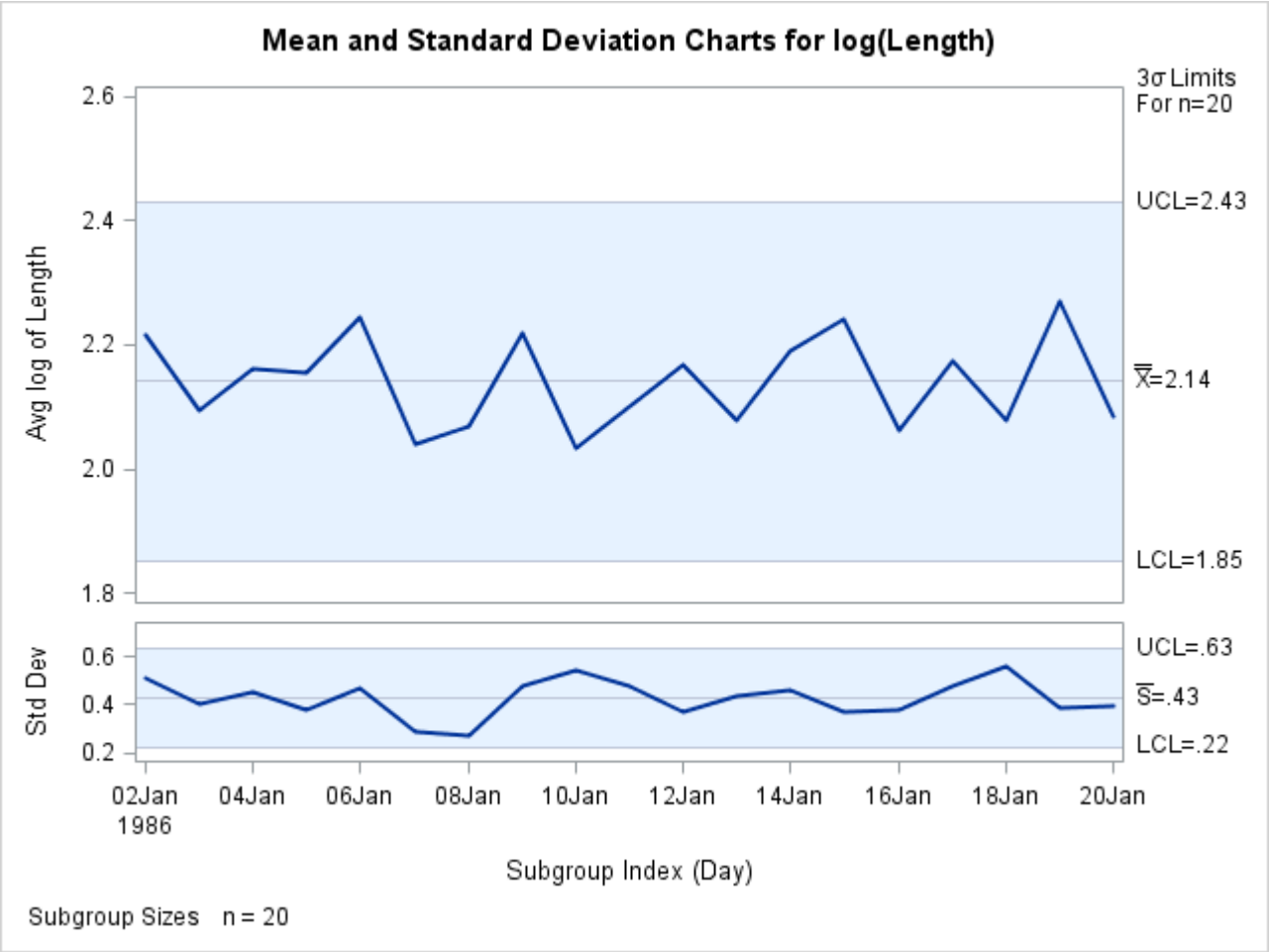


Chart Statement Details: SHEWHART Procedure

The following sections provide details that apply to all the chart statements in the SHEWHART procedure. For descriptions of details that are specific to the different chart statements, see the "Details" sections for BOXCHART, CCHART, IRCHART, MCHART, MRCHART, NPCHART, PCHART, RCHART, SCHART, UCHART, XCHART, XRCHART, and XSCHART.

ODS Tables

Each table created by PROC SHEWHART has a name associated with it, and you must use this name to reference the table when you use ODS statements. These names are listed in Table 19.81.

Table 19.81 ODS Tables Produced by PROC SHEWHART

Table Name	Description	Statement	Options
BoxchartSummary	Box chart summary statistics	BOXCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
CChartSummary	*c* chart summary statistics	CCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
IRChartSummary	Individual measurement and moving range chart summary statistics	IRCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
MChartSummary	Median chart summary statistics	MCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
MRChartSummary	Median and *R* chart summary statistics	MRCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
NPChartSummary	*np* chart summary statistics	NPCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
PChartSummary	*p* chart summary statistics	PCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
RChartSummary	*R* chart summary statistics	RCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
SChartSummary	*s* chart summary statistics	SCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	Descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	All	TABLEALL, TABLELEG
UChartSummary	*u* chart summary statistics	UCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
XChartSummary	$\bar{X}$ chart summary statistics	XCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
XRChartSummary	$\bar{X}$ and *R* chart summary statistics	XRCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
XSChartSummary	$\bar{X}$ and *s* chart summary statistics	XSCHART	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*).

The appearance of ODS Graphics output is determined by the style associated with the ODS destination where the graph is produced. Chart statement options that are used to control the appearance of traditional graphics are ignored for ODS Graphics output.

Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics.

Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1993.

When ODS Graphics is in effect, the SHEWHART procedure assigns names to the graphs it creates. You can use these names to reference the graphs when using ODS. The names are listed in Table 19.82.

Table 19.82 ODS Graphics Produced by PROC SHEWHART

ODS Graph Name	Plot Description	Statement
BoxChart	Box chart	BOXCHART
CChart	*c* chart	CCHART
IRChart	Individual measurements and moving ranges chart	IRCHART
MChart	Median chart	MCHART
MRChart	Median and *R* chart	MRCHART
NPChart	*np* chart	NPCHART
PChart	*p* chart	PCHART
RChart	*R* chart	RCHART
SChart	*s* chart	SCHART
UChart	*u* chart	UCHART
XChart	$\bar{X}$ chart	XCHART
XRChart	$\bar{X}$ and *R* chart	XRCHART
XSChart	$\bar{X}$ and *s* chart	XSCHART

See Chapter 4, "SAS/QC Graphics," for more information about ODS Graphics and other methods that you can use to produce charts.

ODS Graphics Template

When you specify a *symbol-variable* with ODS Graphics enabled, markers are assigned to subgroups based on the values of the *symbol-variable*. By default, the appearance of the markers is determined by the marker shape, color, and contrast color attributes of the GraphData1, . . . , GraphData*N* elements in the current ODS style.

One way to control the marker attributes is to modify the ODS style. Another method is to specify options in the BEGINGRAPH statement in the graph template. You can specify these options by following these steps:

1. Copy the template.
2. Modify the BEGINGRAPH statement marker options.
3. Recompile the template.
4. Save a copy of the template in your local SASUSER library.

To enable you to modify marker attributes without modifying the ODS style or the graph template, the template declares three reserved, global SAS macro variables:

Macro Variable	Marker Symbol Option
&_COLOR	DATACOLORS = (color list)
&_CONTRAST	DATACONTRASTCOLORS = (contrast color list)
&_SYMBOL	DATASYMBOLS = (marker symbol list)

To change an attribute value, you can use a %LET statement in your SAS code to assign a new value (or list of values) to the appropriate macro variable before you submit your procedure code. For example, to change the attributes of the first three group symbols, you can submit the following statements:

```
%let _color    = ( GraphData6:color GraphData3:color PINK );
%let _contrast = ( GraphData6:contrastcolor GraphData3:contrastcolor BLACK );

proc shewhart;
   xchart x*i = group;
   run;

%let _color    = ;  * reset;
%let _contrast = ;  * reset;
%let _symbol   = ;  * reset;
```

The values that you specify for a macro variable are in effect until you assign new values or your SAS session ends. You can assign empty values to the variables to restore the default attribute values.

You can also use these macros when you want to use the same marker for all the subgroups in a chart. In this case, you specify a *symbol-variable* that has the same value for every subgroup. The GraphDataDefault style element determines marker attributes when you do not specify a *symbol-variable*. The following statements change only the shape of the default marker and use that marker for all subgroups:

```
%let _color    = GraphDataDefault:color;
%let _contrast = GraphDataDefault:contrastcolor;
%let _symbol   = SquareFilled;

proc shewhart;
   xchart x*i = constant / markermissinggroup = false
                           symbollegend = none;
   run;

%let _color    = ;  * reset;
%let _contrast = ;  * reset;
%let _symbol   = ;  * reset;
```

The SYMBOLLEGEND=NONE option suppresses the symbol legend, which is not needed in this case.

The macro variables that are described here are declared in the template by using an MVAR statement. You can extend the flexibility of the graph template by using macro variables for other options in the template. For more information about the MVAR and BEGINGRAPH statements, see *SAS Graph Template Language: User's Guide*

Subgroup Variables

The values of the *subgroup-variable*, which is specified in the chart statement, indicate how the observations in the input data set (a DATA=, HISTORY=, or TABLE= data set) are arranged into rational subgroups.[9] Typically, the values of the *subgroup-variable* are one of the following:

- *indices* that give the order in which subgroup samples were collected (for example, 1, 2, 3, . . .). An unformatted numeric *subgroup-variable* is appropriate for this situation. For an example that uses this type of *subgroup-variable*, see "Creating Charts for Means and Ranges from Raw Data" on page 1882.

- the *dates* or *times* at which subgroup samples were collected (for example, 01JUN, 02JUN, 03JUN, . . .). A numeric *subgroup-variable* with a SAS date, time, or datetime format is appropriate for this situation. You can optionally associate a format with the *subgroup-variable* by using a FORMAT statement; refer to *SAS Formats and Informats: Reference* for details. For an example that uses this type of *subgroup-variable*, see Example 19.40.

- *labels* that uniquely identify subgroup samples (for example, Lot39, LotX12, Lot43A). A character *subgroup-variable* (with or without a format) is appropriate for this situation. For an example that uses this type of *subgroup-variable*, see Example 19.38.

The values of the *subgroup-variable* also determine how the horizontal axis of the control chart is scaled and labeled.

The notion of a rational subgroup is fundamental to the application of a Shewhart chart. You should select your subgroups so that if special causes of variation are present, the opportunity for variation within subgroups is minimized while the opportunity for variation between subgroups is maximized. In other words, the conditions within a subgroup should be homogeneous. The reason for this requirement is that the construction of the control limits is based on within-subgroup variability. Refer to Montgomery (1996) and Wheeler and Chambers (1986) for approaches to rational subgrouping.

The selection of subgroups is both a practical and a statistical issue that requires knowledge of the process and the sampling or measurement procedure. The values of the subgroup-variable should reflect the selection of subgroups and should not be assigned arbitrarily. Incorrect subgrouping or assignment of subgroup-variable values can result in control limits that are too tight or too wide.

If the input data set is a HISTORY= or TABLE= data set, each observation represents a distinct subgroup, and, consequently, the observations within each BY group must have distinct subgroup variable values. Similarly, if the input data set is a DATA= data set and you are using the CCHART, IRCHART, NPCHART, PCHART, or UCHART statement, each observation represents a distinct subgroup, and, consequently, the observations

[9] This discussion also applies to the use of *subgroup-variables* in the CUSUM procedure and the MACONTROL procedure.

within each BY group must have distinct subgroup variable values. However, if the input data set is a DATA= data set and you are using the BOXCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, or XSCHART statement, subgroups are identified by groups of consecutive observations with identical values of the subgroup-variable.

The order of the observations in the input data set and the scaling of the horizontal axis depend on the type of the subgroup-variable, which can be numeric or character.

Numeric Subgroup Variables

If the subgroup-variable is numeric, the observations must be sorted in increasing order of the values of the subgroup variable. If you use a BY statement, first sort by the BY variables and then by the subgroup variable.

The unformatted values of the subgroup-variable are used to scale the horizontal axis of the control chart, and the formatted values are used to label the major tick marks on the horizontal axis. As a result, the horizontal distance between two points corresponding to consecutive subgroups is proportional to the difference between their unformatted subgroup values.

If a DATE, DATETIME, WEEKDATE, or WORDDATE format is associated with the subgroup variable, the major tick mark labels are split and displayed in two levels to save space. You can override this default with the TURNHLABELS option (which turns the labels vertically) or with tick label options in an AXIS*n* statement specified with the HAXIS= option.

Character Subgroup Variables

If the subgroup-variable is numeric, the order of the observations is not checked. The horizontal axis is scaled so that the subgroups are spaced uniformly. Formatted subgroup variable values are used to label the major tick marks.

You can use a character subgroup variable to avoid gaps between groups of points or time values on a control chart. You can also use a character subgroup variable to create a chart in which the order of the points depends only on the order in which the subgroups are arranged in the input data set.

You should verify the order of the observations in the input data set before you use a character subgroup variable in conjunction with the TESTS= option. With the exception of Test 1, the tests for special causes are applicable only if the subgroups are provided in chronological order. See "Tests for Special Causes: SHEWHART Procedure" on page 2119 for details.

To avoid collision of adjacent tick labels on the horizontal axis, the labels are thinned by default. You can override this default with the TURNHLABELS option or with tick label options in an AXIS*n* statement specified with the HAXIS= option.

Capability Indices

This section provides formulas for process capability indices, which are saved in the OUTLIMITS= data set when you use the LSL= and USL= options to provide lower and upper specification limits (LSL and USL, respectively) for the *process*. The estimate $\hat{\sigma}$ is computed as described in the previous section, "Methods for Estimating the Standard Deviation" on page 1915

The Index C_p

The process capability index C_p is computed as

$$C_p = (USL - LSL)/6\hat{\sigma}$$

If you do not specify both LSL and USL, the variable _CP_ is assigned a missing value.

The Index CPL

The process capability index *CPL* is computed as

$$CPL = (\overline{\overline{X}} - LSL)/3\hat{\sigma}$$

If you do not specify LSL, the variable _CPL_ is assigned a missing value.

The Index CPU

The process capability index *CPU* is computed as

$$CPU = (USL - \overline{\overline{X}})/3\hat{\sigma}$$

If you do not specify USL, the variable _CPU_ is assigned a missing value.

The Index C_{pk}

The process capability index C_{pk} is computed as

$$C_{pk} = \min(USL - \overline{\overline{X}}, \overline{\overline{X}} - LSL)/3\hat{\sigma}$$

If you specify only USL, the index C_{pk} is computed as

$$C_{pk} = (USL - \overline{\overline{X}})/3\hat{\sigma}$$

and if you specify only LSL, the index C_{pk} is computed as

$$C_{pk} = (\overline{\overline{X}} - LSL)/3\hat{\sigma}$$

The Index C_{pm}

The process capability index C_{pm} is computed as

$$C_{pm} = \frac{\min(T - LSL, USL - T)}{3\sqrt{\hat{\sigma}^2 + (\overline{\overline{X}} - T)^2}}$$

where T is the target value specified with the TARGET= option.

When a single specification limit (SL) and target are specified, C_{pm} is computed as

$$C_{pm} = \frac{|T - SL|}{3\sqrt{\hat{\sigma}^2 + (\overline{\overline{X}} - T)^2}}$$

You can also use the CAPABILITY procedure to compute a variety of capability indices. The SHEWHART procedure and the CAPABILITY procedure use the same formulas to calculate the indices, but they use different estimates for the process standard deviation σ.

- The SHEWHART procedure calculates $\hat{\sigma}$ from subgroup estimates of σ. For details, see the previous section, "Methods for Estimating the Standard Deviation."
- The CAPABILITY procedure calculates $\hat{\sigma}$ as the sample standard deviation of the entire sample. For details, see the section "Standard Deviation" on page 225.

Regardless of which method you use, you should verify that the process is in statistical control before interpreting the indices, and you should verify that the data are normally distributed. The CAPABILITY procedure provides a variety of statistical and graphical tests for checking normality.

Some references use different notation and names for capability indices. For example, the manual ASQC Automotive Division/AIAG (1990) uses the term "process capability indices" for the indices listed in this section, and it uses the term "process performance indices" for the indices computed by the CAPABILITY procedure.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set:

- The label associated with the subgroup variable is used as the horizontal axis label.
- When you specify a DATA= input data set, the label associated with the process variable is used as the vertical axis label.
- Otherwise, the variable whose label is used on the vertical axis depends on whether you specify a HISTORY= or TABLE= input data set, as summarized in Table 19.83, where *Process* is the process variable name.

Table 19.83 Labeling Chart Axes

Chart Statement(s)	HISTORY= Data Set Variable	TABLE= Data Set Variable
BOXCHART, XCHART, XRCHART, XSCHART	Subgroup mean variable, *Process*X	_SUBX_
BOXCHART with CONTROLSTAT=MEDIAN	Subgroup median variable, *Process*M	_SUBMED_
CCHART	Subgroup defects per unit variable, *Process*U	_SUBC_
IRCHART	Subgroup measurement variable, *Process*	_SUBI_
MCHART, MRCHART	Subgroup median variable, *Process*M	_SUBMED_
NPCHART	Subgroup proportion nonconforming variable, *Process*P	_SUBNP_
PCHART	Subgroup proportion nonconforming variable, *Process*P	_SUBP_
RCHART	Subgroup range variable, *Process*R	_SUBR_
SCHART	Subgroup standard deviation variable, *Process*S	_SUBS_
UCHART	Subgroup defects per unit variable, *Process*U	_SUBU_

When you specify an IRCHART, MRCHART, XRCHART, or XSCHART statement, or the TRENDVAR= option in a BOXCHART, MCHART, or XCHART statement, primary and secondary charts are produced. You can provide distinct labels for the primary and secondary vertical axes by specifying a label that contains a split character in the SPLIT= option. The portion of the label before the split character labels the primary vertical axis, and the portion after the split character labels the secondary vertical axis.

For example, the following sets of statements specify the label “Avg Diameter in mm” for the vertical axis of the $\bar{X}$ chart and the label “Range in mm” for the vertical axis of the R chart:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / split = '/' ;
   label Diameter = 'Avg Diameter in mm/Range in mm';
run;

proc shewhart history=Wafersum;
   xrchart Diameter*Batch / split = '/' ;
   label DiameterX = 'Avg Diameter in mm/Range in mm';
run;

proc shewhart table=Wafertab;
   xrchart Diameter*Batch / split = '/' ;
   label _SUBX_ = 'Avg Diameter in mm/Range in mm';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

For more information, see “Labeling Axes” on page 2109.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

INSET and INSET2 Statements: SHEWHART Procedure

Overview: INSET and INSET2 Statements

The INSET and INSET2 statements enable you to enhance a Shewhart chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. The INSET statement places an inset in a primary Shewhart chart while the INSET2 statement places one in a secondary Shewhart chart. An inset can display statistics calculated by the SHEWHART procedure or arbitrary values provided in a SAS data set.

Note that an INSET or INSET2 statement by itself does not produce a display but must be used in conjunction with a chart statement. Insets are not available with line printer charts, so the INSET and INSET2 statements are not applicable when the LINEPRINTER option is specified in the PROC SHEWHART statement.

You can use options in the INSET and INSET2 statements to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

The INSET2 statement differs from the INSET statement in only two respects.

1. An INSET2 statement creates an inset within a secondary chart generated by an IRCHART, MRCHART, XRCHART or XSCHART statement or by the TRENDVAR= option. For example, when following an XRCHART statement an INSET statement produces an inset in the $\bar{X}$ chart and an INSET2 statement produces one in the R chart.
2. The INSET statement can be used to place an inset in one of the margins surrounding the plot area, while the INSET2 statement cannot.

Any of the statistics available for display in an inset can be specified with either an INSET or INSET2 statement. Descriptions of the INSET statement in this section also apply to the INSET2 statement except where explicitly noted.

Getting Started: INSET and INSET2 Statements

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the section "Syntax: INSET and INSET2 Statements" on page 1981.

Displaying Summary Statistics on a Control Chart

In the manufacture of silicon wafers, batches of five wafers are sampled, and their diameters are measured in millimeters. The following statements create a SAS data set named Wafers, which contains the measurements for 25 batches:

```
data Wafers;
   input Batch @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
 1  35.00 34.99 34.99 34.98 35.00
 2  35.01 34.99 34.99 34.98 35.00
 3  34.99 35.00 35.00 35.00 35.00
 4  35.01 35.00 34.99 34.99 35.00
 5  35.00 34.99 34.98 34.99 35.00
 6  34.99 34.99 35.00 35.00 35.00
 7  35.01 34.98 35.00 35.00 34.99
 8  35.00 35.00 34.99 34.98 34.99
 9  34.99 34.98 34.98 35.01 35.00
10  34.99 35.00 35.01 34.99 35.01
11  35.01 35.00 35.00 34.98 34.99
12  34.99 34.99 35.00 34.98 35.01
13  35.01 34.99 34.98 34.99 34.99
14  35.00 35.00 34.99 35.01 34.99
15  34.98 34.99 34.99 34.98 35.00
16  34.99 35.00 35.00 35.01 35.00
17  34.98 34.98 34.99 34.99 34.98
18  35.01 35.02 35.00 34.98 35.00
19  34.99 34.98 35.00 34.99 34.98
20  34.99 35.00 35.00 34.99 34.99
21  35.00 34.99 34.99 34.98 35.00
22  35.00 35.00 35.01 35.00 35.00
23  35.02 35.00 34.98 35.02 35.00
24  35.00 35.00 34.99 35.01 34.98
25  34.99 34.99 34.99 35.00 35.00
;
```

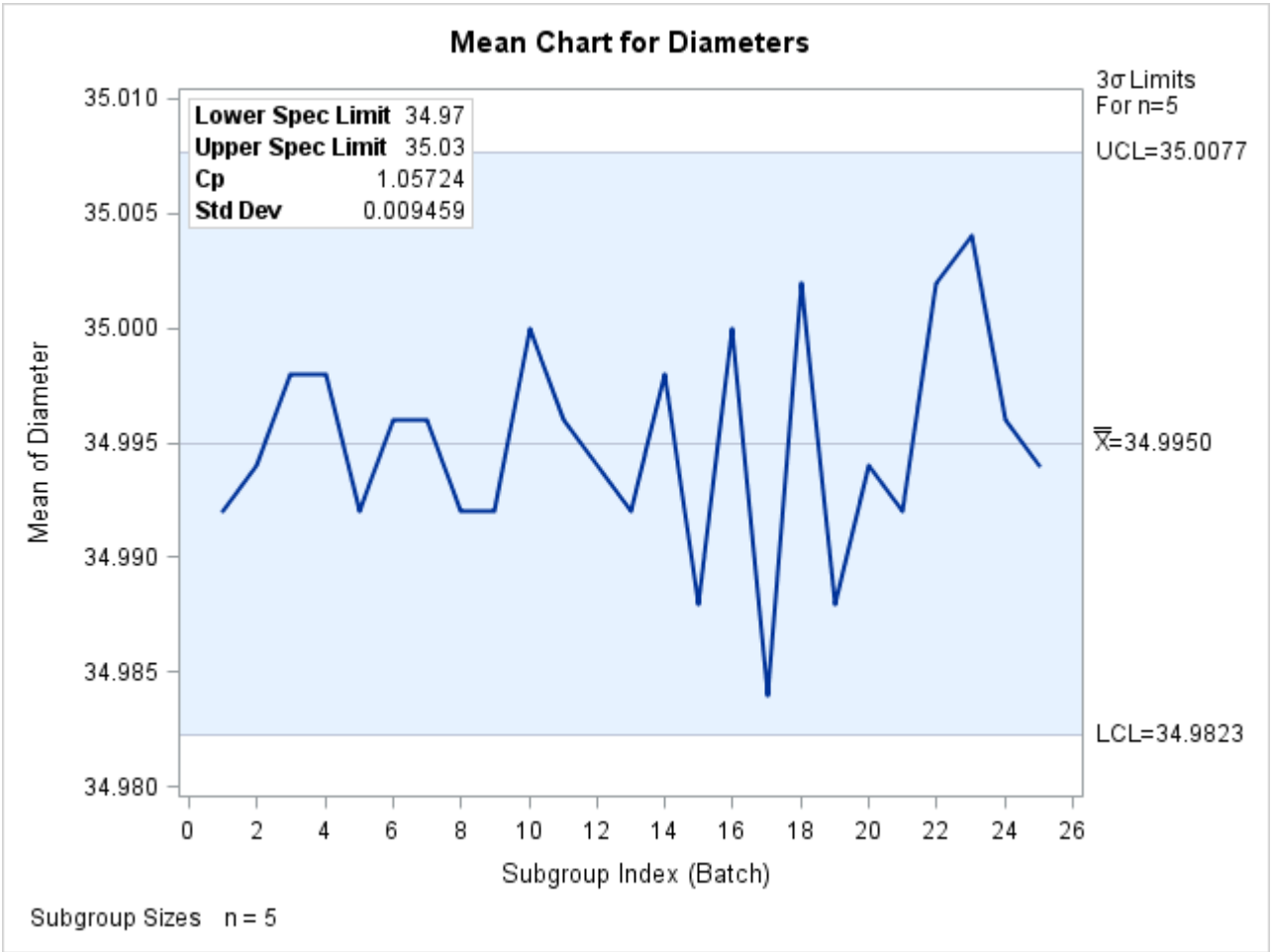
The following statements generate an $\bar{X}$ chart from the Wafers data. Lower and upper specification limits for wafer diameters are given and the process capability index C_p is computed. An INSET statement is used to display the specification limits, the computed value of C_p and the process standard deviation on the chart:

```
ods graphics on;
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch /
      lsl      = 34.97
      usl      = 35.03
      odstitle = title;
   inset lsl usl cp stddev / height = 3;
run;
```

The resulting $\bar{X}$ chart is displayed in Figure 19.122. The INSET statement immediately follows the chart statement that creates the graphical display (in this case, the XCHART statement). Specify the keywords for inset statistics (such as LSL, USL, CP and STDDEV) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords. The HEIGHT= option in the INSET statement specifies the text height used to display the statistics in the inset.

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 1983. Note that the set of keywords available for a particular display depends on both the plot statement that precedes the INSET statement and the options that you specify in the plot statement.

Figure 19.122 An $\bar{X}$ Chart with an Inset



The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you might want to provide your own labels and formats. For example, in Figure 19.122 the default format used for C_p and the process standard deviation prints an excessive number of decimal places. The following statements produce $\bar{X}$ and R charts, each with its own inset. The unwanted decimal places are eliminated and the default specification limits labels are replaced with abbreviations:

```
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xrchart Diameter*Batch /
      lsl      = 34.97
      usl      = 35.03
      odstitle = title;
   inset lsl='LSL' usl='USL' / pos = nw;
   inset2 cp (6.4) stddev (6.4) / pos = nw;
run;
```

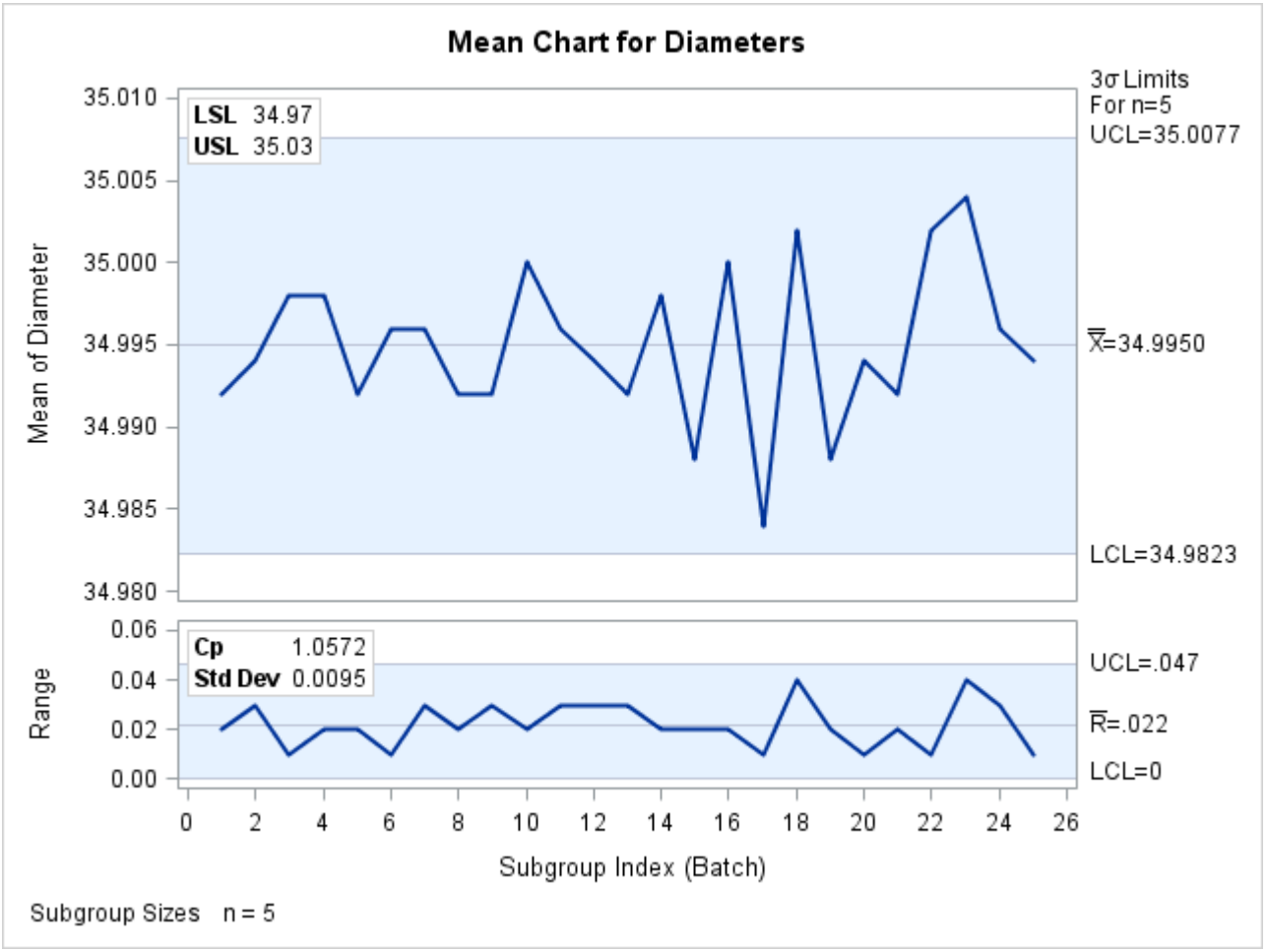
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the $\bar{X}$ and R charts are created using ODS Graphics instead of traditional graphics. The resulting charts are displayed in Figure 19.123.

You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 6.4 specified in parentheses after the CP and STDDEV keywords displays those statistics with a field width of six and four decimal places. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option. For more information about SAS formats, refer to *SAS Formats and Informats: Reference*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format.

Figure 19.123 Formatting Values and Customizing Labels in an Inset



Adding a Header and Positioning the Inset

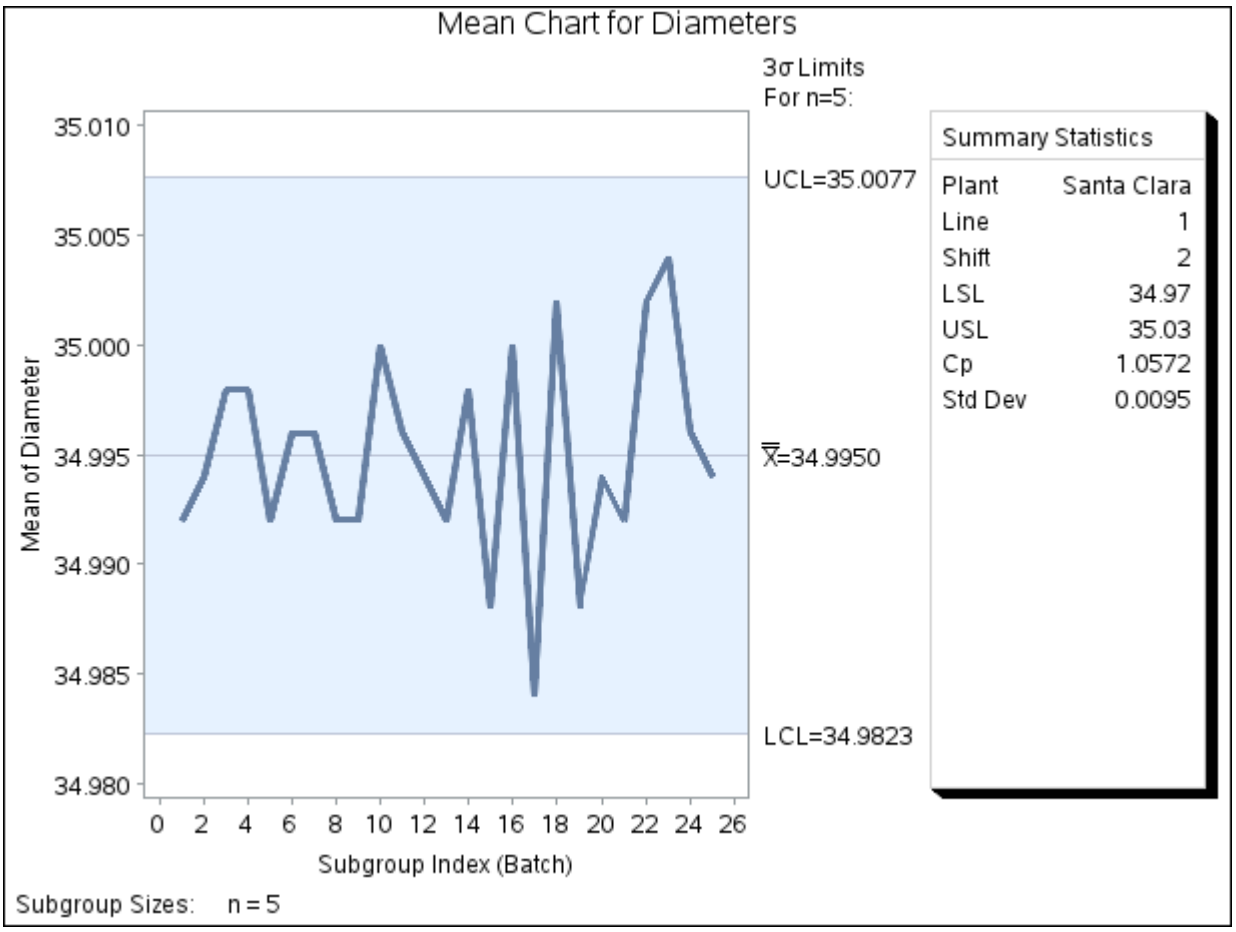
In the previous examples, the insets are displayed in the upper left corners of the plots, the default position for insets added to control charts. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and the chart shown in Figure 19.124:

```
data Location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
   datalines;
Plant     Santa Clara
Line      1
Shift     2
;

ods graphics off;
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch /
      lsl = 34.97
      usl = 35.03;
   inset data= Location lsl='LSL' usl='USL' cp (6.4) stddev (6.4) /
      position = rm
      cshadow  = black
      header   = 'Summary Statistics';
run;
```

The header (in this case, *Summary Statistics*) can be up to 40 characters. Note that a relatively long list of inset statistics is requested. Consequently, POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details: INSET and INSET2 Statements" on page 1988. The CSHADOW= option is used to display a drop shadow on this inset. The *options*, such as HEADER=, POSITION= and CSHADOW= are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 1986.

Note that the contents of the data set Location appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset.

Figure 19.124 Adding a Header and Repositioning the Inset



Syntax: INSET and INSET2 Statements

The syntax for the INSET and INSET2 statements is as follows:

INSET *keyword-list* < *options* > **;**

INSET2 *keyword-list* < *options* > **;**

You can use any number of INSET and INSET2 statements in the SHEWHART procedure. However, when ODS Graphics is enabled, at most two insets are displayed inside the primary and secondary plot areas, and at most two are displayed in the chart margins. Each INSET or INSET2 statement produces a separate inset and must follow one of the chart statements. The inset appears on every panel (page) produced by the last chart statement preceding it. The statistics are displayed in the order in which they are specified. The following statements produce a boxplot with two insets and an $\bar{X}$ and R chart with one inset in the $\bar{X}$ chart and one in the R chart.

```
proc shewhart data=Wafers;
   boxchart Diameter * Batch / lsl=34.9 target=35 usl=35.1;
      inset lsl target usl;
      inset cp cpk cpm;
```

```
   xrchart Diameter * Batch;
      inset nmin nmax nout;
      inset2 nlow2 nhigh2;
run;
```

The statistics displayed in an inset are computed for a specific process variable using observations for the current BY group. For example, in the following statements, there are two process variables (Weight and Diameter) and a BY variable (Location). If there are three different locations (levels of Location), then a total of six $\bar{X}$ charts are produced. The statistics in each inset are computed for a particular variable and location. The labels in the inset are the same for each $\bar{X}$ chart.

```
proc shewhart data=Axles;
   by Location;
   xchart (Weight Diameter) * Batch / tests=1 to 8;
   inset ntests 1 to 8;
run;
```

The components of the INSET and INSET2 statements are described as follows.

keyword-list

can include any of the *keywords* listed in "Summary of INSET Keywords" on page 1983. Some *keywords*, such as NTESTS and DATA=, require operands specified immediately after the *keyword*. Also, some inset statistics are available only if you request chart statements and options for which those statistics are calculated. For example,

- the NHIGH2, NLOW2, NTESTS2, LCL2 and UCL2 keywords are available only when a secondary chart is produced with the IRCHART, MRCHART, XRCHART or XSCHART statements.
- the NTESTS *keyword* requires the TESTS= option;
- the NTESTS2 *keyword* requires the TESTS2= option;
- the capability index *keywords* such as CPK all require one or more of the LSL=, USL= and TARGET= options.

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 1978.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
inset n nmin nmax / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 1985 for a list of all available *options*, and "Dictionary of Options" on page 1986 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

All keywords available with the SHEWHART procedure's INSET and INSET2 statements request a single statistic in an inset, except for the NTESTS, NTESTS2 and DATA= keywords. The NTESTS and NTESTS2 keywords each require a list of indexes specifying the tests for special causes whose counts of positive results are to be displayed:

```
inset ntests 1 2 3 4;
inset ntests2 1 to 4;
```

For each of the requested tests, the number of positive results for the test is displayed in the inset. So if tests 1 through 4 are requested the results occupy four lines in the inset.

The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset. Figure 19.124 shows an inset containing entries from a DATA= data set.

Table 19.84 Summary Statistics

Keyword	Description
DATA=	(Label, Value) pairs from *SAS-data-set*
LCL	Primary chart lower control limit
MEAN	Estimated or specified process mean
N	Nominal subgroup size
NMIN	Minimum subgroup size
NMAX	Maximum subgroup size
NOUT	Number of subgroups outside control limits on primary chart
NLOW	Number of subgroups below lower control limit on primary chart
NHIGH	Number of subgroups above upper control limit on primary chart
NTESTS	Number of positive results of tests for special causes on primary chart
STDDEV	Estimated or specified process standard deviation
UCL	Primary chart lower control limit

Table 19.85 Secondary Chart Summary Statistics

Keyword	Description
LCL2	Secondary chart lower control limit
MEAN2	Mean of subgroup ranges or standard deviations
NOUT2	Number of subgroups outside control limits on secondary chart
NLOW2	Number of subgroups below lower control limit on secondary chart

Table 19.85 *continued*

Keyword	Description
NHIGH2	Number of subgroups above upper control limit on secondary chart
NTESTS2	Number of positive results of tests for special causes on secondary chart
UCL2	Secondary chart upper control limit

Table 19.86 Specification Limits

Keyword	Description
LSL	Lower specification limit
USL	Upper specification limit
TARGET	Target value

Table 19.87 Capability Indices and Confidence Limits

Keyword	Description
CIALPHA	α value for computing capability index confidence limits
CP	Capability index C_p
CPLCL	Lower confidence limit for C_p
CPUCL	Upper confidence limit for C_p
CPK	Capability index C_{pk}
CPKLCL	Lower confidence limit for C_{pk}
CPKUCL	Upper confidence limit for C_{pk}
CPL	Capability index *CPL*
CPLLCL	Lower confidence limit for *CPL*
CPLUCL	Upper confidence limit for *CPL*
CPM	Capability index C_{pm}
CPMLCL	Lower confidence limit for C_{pm}
CPMUCL	Upper confidence interval for C_{pm}
CPU	Capability index *CPU*
CPULCL	Lower confidence limit for *CPU*
CPUUCL	Upper confidence limit for *CPU*

You can use the keywords in Table 19.88 only when producing ODS Graphics output. Greek letters are used in the labels for the statistics requested with the UMU and USIGMA keywords.

Table 19.88 Keywords Specific to ODS Graphics Output

Keyword	Description
TESTLEGEND	Requests a legend of positive tests for special causes
UMU	Estimated or specified process mean
USIGMA	Estimated or specified process standard deviation

Summary of Options

The following table lists the INSET and INSET2 statement options. For complete descriptions, see "Dictionary of Options" on page 1986.

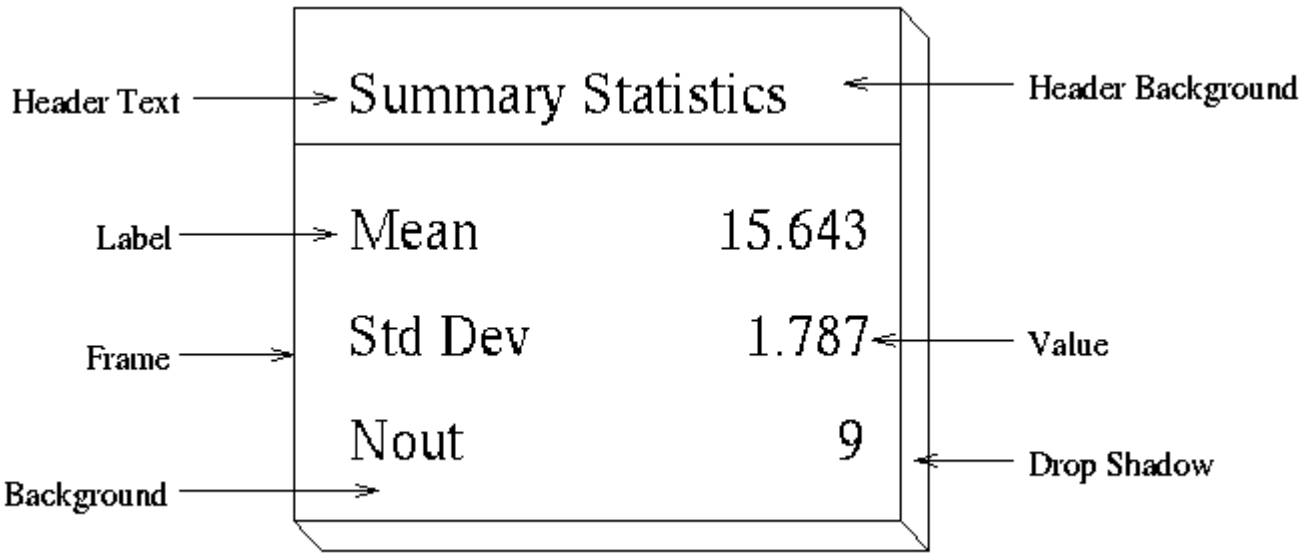
Table 19.89 INSET Options

Option	Description
CFILL=	Specifies color of inset background
CFILLH=	Specifies color of header background
CFRAME=	Specifies color of frame
CHEADER=	Specifies color of header text
CSHADOW=	Specifies color of drop shadow
CTEXT=	Specifies color of inset text
DATA	Specifies data units for POSITION=(x, y) coordinates
FONT=	Specifies font of text
FORMAT=	Specifies format of values in inset
HEADER=	Specifies header text
HEIGHT=	Specifies height of inset text
NOFRAME	Suppresses frame around inset
POSITION=	Specifies position of inset
REFPOINT=	Specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following sections provide detailed descriptions of options for the INSET and INSET2 statements. Terms used in this section are illustrated in Figure 19.125.

Figure 19.125 The Inset



General Options

You can specify the following options whether you use ODS Graphics or traditional graphics:

DATA

specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 1990. See Figure 19.128 for an example.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*

specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*

HEIGHT=SMALL

specifies the height of the text in the inset. By default, the GraphLabelText style element determines the size of inset header text and the GraphValueText style element determines the size of text in the body of the inset.

When you produce traditional graphics, you can specify the *height* in screen percent units to be used for text in both the header and the body of the inset.

When you produce ODS Graphics output, you can specify HEIGHT=SMALL to reduce the height of text in the inset. The GraphValueText size is used for the inset header and the GraphDataText size is used in the inset body.

NOFRAME
suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*
determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details: INSET and INSET2 Statements" on page 1988. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

NOTE: You cannot specify coordinates with the POSITION= option when producing ODS Graphics output.

REFPOINT=BR | BL | TR | TL
RP=BR | BL | TR | TL
specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 19.129 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 1990.

Options for ODS Graphics

You can specify the following options only when ODS Graphics is enabled:

HTRANSPARENCY=*value*
specifies the inset header background transparency when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default inset header background transparency is 0.65.

TRANSPARENCY=*value*
specifies the inset background transparency when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default inset background transparency is 0.05.

Options for Traditional Graphics

You can specify the following options only when you produce traditional graphics:

CFILL=*color* | BLANK
specifies the color of the background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as subgroup data points or control limits) show through the inset. If

you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*
specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*
specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Figure 19.124 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

FONT=*font*
specifies the font used for the text in the inset. By default, the font associated with the GraphLabelText style element is used for inset header and that associated with the GraphValueText style element is used for text in the body of the inset.

Details: INSET and INSET2 Statements

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

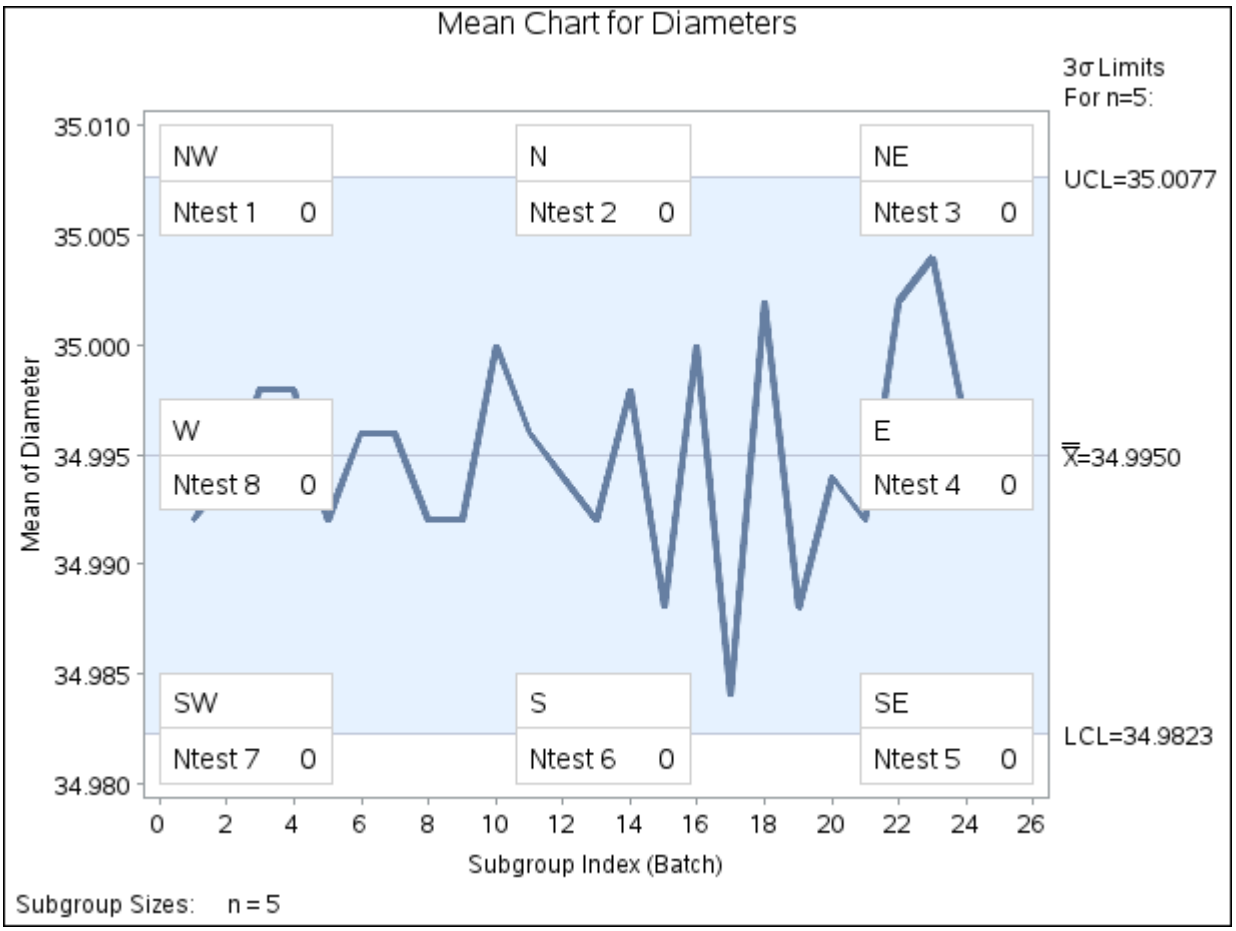
- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 19.126, which demonstrates all eight compass positions. The default is NW.

```
ods graphics off;
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch / tests= 1 to 8;
   inset ntests 1 / height=3 cfill=blank header='NW' pos=nw;
   inset ntests 2 / height=3 cfill=blank header='N ' pos=n ;
   inset ntests 3 / height=3 cfill=blank header='NE' pos=ne;
   inset ntests 4 / height=3 cfill=blank header='E ' pos=e ;
   inset ntests 5 / height=3 cfill=blank header='SE' pos=se;
   inset ntests 6 / height=3 cfill=blank header='S ' pos=s ;
   inset ntests 7 / height=3 cfill=blank header='SW' pos=sw;
   inset ntests 8 / height=3 cfill=blank header='W ' pos=w ;
run;
```

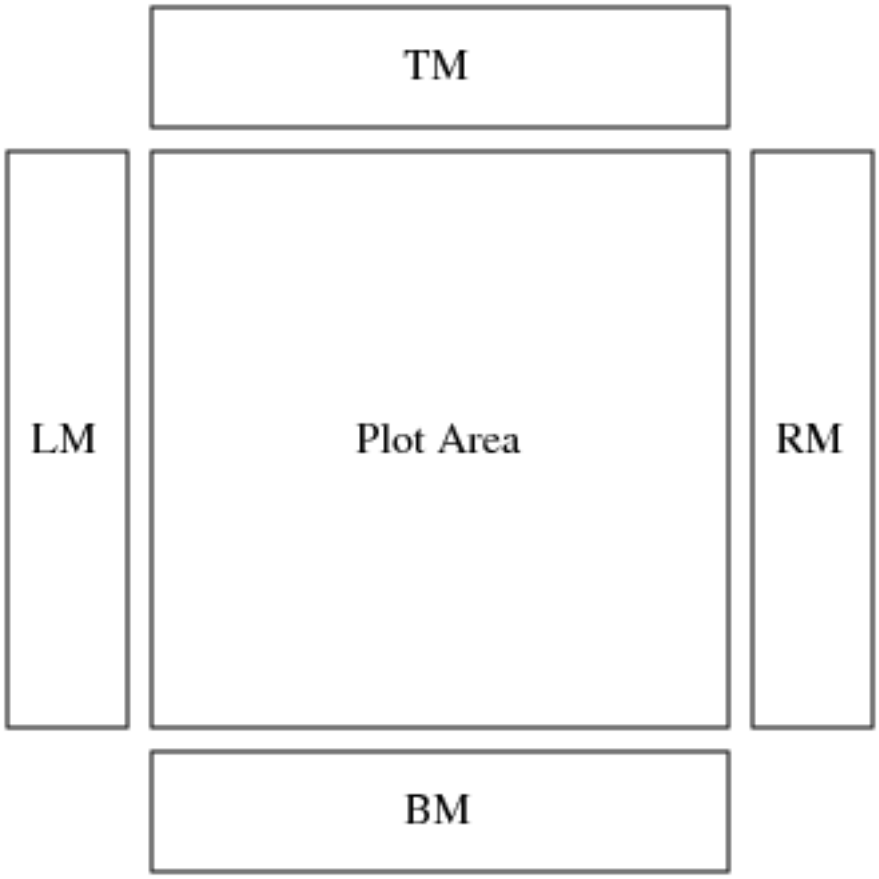
Figure 19.126 Insets Positioned Using Compass Points



Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 19.127. The INSET2 statement cannot be used to produce an inset in a margin.

Figure 19.127 Positioning Insets in the Margins



For an example of an inset placed in the right margin, see Figure 19.124. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

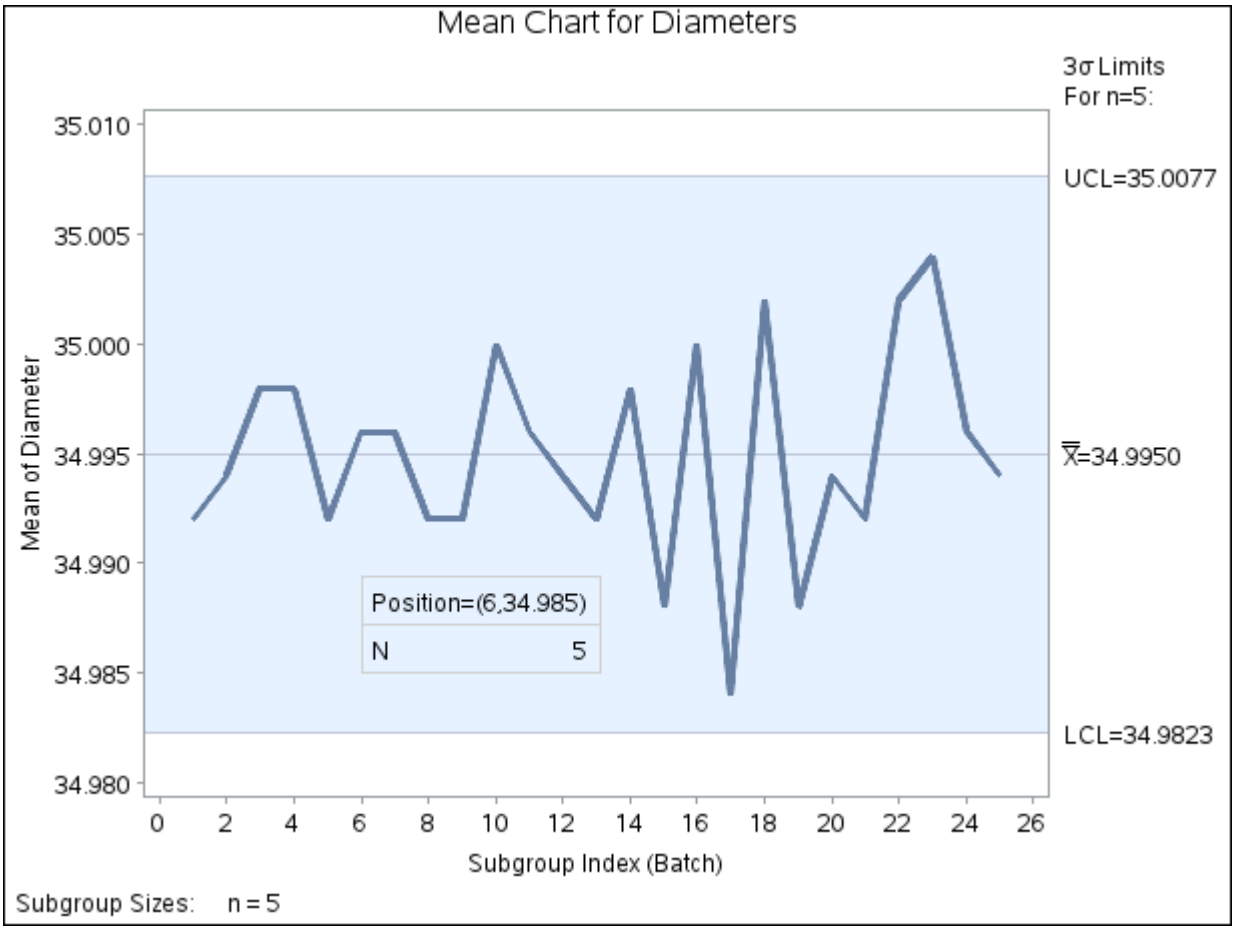
Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 6 on the horizontal axis and 34.985 on the vertical axis:

```
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch;
   inset n /
      header   = 'Position=(6,34.985)'
      position = (6,34.985) data;
run;
```

The control chart is displayed in Figure 19.128. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

Figure 19.128 Inset Positioned Using Data Unit Coordinates



Axis Percent Unit Coordinates

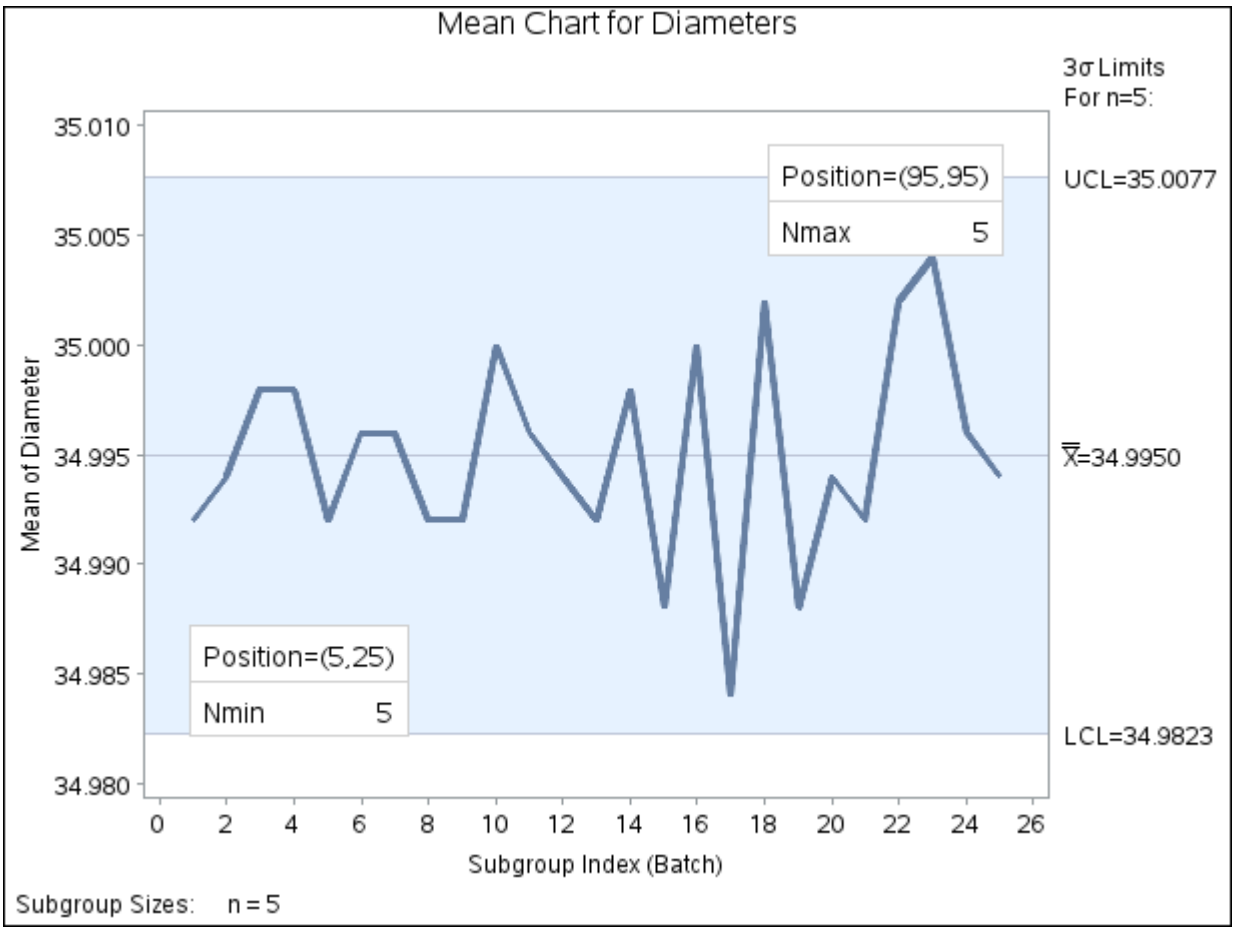
If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a $\bar{X}$ chart with two insets, both positioned using coordinates in axis percent units:

```
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch;
   inset nmin / position = (5,25)
                header   = 'Position=(5,25)'
                height   = 3
                cfill    = blank
                refpoint = tl;
   inset nmax / position = (95,95)
                header   = 'Position=(95,95)'
                height   = 3
```

```
                    cfill    = blank
                    refpoint = tr;
run;
```

The display is shown in Figure 19.129. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

Figure 19.129 Inset Positioned Using Axis Percent Unit Coordinates



Dictionary of Options: SHEWHART Procedure

The section provides detailed descriptions of options that you can specify in the following chart statements:

- BOXCHART
- CCHART
- IRCHART
- MCHART
- MRCHART
- NPCHART
- PCHART
- RCHART
- SCHART
- UCHART
- XCHART
- XRCHART
- XSCHART

Options are specified after the slash (/) in a chart statement. For example, to request tests for special causes with an $\bar{X}$ and R chart, you can use the TESTS= option as follows:

```
proc shewhart data=Measures;
   xrchart Length*Sample / tests=1 to 4 ;
run;
```

The options described in these sections are listed alphabetically. For tables of options organized by function, see the "Summary of Options" tables in the sections for the various chart statements.

Unless indicated otherwise, the options listed here are available with every chart statement. For statements that create two charts, the term *primary chart* refers to the upper chart (for instance, the $\bar{X}$ chart created with the XRCHART statement), and the term *secondary chart* refers to the lower chart (for instance, the R chart created with the XRCHART statement). The term *primary chart* also refers to the single chart created by some statements (for instance, the p chart created with the PCHART statement).

The section "General Options" on page 1994 contains descriptions of general chart statement options, which are applicable regardless of the kind of graphics output you produce. The section "Options for ODS Graphics" on page 2051 describes options that apply only when ODS Graphics is enabled. The section "Options for Traditional Graphics" on page 2056 describes options that apply only when producing traditional graphics, as when ODS Graphics is disabled. The section "Options for Legacy Line Printer Charts" on page 2070 contains descriptions of options that apply only to legacy line printer charts, which are produced when the LINEPRINTER option is specified in the PROC SHEWHART statement.

General Options

ACTUALALPHA

requests that the actual probability of a point being outside an attribute chart's probability limits be displayed in the limits legend. This probability is based on the Poisson distribution for *c* and *u* charts; it is based on the binomial distribution for *np* and *p* charts.

Because attribute chart data are discrete, it is not possible in general to compute probability limits so that the probability of a point being outside the limits is α, for any α. Therefore, the specified and actual probabilities are usually different. The actual α is the sum of the probability of a point being below the lower control limit and the probability of a point being above the upper control limit.

This option is available only in the CCHART, NPCHART, PCHART, and UCHART statements. It applies only when you request probability limits by specifying the ALPHA= option and when the probability limits are constant. By default, the α value you specify in the ALPHA= option is displayed in the limits legend.

ALLLABEL=VALUE

ALLLABEL=(*variable*)

labels every point on the primary chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. You should use this option with care to avoid cluttering the chart. By default, points are not labeled. Related options are CFRAMELAB=, OUTLABEL=, LABELFONT=, LABELHEIGHT=, and TESTLABEL=, but note that the OUTLABEL= option cannot be specified with the ALLLABEL= option.

ALLLABEL2=VALUE

ALLLABEL2=(*variable*)

labels every point on an *R*, *s*, or trend chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. You should use this option with care to avoid cluttering the chart. By default, points are not labeled. Related options are CFRAMELABN=, OUTLABEL2=, LABELFONT=, LABELHEIGHT=, and TESTLABEL2=, but note that the OUTLABEL2= option cannot be specified with the ALLLABEL2= option. The option is available in the IRCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

ALLN

plots summary statistics for all subgroups, regardless of whether the subgroup sample size equals the nominal control limit sample size n specified by the LIMITN= option or the variable _LIMITN_ in the LIMITS= data set. Use the ALLN option in conjunction with the LIMITN= option or the variable _LIMITN_.

The ALLN option is useful in applications where almost all of the subgroups have a common sample size n, and you want to display fixed (rather than varying) control limits corresponding to the nominal sample size n. The disadvantage of using the ALLN option with widely differing subgroup sample sizes is that the interpretation of the control limits is meaningful only for those subgroups whose sample size is equal to n. To request special symbol markers indicating that not all the sample sizes are equal to n, use the NMARKERS option in conjunction with the ALLN option.

The ALLN option is not available in the IRCHART statement.

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a subgroup summary statistic exceeds its control limits. This assumes that the process is in statistical control and that the data follow a certain theoretical distribution, which depends on the chart statement. The Poisson distribution is assumed for the CCHART and UCHART statements, and the binomial distribution is assumed for the NPCHART and PCHART statements. The normal distribution is assumed for all other chart statements. For the equations used to compute probability limits, see the "Details" subsection in the section for the chart statement that you are using.

The value of α can range between 0 and 1 for most statements. However, for the MCHART statement, the MRCHART statement, and the BOXCHART statement with the CONTROLSTAT=MEDIAN option, the value of α must be one of the following: 0.001, 0.002, 0.01, 0.02, 0.025, 0.04, 0.05, 0.10, or 0.20.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option. See "Input Data Sets" in the section for the chart statement in which you are interested.
- As an alternative to specifying ALPHA=α (or reading the variable _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=k (or reading the variable _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

BLOCKLABELPOS=ABOVE | LEFT | RIGHT

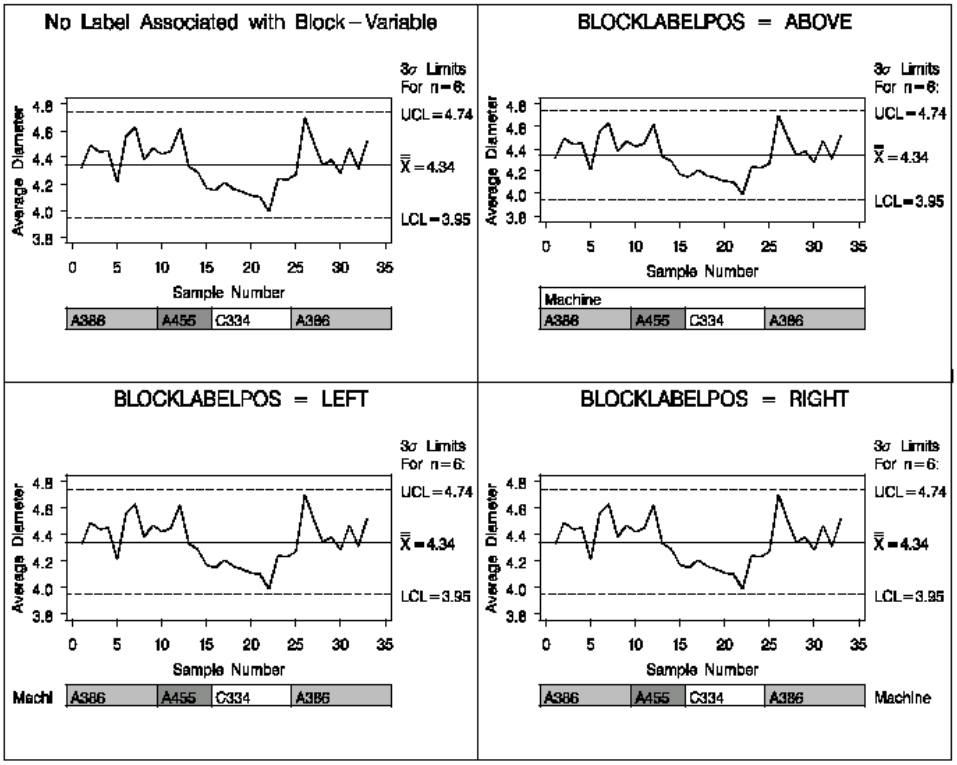
specifies the position of a block-variable label in the block legend. You can specify the following keywords, which are illustrated in Figure 19.130:

ABOVE	places the label immediately above the legend
LEFT	places the label to the left of the legend
RIGHT	places the label to the right of the legend

Use the keywords **LEFT** and **RIGHT** with labels that are short enough to fit in the margins on each side of the chart; otherwise, they will be truncated. Use the keyword **RIGHT** only when the legend is below

the control chart (BLOCKPOS=3 or BLOCKPOS=4). The default position is **ABOVE**. Related options are BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

Figure 19.130 Positions for *block-variable* Labels



BLOCKLABTYPE=SCALED | TRUNCATED | ROTATE | ROTATEALL

BLOCKLABTYPE=*height*

specifies how lengthy block variable values are treated when there is insufficient space to display them in the block legend. By default, lengthy values are not displayed.

If you specify the BLOCKLABTYPE=SCALED option, the values are uniformly reduced in height so that they fit. If you specify the BLOCKLABTYPE=TRUNCATED option, lengthy values are truncated on the right until they fit. When producing traditional graphics, you can also specify a text *height* in vertical percent screen units for the values. For ODS Graphics output, you can specify BLOCKLABTYPE=ROTATE to rotate the values of the block variable displayed closest to the chart by 90 degrees, and BLOCKLABTYPE=ROTATEALL to rotate the values of all block variables. Related options are BLOCKLABELPOS=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

NOTE: In ODS Graphics output only BLOCKLABTYPE=TRUNCATED is supported.

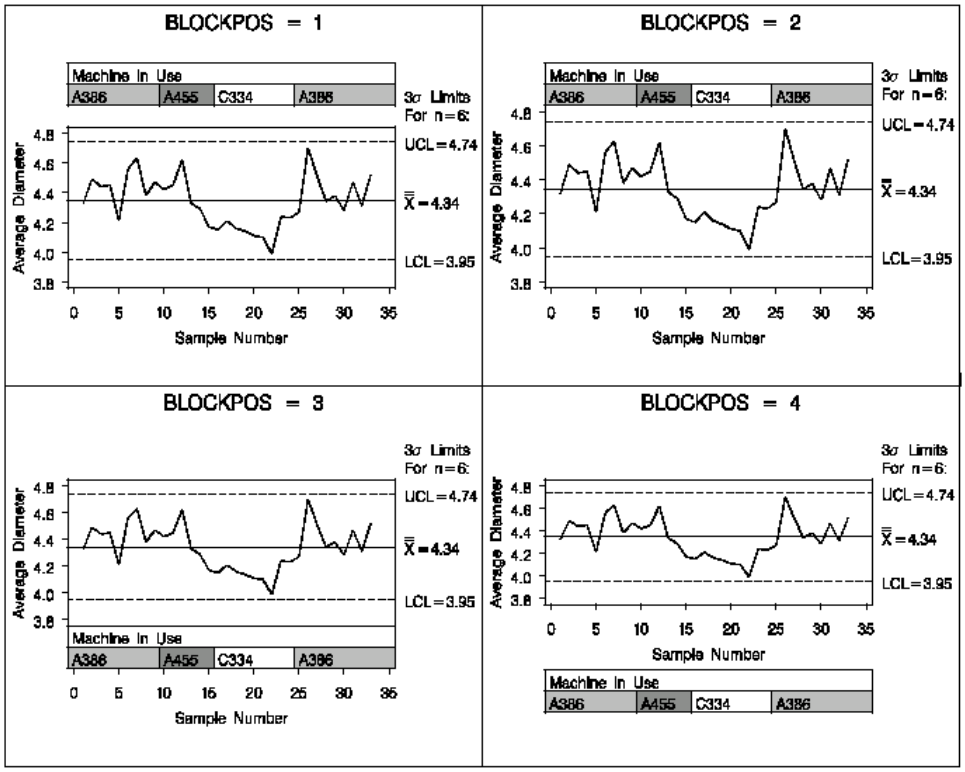
BLOCKPOS=*n*

specifies the vertical position of the legend for the values of the *block-variables* (see "Displaying Stratification in Blocks of Observations" on page 2074). Values of *n* and the corresponding positions are as follows. By default, BLOCKPOS=1.

n	**Legend Position**
1	Top of chart, offset from axis frame
2	Top of chart, immediately above axis frame
3	Bottom of chart, immediately above horizontal axis
4	Bottom of chart, below horizontal axis label

Figure 19.131 illustrates the various positions that can be specified.

Figure 19.131 Positions for *block-variable* Legends



Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

BLOCKREP

specifies that block variable values for all subgroups are to be displayed. By default, only the first block variable value in any block is displayed, and repeated block variable values are not displayed. Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=. For more information about block variables, see "Displaying Stratification in Blocks of Observations" on page 2074.

BLOCKVAR=*variable* | **(***variable-list***)**

specifies variables whose values are used to assign colors for filling the background of the legend associated with block variables. A list of BLOCKVAR= variables must be enclosed in parentheses. BLOCKVAR= variables are matched with block variables by their order in the respective variable lists. While the values of a CBLOCKVAR= variable are color names, values of a BLOCKVAR= variable are

used to group block legends for assigning fill colors from the ODS style. Block legends with the same BLOCKVAR= variable value are filled with the same color.

BOXCONNECT

BOXCONNECT=MEAN | MEDIAN | MAX | MIN | Q1 | Q3

specifies that the points representing subgroup means, medians, maximum values, minimum values, first quartiles or third quartiles in box-and-whisker plots created with the BOXCHART statement are to be connected. If BOXCONNECT is specified without a keyword identifying the points to be connected, subgroup means are connected. By default, no points are connected. The BOXCONNECT option is available only in the BOXCHART statement.

BOXES=*variable*

specifies a variable whose values are used to assign colors for the outlines of box-and-whiskers plots. While the values of a CBOXES= variable are color names, values of the BOXES= variable are used to group box-and-whiskers plots for assigning outline colors from the ODS style. The outlines of box-and-whiskers plots of groups with the same BOXES= variable value are drawn using the same color.

BOXFILL=*variable* **| NONE | EMPTY**

specifies how box-and-whisker plots are filled with colors from the ODS style. You can specify a variable whose values are used to group box-and-whiskers plots for assigning fill colors from the ODS style. Boxes associated with groups having the same BOXFILL= variable value are filled with the same color. You can specify the keyword NONE or EMPTY to produce unfilled boxes. When producing traditional graphics, you can use the CBOXFILL= option to select specific colors for filling the boxes. By default, all boxes are filled with a single color from the ODS style.

BOXSTYLE=*keyword*

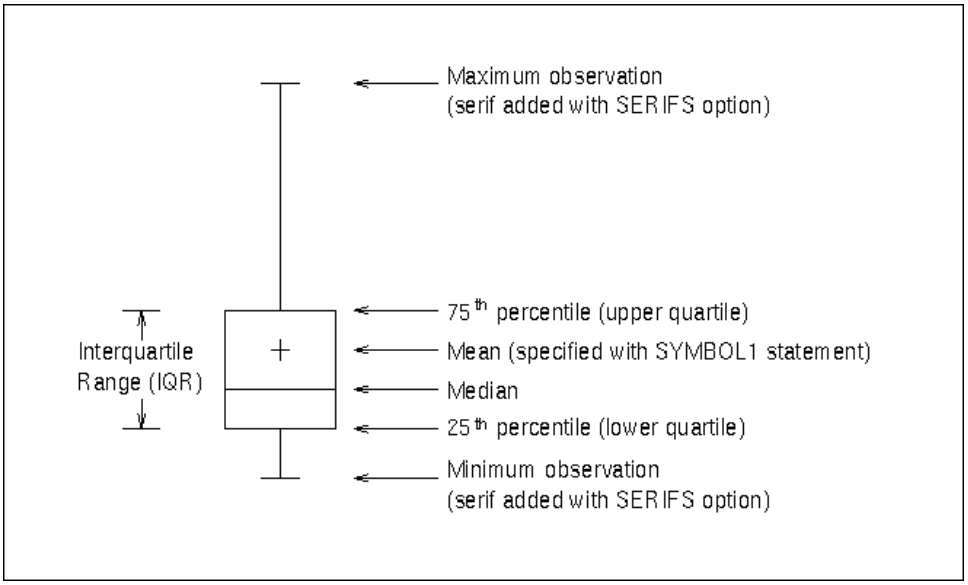
specifies the style of the box-and-whisker plots that are displayed for subgroup samples by the BOXCHART statement. You can specify the following keywords:

SKELETAL	draws whiskers to the extreme values of the subgroup
SCHEMATIC	draws whisker to the most extreme value within or equal to the lower/upper fence
SCHEMATICID	labels outliers in schematic box-and-whisker plot
SCHEMATICIDFAR	labels far outliers in schematic box-and-whisker plot
POINTS	plots the values in a subgroup as points
POINTSJOIN	plots the values in a subgroup as points joined with a vertical line
POINTSBOX	plots the values in a subgroup as points enclosed in a box
POINTSID	labels the points plotted in a subgroup
POINTSJOINID	labels the points plotted in a subgroup joined by a vertical line
POINTSSCHEMATIC	plots the values in a subgroup as points overlaid with a schematic box chart

The SKELETAL, SCHEMATIC, SCHEMATICID, and SCHEMATICIDFAR keywords are useful for creating conventional box-and-whisker displays. The keywords POINTS, POINTSJOIN, POINTSBOX, POINTSID, and POINTSJOINID are used to generalize the BOXSTYLE= option and, in particular, to facilitate the creation of so-called "multi-vari" charts, as illustrated in Output 19.7.2 and Output 19.7.3. The keyword POINTSSCHEMATIC combines the POINT and SCHEMATIC boxstyles.

If you specify BOXSTYLE=SKELETAL, the whiskers are drawn from the edges of the box to the extreme values of the subgroup sample. This plot is sometimes referred to as a *skeletal box-and-whisker plot*. By default, the whiskers are drawn without serifs, but you can add serifs with the SERIFS option. Figure 19.132 illustrates the elements of a typical skeletal boxplot.

Figure 19.132 BOXSTYLE= SKELETAL

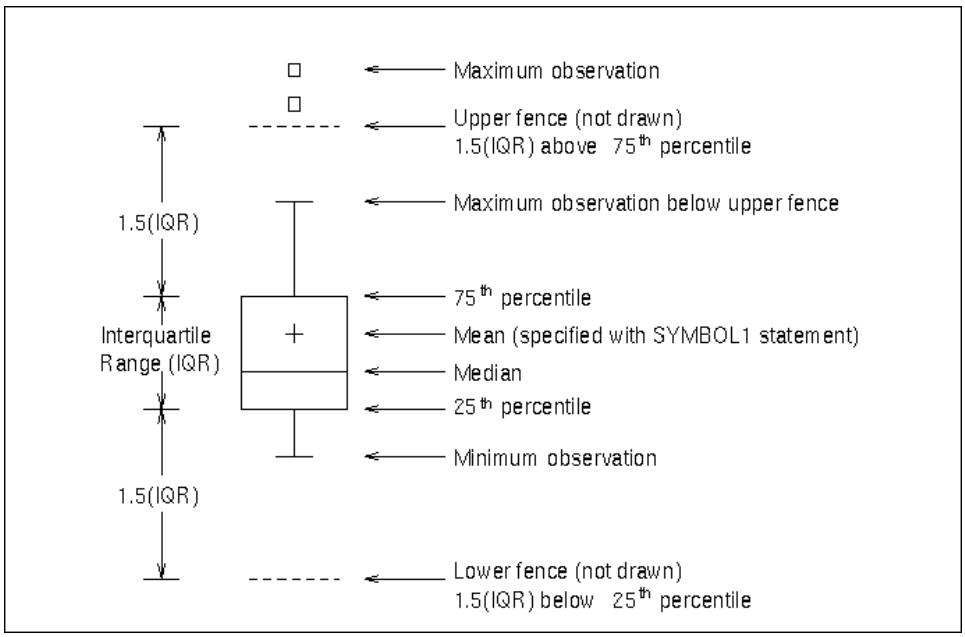


If you specify BOXSTYLE=SCHEMATIC, a whisker is drawn from the upper edge of the box to the largest value less than or equal to the upper fence and from the lower edge of the box to the smallest value greater than or equal to the lower fence. Figure 19.133 illustrates a typical schematic boxplot and the locations of the fences (which are not displayed in actual output). Serifs are added to the whiskers by default. Observations outside the fences are identified with a special symbol; you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. The default symbol is a square. This type of plot corresponds to the *schematic box-and-whisker plot* described in Chapter 2 of Tukey (1977).

If you specify BOXSTYLE=SCHEMATICID, a schematic box-and-whisker plot is displayed in which the value of the first variable listed in the ID statement is used to label the symbol marking each observation outside the upper and lower fences.

If you specify BOXSTYLE=SCHEMATICIDFAR, a schematic box-and-whisker plot is displayed in which the value of the first variable listed in the ID statement is used to label the symbol marking each observation outside the *lower* and *upper far fences*. The lower and upper far fences are located 3×IQR below the 25th percentile and above the 75th percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled with the ID variable.

Figure 19.133 BOXSTYLE= SCHEMATIC



NOTE: To make side-by-side box charts (as opposed to a control chart with subgroup box plots), you should use the BOXCHART statement with the NOLIMITS option in addition to the BOXSTYLE= option.

If you specify BOXSTYLE=POINTS, all the values in the subgroup sample are plotted as points, and neither a box nor whiskers are drawn. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option. You can specify the color of the symbols with the IDCOLOR= option (the default color is the color specified with the CBOXES= option).

If you specify BOXSTYLE=POINTSJOIN, all the values in the subgroup sample are plotted as points joined with a vertical line. Neither a box nor whiskers are drawn. See Output 19.7.2 for an illustration. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option, and you can specify the color of the symbol with the IDCOLOR= option. You can specify the color of the vertical line with the CBOXES= option.

If you specify BOXSTYLE=POINTSBOX, all the values in the subgroup sample are plotted as points enclosed in a box. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option, and you can specify the color of the symbol with the IDCOLOR= option. You can specify the color of the box with the CBOXES= option, the fill color of the box with the CBOXFILL= option, and the line type of the box with the LBOXES= option.

If you specify BOXSTYLE=POINTSID, all the values in the subgroup sample are plotted using labels specified as the values of the first variable in the ID statement. See Output 19.7.3 for an illustration. It is recommended that you use single-character labels. You can specify a font for the labels with the IDFONT= option. You can specify the height of the labels with the IDHEIGHT= option. You can specify the color of the labels with the IDCTEXT= option.

If you specify BOXSTYLE=POINTSJOINID, all the values in the subgroup sample are plotted using labels specified as the values of the first variable in the ID statement, and the values are joined by a vertical line. It is recommended that you use single-character labels. You can specify a font for the

labels with the IDFONT= option. You can specify the height of the labels with the IDHEIGHT= option. You can specify the color of the labels with the IDCTEXT= option, and you can specify the color of the vertical line with the CBOXES= option.

If you specify BOXSTYLE=POINTSSCHEMATIC, a schematic box chart is overlaid with points plotting all observations in the subgroups.

The BOXSTYLE= option is available only in the BOXCHART statement; see Example 19.2. The styles SCHEMATIC, SCHEMATICID, and SCHEMATICIDFAR are available only when the input data set is a DATA= data set. By default, BOXSTYLE= SKELETAL. Related options include BOXWIDTH=, BOXWIDTHSCALE=, IDCOLOR=, and IDSYMBOL=.

Note that the keywords POINTS, POINTSJOIN, POINTSBOX, POINTSID, and POINTSJOINID for the BOXSTYLE= option can be used in conjunction with the CPHASEBOX=, CPHASEBOXFILL=, CPHASEBOXCONNECT=, CPHASEMEANCONNECT=, and PHASEMEANSYMBOL= options to create "multi-vari" displays.

BOXWIDTH=*value*

specifies the width of box-and-whisker plots created with the BOXCHART statement. For traditional graphics, the width is specified in horizontal percent screen units. For ODS Graphics output, the width is specified in pixels. The default width is chosen so that the boxes are as wide as possible without colliding. You should use the BOXWIDTH= option in situations where the number of subgroups per panel is very small and you want to reduce the width. The BOXWIDTH= option is available only in the BOXCHART statement.

BOXWIDTHSCALE=*value*

specifies that the widths of box-and-whisker plots created with the BOXCHART statement are to vary according to a particular function of the subgroup sample size n. The function, $f(n)$, is determined by the specified *value* (≥ 0) and is identified on the chart with a legend.

If you specify a positive *value*, $f(n) = n^{value}$. In particular, if you specify BOXWIDTHSCALE=1, $f(n) = n$. If you specify BOXWIDTHSCALE=0.5, $f(n) = \sqrt{n}$, as described by McGill, Tukey, and Larsen (1978).

If you specify BOXWIDTHSCALE=0, $f(n) = \log(n)$.

The box widths vary between minimum (w_{min}) and maximum (w_{max}) widths that are determined by the output destination. The width of the ith box is

$$w_i = w_{min} + (w_{max} - w_{min}) \frac{f(n_i) - f(n_{min})}{f(n_{max}) - f(n_{min})}$$

where n_{min} is the minimum subgroup sample size and n_{max} is the maximum subgroup sample size.

By default, the box widths are constant.

The BOXWIDTHSCALE= option is available only in the BOXCHART statement. See Example 19.4 for an illustration of the BOXWIDTHSCALE= option.

CFRAMELAB=*color*

CFRAMELAB

specifies the color for filling rectangles that frame the point labels displayed with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options. Specify CFRAMELAB with no argument to produced unfilled frames. By default, the points are not framed.

CIINDICES < (< TYPE=*keyword* > < ALPHA=*value* >) >
requests capability index confidence limits based on subgroup summary data, calculated using "effective degrees of freedom" as described by Bissell (1990). These confidence limits are approximate. When you specify the CIINDICES option, the calculated confidence limits are available for display in an inset and are included in the OUTLIMITS= data set, if one is produced.

TYPE=*keyword*
specifies the type of confidence limit. Valid values are LOWER, UPPER and TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*
specifies the default confidence level to compute confidence limits. The percentage for the confidence limits is $(1 - value) * 100$. For example, ALPHA=.05 results in a 95% confidence limit. The default value is .05 and the possible range of values is from 0 to 1.

CINFILL=*color* | EMPTY | NONE
specifies the color for the area inside the upper and lower control limits. By default, this area filled with an appropriate color from the ODS style. You can specify the keyword EMPTY or NONE to leave the area between the control limits unfilled. See also the COUTFILL= option.

CLIPFACTOR=*factor*
requests clipping of extreme points on the control chart. The *factor* that you specify determines the extent to which these values are clipped, and it must be greater than one (useful values are in the range 1.5 to 2).

For examples of the CLIPFACTOR= option, see Figure 19.170 and Figure 19.171. The CLIPFACTOR= option should not be used in any statement in which the STARVERTICES= option is also used. Related clipping options are CCLIP=, CLIPCHAR=, CLIPLEGEND=, CLIPLEGPOS=, CLIPSUBCHAR=, and CLIPSYMBOL=.

CLIPLEGEND='*label*'
specifies the *label* for the legend that indicates the number of clipped points when the CLIPFACTOR= option is used. The *label* must be no more than 16 characters and must be enclosed in quotes. For an example, see Figure 19.171.

CLIPSUBCHAR='*character*'
specifies a substitution character (such as #) for the label provided with the CLIPLEGEND= option. The substitution character is replaced with the number of points that are clipped. For example, suppose that the following statements produce a chart in which three extreme points are clipped:

```
proc shewhart data=Pistons;
   xrchart Diameter*Hour /
      clipfactor  = 1.5
      cliplegend  = 'Points clipped=#'
      clipsubchar = '#' ;
run;
```

Then the clipping legend displayed on the chart will be

```
Points clipped=3
```

CONTROLSTAT=MEAN | MEDIAN

specifies whether the control limits displayed in a box chart are computed for subgroup means or for subgroup medians. By default, CONTROLSTAT=MEAN. The CONTROLSTAT= option is available only in the BOXCHART statement.

COUT=*color*

COUT

specifies the color for the plotting symbols and the portions of connecting line segments that lie outside the control limits. Specify COUT with no argument to use an appropriate contrasting color from the ODS style. This option is useful for highlighting out-of-control subgroups.

When ODS Graphics is enabled and the BOXCHART statement or STARVERTICES= option is used, COUT highlights the boxes or stars whose subgroup values fall outside the control limits.

CPHASEBOX=*color*

CPHASEBOX

PHASEBOX

specifies the color for a box that encloses all of the plotted points for a phase (group of consecutive observations that have the same value of the variable _PHASE_). Specify CPHASEBOX or PHASEBOX with no argument to request phase boxes drawn using an appropriate color from the ODS style. By default, an enclosing box is not drawn. This option is available only in the BOXCHART statement.

CPHASEBOXCONNECT=*color*

CPHASEBOXCONNECT

PHASEBOXCONNECT

specifies the color for line segments that connect the vertical edges of adjacent enclosing boxes requested with the CPHASEBOX= option or the CPHASEBOXFILL= option. The vertical coordinates of the attachment points represent the average of the values plotted inside the box. The CPHASEBOXCONNECT= option is an alternative to the CPHASEMEANCONNECT= option. Specify CPHASEBOXCONNECT or PHASEBOXCONNECT with no argument to connect the phase boxes with lines drawn in an appropriate color from the ODS style. This option is available only in the BOXCHART statement.

CPHASEBOXFILL=*color*

CPHASEBOXFILL

PHASEBOXFILL

specifies the fill color for a box that encloses all of the plotted points for a phase. Specify CPHASEBOXFILL or PHASEBOXFILL with no argument to fill the phase boxes with an appropriate color from the ODS style. By default, an enclosing box is not drawn. This option is available only in the BOXCHART statement.

CPHASEMEANCONNECT=*color*

CPHASEMEANCONNECT

PHASEMEANCONNECT

specifies the color for line segments that connect points representing the average of the values plotted within a phase. This option must be used in conjunction with the CPHASEBOX= or PHASEBOXFILL= options, and it is an alternative to the CPHASEBOXCONNECT= option. The points are centered horizontally within the enclosing boxes. Specify CPHASEMEANCONNECT or PHASEMEANCONNECT with no argument to connect phase means with lines drawn in an appropriate color from the ODS style. This option is available only in the BOXCHART statement.

CSTAROUT=*color*

CSTAROUT

specifies a color for those portions of the outlines of stars (requested with the STARVERTICES= option) that exceed the inner or outer circles. This option applies only with the STARTYPE=RADIAL and STARTYPE=SPOKE options, and it is useful for highlighting extreme values of star vertex variables. Specify CSTAROUT with no argument to use an appropriate contrasting color from the ODS style. See "Displaying Auxiliary Data with Stars" on page 2090.

CSYMBOL='*label***'**

CSYMBOL=C | CBAR | CPM | CPM2 | C0

specifies a label for the central line in a *c* chart. You can use the option in two ways:

- You can specify a quoted *label* of length 16 or less.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
C	C	C
CBAR	$\overline{C}$	$\overline{C}$
CPM	C'	C'
CPM2	C''	C''
C0	C_0	C0

See Example 19.9 for an example. The default keyword is CBAR. The CSYMBOL= option is available only in the CCHART statement.

DATAUNIT=PERCENT | PROPORTION

enables you to use percents or proportions as the values for *processes* when you are using the PCHART or NPCHART statements and reading a DATA= input data set. Specify DATAUNIT=PERCENT to indicate that the values are percents of nonconforming items. Specify DATAUNIT=PROPORTION to indicate that the values are proportions of nonconforming items. Values for percents can range from 0 to 100, while values for proportions can range from 0 to 1. By default, the values of *processes* read from a DATA= data set for PCHART and NPCHART statements are assumed to be numbers (counts) of nonconforming items. The DATAUNIT= option is available only in the NPCHART and PCHART statements.

DISCRETE
specifies that numeric subgroup variable values be treated as discrete values, so that each tick value on the default subgroup axis corresponds to a unique subgroup variable value. By default, a continuous subgroup axis is created, and if the subgroup variable values are not evenly spaced, the axis contains ticks with no corresponding subgroup data.

EXCHART
creates a control chart only when exceptions occur, specifically, when the control limits are exceeded or when any of the tests requested with the TESTS= option or the TESTS2= option are positive.

FRONTREF
draws reference lines specified with the HREF= and VREF= options in front of box-and-whiskers plots. By default, reference lines are drawn behind the box-and-whiskers plots and can be obscured by filled boxes.

GRID
adds a grid to the control chart. Grid lines are horizontal and vertical lines positioned at labeled major tick marks, and they cover the length and height of the plotting area.

HAXIS=*values*
HAXIS=AXIS*n*
specifies tick mark values for the horizontal (subgroup) axis. If the subgroup variable is numeric, the *values* must be numeric and equally spaced. Numeric values can be given in an explicit or implicit list. If the subgroup variable is character, *values* must be quoted strings of length 32 or less. If a date, time, or datetime format is associated with a numeric subgroup variable, SAS datetime literals can be used. Examples of HAXIS= lists follow:

```
haxis=0 2 4 6 8 10
haxis=0 to 10 by 2
haxis='LT12A' 'LT12B' 'LT12C' 'LT15A' 'LT15B' 'LT15C'
haxis='20MAY88'D to '20AUG88'D by 7
haxis='01JAN88'D to '31DEC88'D by 30
```

If the subgroup variable is numeric, the HAXIS= list must span the subgroup variable values, and if the subgroup variable is character, the HAXIS= list must include all of the subgroup variable values. You can add subgroup positions to the chart by specifying HAXIS= values that are not subgroup variable values.

If you specify a large number of HAXIS= values, some of these might be thinned to avoid collisions between tick mark labels. To avoid thinning, use one of the following methods:

- Shorten values of the subgroup variable by eliminating redundant characters. For example, if your subgroup variable has values LOT1, LOT2, LOT3, and so on, you can use the SUBSTR function in a DATA step to eliminate "LOT" from each value, and you can modify the horizontal axis label to indicate that the values refer to lots.
- Use the TURNHLABELS option to turn the labels vertically.
- Use the NPANELPOS= option to force fewer subgroup positions per panel.

If you are producing traditional graphics, you can also specify a previously defined AXIS statement with the HAXIS= option.

HOFFSET=*value*

specifies the length of the offset at each end of the horizontal axis. For traditional graphics, the offset is specified in percent screen units. For ODS Graphics output, the offset is specified in pixels. You can eliminate the offset by specifying HOFFSET=0.

HREF=*values*

HREF=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the primary chart. You can use this option in the following ways:

- You can specify the *values* for the lines with an HREF= list. If the subgroup variable is numeric, the *values* must be numeric. If the subgroup variable is character, the *values* must be quoted strings of up to 32 characters. If the subgroup variable is formatted, the *values* must be given as internal values.

  Examples of HREF=*values* follow:

```
href=5
href=5 10 15 20 25 30
href='Shift 1' 'Shift 2' 'Shift 3'
```

- You can specify the values for the lines as the values of a variable named _REF_ in an HREF= data set. The type and length of _REF_ must match those of the *subgroup variable* specified in the chart statement. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in charts for different *processes* specified in the chart statement, you must include a character variable of length 32 or less named _VAR_, whose values are the *processes*. If you do not include the variable _VAR_, all of the lines are displayed in all of the charts.

  Each observation in the HREF= data set corresponds to a reference line. If BY variables are used in the input data set (DATA=, HISTORY=, or TABLE=), the same BY variable structure must be used in the HREF= data set unless you specify the NOBYREF option.

Related options are CHREF=, HREFCHAR=, HREFLABELS=, HREFLABPOS=, LHREF=, and NOBYREF.

HREF2=*values*

HREF2=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the secondary chart. The conventions for specifying the HREF2= option are identical to those for specifying the HREF= option. Related options are CHREF=, HREFCHAR=, HREF2LABELS=, HREFLABPOS=, LHREF=, and NOBYREF. The HREF2= option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

HREF2DATA=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the secondary chart. The HREF2DATA= option must be used in place of the HREF2= option to specify a data set using the quoted filename notation.

HREF2LABELS=‘*label1*’ **...** ‘*labeln*’

HREF2LABEL=‘*label1*’ **...** ‘*labeln*’

HREF2LAB=‘*label1*’ **...** ‘*labeln*’

specifies labels for the reference lines requested by the HREF2= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. The HREF2LABELS= option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

HREFDATA=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the primary chart. The HREFDATA= option must be used in place of the HREF= option to specify a data set using the quoted filename notation.

HREFLABELS=‘*label1*’ **...** ‘*labeln*’

HREFLABEL=‘*label1*’ **...** ‘*labeln*’

HREFLAB=‘*label1*’ **...** ‘*labeln*’

specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

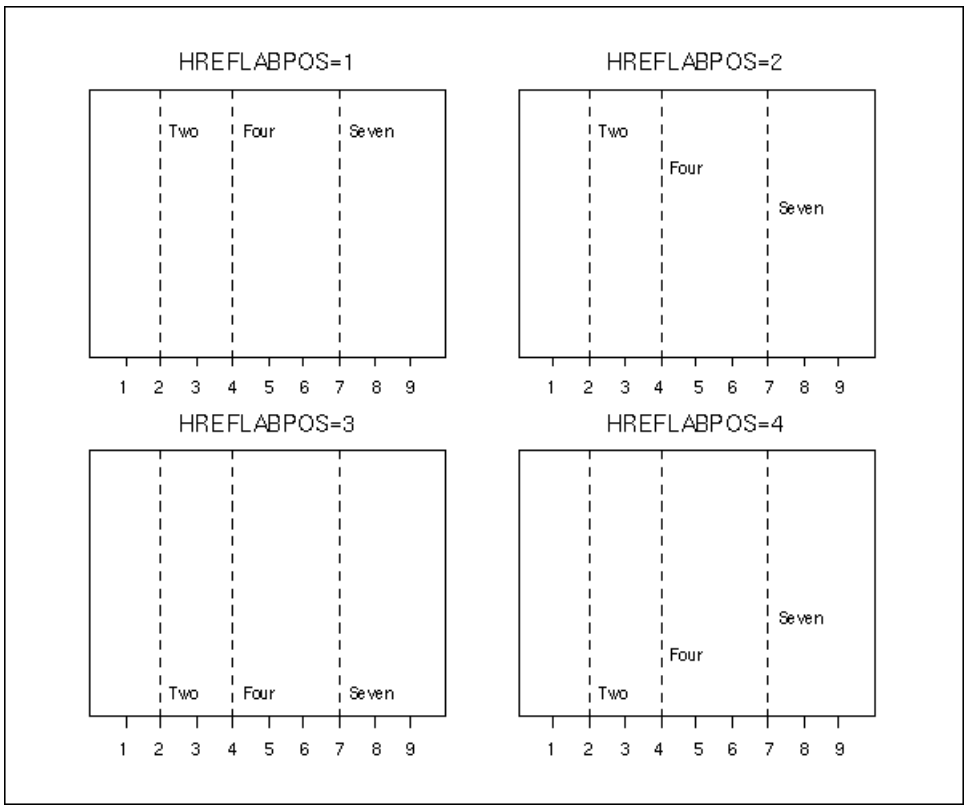
specifies the vertical position of the HREFLABEL= and HREF2LABEL= labels, as described in the following table. By default, $n = 2$.

n	**Position**
1	along top of subplot area
2	staggered from top to bottom of subplot area
3	along bottom of subplot area
4	staggered from bottom to top of subplot area

Figure 19.134 illustrates label positions for values of the HREFLABPOS= option when the HREF= and HREFLABELS= options are as follows:

```
href       = 2 4 7
hreflabels = 'Two' 'Four' 'Seven'
```

Figure 19.134 Positions for Reference Line Labels



INDEPENDENTZONES
INDEPZONES

specifies that the widths of the zones requested with the ZONES option be computed independently above and below the center line of the chart, so that the width of each zone is one-third of the difference between the process mean and the control limit on its side of the chart. By default, the width of all zones is one-third of the difference between the upper control limits and the process mean, with zones below the center line truncated if necessary. The INDEPENDENTZONES option has no effect when the control limits are symmetric.

INTERVAL=DAY | DTDAY | HOUR | MINUTE | MONTH | QTR | SECOND

specifies the natural time interval between consecutive subgroup positions when a time, date, or datetime format is associated with a numeric subgroup variable. By default, the INTERVAL= option uses the number of subgroup positions per panel that you specify with the NPANELPOS= option. The default time interval keywords for various time formats are shown in the following table.

Format	Default Keyword	Format	Default Keyword
DATE	DAY	MONYY	MONTH
DATETIME	DTDAY	TIME	SECOND
DDMMYY	DAY	TOD	SECOND
HHMM	HOUR	WEEKDATE	DAY
HOUR	HOUR	WORDDATE	DAY
MMDDYY	DAY	YYMMDD	DAY
MMSS	MINUTE	YYQ	QTR

You can use the INTERVAL= option to modify the effect of the NPANELPOS= option, which specifies the number of subgroup positions per panel (screen or page). The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the subgroup variable without having to associate a different format with the subgroup variable.

For example, suppose your formatted subgroup values span an overall time interval of 100 days and a DATETIME format is associated with the subgroup variable. Because the default interval for the DATETIME format is DTDAY and because NPANELPOS=50 by default, the chart is displayed with two panels (screens or pages).

Now, suppose your data span an overall time interval of 100 hours and a DATETIME format is associated with the subgroup variable. The chart for these data are created in a single panel, but the data occupy only a small fraction of the chart because the scale of the data (hours) does not match that of the horizontal axis (days). If you specify INTERVAL=HOUR, the horizontal axis is scaled for 50 hours, matching the scale of the data, and the chart is displayed with two panels.

INTSTART=*value*

specifies the starting value for a numeric horizontal axis, when a date, time, or datetime format is associated with the subgroup variable. If the value specified is greater than the first subgroup variable value, this option has no effect.

LCLLABEL='*label*'

specifies a label for the lower control limit in the primary chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form *LCL=value* if the control limit has a fixed value; otherwise, the default label is *LCL*. Related options are LCLLABEL2=, UCLLABEL=, and UCLLABEL2=.

LCLLABEL2='*label*'

specifies a label for the lower control limit in the secondary chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form *LCL=value* if the control limit has a fixed value; otherwise, the default label is *LCL*. The LCLLABEL2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements. Related options are LCLLABEL=, UCLLABEL=, and UCLLABEL2=.

LIMITN=*n*

LIMITN=VARYING

specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=*n*, the control limits are computed for the fixed value *n*, and they do not vary with the subgroup sample sizes. Moreover, subgroup summary statistics are plotted *only* for those subgroups with a sample size equal to *n*. You can specify ALLN in conjunction with LIMITN=*n* to force all of the statistics to be plotted, regardless of subgroup sample size.

If you do not specify LIMITN=*n* and the subgroup sample sizes are constant, the default value of *n* is the constant subgroup sample size.

Depending on the chart statement, there are restrictions on the value of *n* that you can specify with the LIMITN= option. For the MRCHART, RCHART, and XRCHART statements, $2 \leq n \leq 25$. For the SCHART and XSCHART statements, $n \geq 2$. For the BOXCHART, MCHART, and XCHART statements, $n \geq 1$. If you omit the STDDEVIATIONS option for the MCHART or XCHART statements (or use the RANGES option with the BOXCHART statement) $n < 26$. For the CCHART

and UCHART statements, $n > 0$, and n can assume fractional values (for all other chart statements, n must be a whole number). For the PCHART and NPCHART statements, $n \geq 1$.

For the IRCHART statement, n has a somewhat different interpretation; it specifies the number of consecutive measurements from which the moving ranges are to be computed, and $n \geq 2$. You can think of n as a *pseudo* nominal sample size for the control limits, because the data for an individual measurements and moving range chart are not subgrouped.

Note the difference between the LIMITN= option and the SUBGROUPN= option that is available in the CCHART, NPCHART, PCHART, and UCHART statements. The LIMITN= option specifies a nominal sample size for the *control limits*, whereas the SUBGROUPN= option provides the sample sizes for the *data*.

By default, LIMITN=2 in an IRCHART statement. You cannot specify LIMITN= VARYING in an IRCHART statement. For all other chart statements, LIMITN= VARYING is the default.

The following table identifies the chart features that vary when you use LIMITN= VARYING:

Chart Statement	Features Affected by LIMITN=VARYING
BOXCHART	Control limits
CCHART	Control limits, central line
MCHART	Control limits
MRCHART	Control limits on both charts, central line on *R* chart
NPCHART	Control limits, central line
PCHART	Control limits
RCHART	Control limits, central line
SCHART	Control limits, central line
UCHART	Control limits
XCHART	Control limits
XRCHART	Control limits on both charts, central line on *R* chart
XSCHART	Control limits on both charts, central line on *s* chart

NOTE: As an alternative to specifying the LIMITN= option, you can read the nominal control limit sample size from the variable _LIMITN_ in a LIMITS= data set. See "Input Data Sets" in the section for the chart statement in which you are interested.

LIMLABSUBCHAR='*character***'**

specifies a substitution character (such as #) for labels provided as quoted strings with the LCLLABEL=, LCLLABEL2=, UCLLABEL=, UCLLABEL2=, CSYMBOL=, NPSYMBOL=, PSYMBOL=, RSYMBOL=, SSYMBOL=, USYMBOL=, and XSYMBOL= options. The substitution character must appear in the label. When the label is displayed on the chart, the character is replaced with the value of the corresponding control limit or center line, provided that this value is constant across subgroups. Otherwise, the default label for a varying control limit or center line is displayed.

LSL=*value-list*

provides lower specification limits used to compute capability indices. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

The SHEWHART procedure uses the specification limits to compute capability indices, and it saves the limits and indices in the OUTLIMITS= data set. For more information, see "Capability Indices" on page 1971 and "Output Data Sets" in the section for the chart statement in which you are interested. Also see the entry for the USL= option. The LSL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

LTMARGIN=*value*

LTM=*value*

specifies the width of the left marginal area for the plot requested with the LTMPLOT= option. For traditional graphics, the width is specified in horizontal percent screen units. For ODS Graphics output, the width is specified in pixels. The LTMARGIN= option is available only in the IRCHART statement.

LTMPLOT=*keyword*

requests a univariate plot of the control chart statistics that is positioned in the left margin of the control chart. The keywords that you can specify and the associated plots are listed in the following table:

Keyword	Marginal Plot
HISTOGRAM	Histogram
DIGIDOT	Digidot plot
SKELETAL	Skeletal box-and-whisker plot
SCHEMATIC	Schematic box-and-whisker plot
SCHEMATICID	Schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	Schematic box-and-whisker plot with far outliers labeled

NOTE: Digidot plots are not available in ODS Graphics output.

The LTMPLOT= option is available only in the IRCHART statement; see Example 19.13 for an example. Refer to Hunter (1988) for a description of digidot plots, and see the entry for the BOXSTYLE= option for a description of the various box-and-whisker plots. Related options are LTMARGIN=, RTMARGIN=, and RTMPLOT=.

MAXPANELS=*n*

specifies the maximum number of pages or screens for a chart. By default, $n = 20$.

MEDCENTRAL=AVGMEAN | AVGMED | MEDMED

identifies a method for estimating the process mean μ, which is represented by the central line on a median chart. The methods corresponding to each keyword are given in the following table:

Keyword	Method for Estimating Process Mean
AVGMEAN	Average of subgroup means
AVGMED	Average of subgroup medians
MEDMED	Median of subgroup medians

The default keyword is AVGMED. The MEDCENTRAL= option is available only in the MCHART and MRCHART statements and in the BOXCHART statement with the CONTROLSTAT=MEDIAN option.

MISSBREAK

determines how subgroups are formed when observations are read from a DATA= data set and a character *subgroup-variable* is provided. When you specify the MISSBREAK option, observations with missing values of the *subgroup variable* are not processed. Furthermore, the next observation with a nonmissing value of the *subgroup-variable* is treated as the beginning observation of a new subgroup even if this value is identical to the most recent nonmissing subgroup value. In other words, by specifying the option MISSBREAK and by inserting an observation with a missing *subgroup-variable* value into a group of consecutive observations with the same *subgroup-variable* value, you can split the group into two distinct subgroups of observations.

By default, if MISSBREAK is not specified, observations with missing values of the *subgroup variable* are not processed, and all remaining observations with the same consecutive value of the *subgroup-variable* are treated as a single subgroup.

MRRESTART

MRRESTART=*value*

causes the moving range computation on the IRCHART to be restarted when a missing value is encountered. Without the MRRESTART option, a missing value is simply skipped, and the moving range for the next nonmissing subgroup is computed using the most recent previous nonmissing value. MRRESTART restarts the moving range computation, so only the observations after the missing value are used in subsequent moving range computations. MRRESTART restarts the moving range computation on any missing value; you can also specify MRRESTART=value to restart only on a particular missing value. For example, MRRESTART=R will restart the computation only when the missing value ".R" is encountered.

MU0=*value*

specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data. The MU0= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, XCHART, XRCHART, and XSCHART statements.

NOTE: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set. See "Input Data Sets" in the section for the chart statement in which you are interested.

NDECIMAL=*n*

specifies the number of decimal digits in the default labels for the control limits and the central line in the primary chart. The default is one more than the maximum number of decimal digits in the vertical axis tick mark labels. For example, if the vertical axis tick mark label with the largest number of digits after the decimal point is 110.05, the default is $n = 3$.

NDECIMAL2=*n*

specifies the number of decimal digits in the default labels for the control limits and central line in a secondary chart. The default is one more than the maximum number of decimal digits in the vertical axis tick mark labels. The NDECIMAL2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NEEDLES

connects plotted points to the central line with vertical line segments (needles). See Example 19.19 for an example. By default, adjacent points are connected to one another. The NEEDLES option is available in all chart statements except the BOXCHART statement.

NMARKERS

identifies a plotted subgroup summary statistic with a special symbol marker (character) when the corresponding subgroup sample size is not equal to the nominal control limit sample size n. Specify the nominal control limit sample size n with the LIMITN= option or with the variable _LIMITN_ read from a LIMITS= data set. The following table summarizes the identification:

Sample Size	Graphics Symbol	Line Printer Character
$< n$	∇	L
$> n$	$\triangle$	G

A legend that explains the symbols is displayed at the bottom of the chart. This legend can be suppressed with the NOLEGEND option.

The NMARKERS option is not available in the IRCHART statement. The NMARKERS option applies only when specified in conjunction with the ALLN option and a fixed nominal control limit sample size provided with the LIMITN= option or the variable _LIMITN_. See Example 19.40 for an illustration.

NO3SIGMACHECK

suppresses the check for 3σ limits when tests for special causes are requested. This enables tests for special causes to be applied when the SIGMAS= option is used to specify control limits other than the default 3σ limits. This option should not be used for standard control chart applications, because the standard tests for special causes assume 3σ limits.

NOBYREF

specifies that the reference line information in an HREF=, HREF2=, VREF=, or VREF2= data set is to be applied uniformly to charts created for all the BY groups in the input data set (DATA=, HISTORY=, or TABLE=). If you specify the NOBYREF option, you do not need to provide BY variables in the reference line data set. By default, you must provide BY variables.

NOCHART

suppresses the creation of the chart. You typically specify the NOCHART option when you are using the procedure to compute control limits and save them in an output data set. You can also use the NOCHART option when you are tabulating results with the TABLE and related options.

In the IRCHART, MRCHART, XRCHART, and XSCHART statements, the NOCHART option suppresses the creation of both the primary and secondary charts. If you are producing traditional graphics and specify the NOCHART option, the chart is not saved in a graphics catalog. To save the chart in a graphics catalog while suppressing the display of the chart, specify the NODISPLAY option in a GOPTIONS statement.

NOCHART2

suppresses the creation of a secondary chart. You typically use this option in the IRCHART statement to create a chart for individual measurements and suppress the accompanying chart for moving ranges. The NOCHART2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOCONNECT

suppresses line segments that connect points on the chart. By default, points are connected except in box charts produced with the BOXCHART statement (see the BOXCONNECT option).

NOCTL
suppresses the display of the central line in a primary chart.

NOCTL2
suppresses the display of the central line in a secondary chart. The NOCTL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOHLABEL
suppresses the label for the horizontal (subgroup) axis. Use the NOHLABEL option when the meaning of the axis is evident from the tick mark labels, such as when a date format is associated with the subgroup variable.

NOLCL
suppresses the display of the lower control limit in a primary chart.

NOLCL2
suppresses the drawing of the lower control limit in a secondary chart. The NOLCL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOLEGEND
suppresses the default legend for subgroup sample sizes, which appears by default below the chart. This option also suppresses the legend displayed by the NMARKERS option. Use the NOLEGEND option when the subgroup sample sizes are constant and equal to the control limit sample size, because the control limit sample size is displayed in the upper right corner of the chart.

NOLIMIT0
suppresses the display of a fixed lower control limit if and only if the value of the limit is zero. This option is useful in situations where a lower limit of zero is considered to be uninformative or visually distracting (for instance, on certain p charts or R charts). The NOLIMIT0 option is available with all chart statements except BOXCHART, MCHART, and XCHART. For the IRCHART, MRCHART, XRCHART, and XSCHART statements, the NOLIMIT0 option applies only to the secondary chart.

NOLIMIT1
suppresses the display of a fixed upper control limit on a p chart if and only if the value of the control limit is 1 (or 100%), or on an np chart if and only if the value of the control limit is n. The NOLIMIT1 option is available only in the NPCHART and PCHART statements.

NOLIMITLABEL
suppresses the default labels for the control limits and central lines.

NOLIMITS
suppresses the display of control limits. This option is particularly useful if you are using the BOXCHART statement to create side by side box-and-whisker plots; in this case, you should also use one of the BOXSTYLE= options.

NOLIMITSLEGEND
suppresses the legend for the control limits (for example, *3σ Limits For n=5*), which appears by default in the upper right corner of the chart.

NOOVERLAYLEGEND

suppresses the legend for overlay variables which is displayed by default when the OVERLAY= or OVERLAY2= option is specified.

NOREADLIMITS

specifies that the control limits for each *process* listed in the chart statement *not* be read from the LIMITS= data set specified in the PROC SHEWHART statement. There are two basic methods of displaying control limits: calculating control limits from the data and reading control limits from a LIMITS= data set. If you want control limits calculated from the data, you can do one of the following:

1. Do not specify a LIMITS= data set.
2. If you specify a LIMITS= data set, also specify the NOREADLIMITS option.

Otherwise, if you specify a LIMITS= data set in the PROC SHEWHART statement, the procedure reads control limits from that data set.

The following example illustrates the NOREADLIMITS option:

```
proc shewhart data=Pistons limits=Diamlim;
   xrchart Diameter*Hour;
   xrchart Diameter*Hour / noreadlimits;
run;
```

The first XRCHART statement reads the control limits from the first observation in the data set Diamlim for which the variable _VAR_ is equal to 'Diameter' and the variable _SUBGRP_ is equal to 'Hour'. The second XRCHART statement computes the control limits from the measurements in the data set Pistons. Note that the second XRCHART statement is equivalent to the following statements, which are more commonly used:

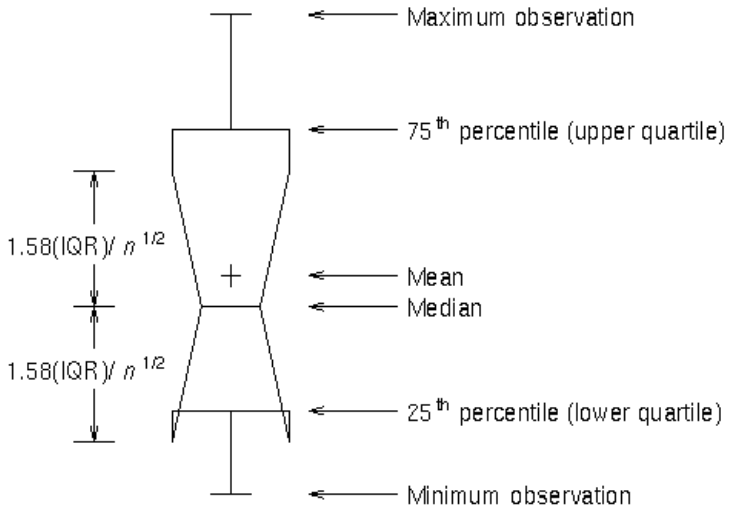
```
proc shewhart data=Pistons;
   xrchart Diameter*Hour;
run;
```

For more information about reading control limits from a LIMITS= data set, see the entry for the READLIMITS option and "Displaying Multiple Sets of Control Limits" on page 2081.

NOTCHES

specifies that box-and-whisker plots created by the BOXCHART statement be notched. The endpoints of the notches are located at the median plus and minus $1.58(\mathrm{IQR}/\sqrt{n})$, where IQR is the interquartile range and n is the subgroup sample size. The medians (central lines) of two box-and-whisker plots are significantly different at approximately the 0.05 level if the corresponding notches do not overlap. Refer to McGill, Tukey, and Larsen (1978). Figure 19.135 illustrates the NOTCHES option. Notice the folding effect at the bottom, which happens when the endpoint of a notch is beyond its corresponding quartile. This situation occurs typically only when the subgroup sample size is small.

Figure 19.135 NOTCHES Option for Box-and-Whisker Plots



The NOTCHES option is also illustrated in Output 19.3.1 and is available only in the BOXCHART statement.

NOTESTACROSS

specifies that tests for special causes requested with the TESTS= or TESTS2= options not be applied across the boundaries of phases (blocks of consecutive subgroups) determined by the READPHASES= option and the variable _PHASE_ in the input data set. With constant control limits, if you specify the READPHASES= option but do not specify the NOTESTACROSS option, tests for special causes are applied without regard to phase boundaries. With varying control limits, tests are applied only within phases by default, and you can use the TESTACROSS option to specify that they be applied across phase boundaries. See "Tests for Special Causes: SHEWHART Procedure" on page 2119.

NOTICKREP

applies to character-valued *subgroup-variables* and specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on the horizontal axis.

NOTRENDCONNECT

suppresses line segments that connect points on a trend chart. Points are connected by default. The NOTRENDCONNECT option is available only in the BOXCHART, MCHART, and XCHART statements when the TRENDVAR= option is used.

NOTRUNC

overrides the vertical axis truncation at zero, which is applied by default to *c* charts, moving range charts, *np* charts, *p* charts, *R* charts, *s* charts, and *u* charts. This option is useful if you are creating a customized version of one of these charts and want to replace the plotted statistics and control limits with values read from a TABLE= input data set that can be positive or negative. Do not use the NOTRUNC option in standard control chart applications. This option is not available in the BOXCHART, MCHART, and XCHART statements.

NOUCL
suppresses the display of the upper control limit in a primary chart.

NOUCL2
suppresses the display of the upper control limit in a secondary chart. The NOUCL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NPANELPOS=*n*
NPANEL=*n*
specifies the number of subgroup positions per panel on each chart. A *panel* is defined as a screen or page (or a half-screen or half-page if you are also using the BILEVEL option). You typically specify the NPANELPOS= option to display more points on a panel than the default number, which is $n = 50$ for all chart statements except the BOXCHART statement, for which the default is $n = 20$.

You can specify a positive or negative number for *n*. The absolute value of *n* must be at least 5. If *n* is positive, the number of positions is adjusted so that it is approximately equal to *n* and so that all panels display approximately the same number of subgroup positions. If *n* is negative, no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions. In this case, the approximation is due only to axis scaling.

You can use the INTERVAL= option to change the effect of the NPANELPOS= option when a date or time format is associated with the *subgroup-variable*. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the subgroup variable without having to associate a different format with the subgroup variable.

NPSYMBOL='*label***'**
NPSYMBOL=NP | NPBAR | NPPM | NPPM2 | NP0
specifies a label for the central line in an *np* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters in length.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is one of the symbols given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
NP	NP	NP
NPBAR	$\overline{\text{NP}}$	$\overline{\text{NP}}$
NPPM	NP′	NP′
NPPM2	NP″	NP″
NP0	NP_0	NP0

The default keyword is NPBAR. The NPSYMBOL= option is available only in the NPCHART statement.

OUTBOX=*SAS-data-set*
creates an output data set that contains subgroup summary statistics, control limits, and outlier values for a box chart. An OUTBOX= data set is the only type of summary data set produced by the SHEWHART procedure from which you can reconstruct a schematic box chart. The OUTBOX= option is available only in the BOXCHART statement. See "OUTBOX= Data Set" on page 1450 for details.

OUTFILL

COUTFILL

fills the areas outside the control limits that lie between the connected points and the control limits and are bounded by connecting lines. The areas are filled with an appropriate contrasting color from the ODS style. This option is useful for highlighting out-of-control points.

OUTHISTORY=*SAS-data-set*

creates an output data set that contains the subgroup summary statistics. You can use an OUTHISTORY= data set as a HISTORY= input data set in a subsequent run of the procedure. You cannot request an OUTHISTORY= data set if the input data set is a TABLE= data set. See "Output Data Sets" in the section for the chart statement in which you are interested. A related option is OUTPHASE=.

OUTINDEX='*label***'**

specifies the value of the _INDEX_ variable in the OUTLIMITS= output data set. This is a bookkeeping variable that provides information identifying the control limits saved in the data set. See "Output Data Sets" in the section for the chart statement in which you are interested.

The *label* can be up to 128 characters and should be enclosed in quotes. You should use a label that uniquely identifies the control limits. For example, you might specify OUTINDEX='April 1-15' to indicate that the limits were computed from data collected during the first half of April.

The OUTINDEX= option is intended to be used in conjunction with the OUTLIMITS= option. The _INDEX_ variable is created only if you specify the OUTINDEX= option. If you specify the OUTINDEX= option and do not specify the name of the OUTLIMITS= data set with the OUTLIMITS= option, the procedure creates an OUTLIMITS= data set whose name is of the form WORK.DATA*n*.

NOTE: You cannot use the OUTINDEX= and READINDEXES= options in the same chart statement.

OUTLABEL=VALUE

OUTLABEL=(*variable***)**

labels each point that falls outside the control limits on the primary chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, it can be up to 16 characters. For each subgroup of observations whose summary statistic falls outside the control limits, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. By default, points are not labeled. The OUTLABEL= option takes precedence over the TESTLABEL= option when TESTS=1 is specified. You cannot specify both the OUTLABEL= and ALLLABEL= options.

OUTLABEL2=VALUE

OUTLABEL2=(*variable***)**

labels each point that falls outside the control limits on an *R* or *s* chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations whose summary statistic falls outside the control limits, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. By default,

points are not labeled. The OUTLABEL2= option takes precedence over the TESTLABEL2= option when TESTS2=1 is specified. You cannot specify both the OUTLABEL2= and ALLLABEL2= options. The OUTLABEL2= option is available only in the IRCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

OUTLIMITS=*SAS-data-set*
creates an output data set that saves the control limits. You can use an OUTLIMITS= data set as an input LIMITS= data set in a subsequent run of the procedure. See "Output Data Sets" in the section for the chart statement in which you are interested. A related option is OUTINDEX=.

OUTPHASE='*label*'
specifies the value of the _PHASE_ variable in the OUTHISTORY= data set. This is a bookkeeping variable that provides information identifying the summary statistics saved in the data set. See "Output Data Sets" in the section for the chart statement in which you are interested.

You should use the OUTPHASE= option if you create OUTHISTORY= data sets at different stages (phases) for the same *processes* and concatenate the data sets to build a master historical data set. The _PHASE_ variable then identifies the block of observations that corresponds to each phase.

The *label* can be up to 128 characters and should be enclosed in quotes. You should use a *label* that uniquely identifies the saved data. For example, you might specify OUTPHASE='April 1-15' to indicate that the data were collected during the first half of April.

The _PHASE_ variable is created only if you specify the OUTPHASE= option. If you specify the OUTPHASE= option and do not specify the name of the OUTHISTORY= data set with the OUTHISTORY= option, the procedure creates an OUTHISTORY= data set whose name is of the form WORK.DATA*n*.

OUTTABLE=*SAS-data-set*
creates an output SAS data set that saves the information plotted on the chart, including the subgroup variable values and their corresponding summary statistics and control limits.

You can use the OUTTABLE= data set to create a customized report with the reporting procedures and methods described in *Base SAS Procedures Guide*. You can also use an OUTTABLE= data set as a TABLE= input data set in a subsequent run of the procedure. See "Output Data Sets" in the section for the chart statement in which you are interested.

OVERLAY=(*variable-list***)**
specifies variables to be overlaid on the primary control chart. A point is plotted for each overlay variable at each subgroup for which it has a nonmissing value. The value of a particular overlay variable should be the same for each observation in the input data set with a given value of the subgroup variable. If values differ within a subgroup, the first value appearing in that subgroup is used. The OVERLAY= option cannot be specified with the STARVERTICES= option.

OVERLAY2=(*variable-list***)**
specifies variables to be overlaid on a secondary control chart. A point is plotted for each overlay variable at each subgroup for which it has a nonmissing value. The value of a particular overlay variable should be the same for each observation in the input data set with a given value of the subgroup variable. If values differ within a subgroup, the first value appearing in that subgroup is used. The OVERLAY2= option cannot be specified with the STARVERTICES= option.

OVERLAY2ID=(*variable-list***)**
specifies variables whose formatted values are used to label points on secondary chart overlays. Variables in the OVERLAY2ID= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2ID= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYID=(*variable-list***)**
specifies variables whose formatted values are used to label points on primary chart overlays. Variables in the OVERLAYID= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYID= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYLEGLAB='*label*'
specifies the label displayed to the left of the legend for overlays requested with the OVERLAY= or OVERLAY2= option. The label can be up to 16 characters and must be enclosed in quotes.

P0=*value*
specifies a known (standard) value p_0 for the proportion of nonconforming items produced by the process. By default, p_0 is estimated from the data. The P0= option is available only in the NPCHART and PCHART statements.

NOTE: As an alternative to specifying P0=p_0, you can read a predetermined value for p_0 from the variable _P_ in a LIMITS= data set. See "Input Data Sets" in the section for the chart statement in which you are interested.

PAGENUM='*string*'
specifies the form of the label used for pagination.

The *string* must be no longer than 16 characters, and it must include one or two occurrences of the substitution character #. The first # is replaced with the page number, and the optional second # is replaced with the total number of pages.

The PAGENUM= option is useful when you are working with a large number of subgroups, resulting in multiple pages of output. For example, suppose that each of the following XRCHART statements produces multiple pages:

```
proc shewhart data=Pistons;
   xrchart Diameter*Hour / pagenum='Page #';
   xrchart Diameter*Hour / pagenum='Page # of #';
   xrchart Diameter*Hour / pagenum='#/#';
run;
```

The third page produced by the first statement would be labeled *Page 3*. The third page produced by the second statement would be labeled *Page 3 of 5*. The third page produced by the third statement would be labeled *3/5*.

By default, no page number is displayed.

PAGENUMPOS=TL | TR | BL | BR | TL100 | TR100 | BL0 | BR0
specifies where to position the page number requested with the PAGENUM= option. The keywords TL, TR, BL, and BR correspond to the positions top left, top right, bottom left, and bottom right,

respectively. You can use the TL100 and TR100 keywords to ensure that the page number appears at the very top of a page when a title is displayed. The BL0 and BR0 keywords ensure that the page number appears at the very bottom of a page when footnotes are displayed. The default keyword is BR.

PCTLDEF=*index*

specifies one of five definitions used to calculate percentiles in the construction of box-and-whisker plots requested with the BOXCHART statement. The *index* can be 1, 2, 3, 4, or 5. The five corresponding percentile definitions are discussed in "Percentile Definitions" on page 1460. The default is 5. The PCTLDEF= option is available only in the BOXCHART statement.

PHASEBREAK

specifies that the last point in a phase (defined as a block of consecutive subgroups with the same value of the _PHASE_ variable) is not to be connected to the first point in the next phase. By default, the points are connected.

PHASELABTYPE=SCALED | TRUNCATED

PHASELABTYPE=*height*

specifies how lengthy _PHASE_ variable values are displayed when there is insufficient space in the legend requested with the PHASELEGEND option. By default, lengthy values are not displayed.

If you specify PHASELABTYPE=SCALED, the values are uniformly reduced in height so that they fit. If you specify PHASELABTYPE=TRUNCATED, lengthy values are truncated on the right until they fit. When producing traditional graphics, you can also specify a text *height* in vertical percent screen units for the values. Related options are PHASELEGEND and PHASEREF.

NOTE: In ODS Graphics output only PHASELABTYPE=TRUNCATED is supported.

PHASELEGEND

PHASELEG

identifies the phases requested with the READPHASES= option in a legend across the top of the chart. Related options are PHASELABTYPE= and PHASEREF.

PHASELIMITS

specifies that the control limits and center line be labeled for each phase specified with the READPHASES= option, providing the limits are constant within that phase.

PHASEMEANSYMBOL=*symbol*

PHASEMEAN

specifies a symbol marker for the average of the values plotted within a phase. Specify PHASEMEAN without an argument to plot the phase average in ODS Graphics output. This option is available only in the BOXCHART statement.

PHASEREF

delineates the phases specified with the READPHASES= option with reference lines drawn vertically. Related options are PHASELABTYPE= and PHASELEGEND.

PHASEVARLABEL

displays the label associated with the variable _PHASE_ above the phase values in the phase legend. If there is no label associated with _PHASE_, or if the PHASELEGEND option is not specified, PHASEVARLABEL has no effect.

PHASEVALSEP

displays vertical lines separating phase values in the phase legend. If the PHASELEGEND option is not specified, PHASEVALSEP has no effect.

PROBLIMITS=DISCRETE

requests that discrete-valued probability limits be computed for attribute charts. This option is available only in the CCHART, NPCHART, PCHART, and UCHART statements, and it applies only when you request probability limits by specifying the ALPHA= option.

The possible values for the discrete probability limits are the same as for the subgroup values that are plotted on the control chart. For *c* and *np* charts these are integer values; for *p* and *u* charts these are multiples of $1/n$, where n is the subgroup sample size. Because attribute chart data are discrete, it is not possible in general to compute probability limits so that the probability of a point being outside the limits is α, for any arbitrary α.

The *c* and *u* charts are based on the Poisson distribution, which has the probability function

$$g(x) = \frac{\mu^x e^{-\mu}}{x!}, x = 0, 1, 2, \ldots$$

and the cumulative distribution function

$$G(x) = \sum_{i=0}^{x} g(i) = e^{-\mu} \sum_{i=0}^{x} \frac{\mu^i}{i!}, x = 0, 1, 2, \ldots$$

The *np* and *p* charts are based on the binomial distribution, which has the probability function

$$g(x) = \binom{n}{x} p^x (1-p)^{n-x}, x = 0, 1, 2, \ldots$$

and the cumulative distribution function

$$G(x) = \sum_{i=0}^{x} g(i) = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}, x = 0, 1, 2, \ldots$$

For *c* and *np* charts, the discrete lower control limit x_L is the smallest integer such that

$$G(x_L) \geq 1 - \alpha/2$$

and the discrete upper control limit x_U is the smallest integer such that

$$G(x_U) > \alpha/2$$

For *p* and *u* charts, the discrete lower control limit x_L is the smallest multiple of $1/n$ such that

$$G(nx_L) \geq 1 - \alpha/2$$

and the discrete upper control limit x_U is the smallest multiple of $1/n$ such that

$$G(nx_U) > \alpha/2$$

You can specify the ACTUALALPHA option to display the actual probability (instead of the probability you specify in the ALPHA= option) of a point being outside an attribute chart's probability limits.

PSYMBOL='*label***'**

PSYMBOL=P | PBAR | PPM | PPM2 | P0

specifies a label for the central line in a *p* chart. You can use the option in the following ways:

- Specify a quoted *label* up to 16 characters.
- Specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
P	P	P
PBAR	$\overline{P}$	$\overline{P}$
PPM	P'	P'
PPM2	P''	P''
P0	P_0	P0

The default keyword is PBAR. The PSYMBOL= option is available only in the PCHART statement.

RANGES

estimates the process standard deviation for a boxplot using subgroup ranges. By default, the process standard deviation for a boxplot is estimated from the subgroup standard deviations.

READALPHA

specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, be read from a LIMITS= data set when both variables are available in the data set. Thus, the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.

READINDEX=*value-list* **| ALL**

READINDEXES=*value-list* **| ALL**

READINDICES=*value-list* **| ALL**

reads one or more sets of control limits from a LIMITS= data set (specified in the PROC SHEWHART statement) for each *process* listed in the chart statement. The *i*th set of control limits for a particular *process* is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*
- the value of _INDEX_ matches *value*

The *values* can be up to 128 characters and must be enclosed in quotes.

NOTE: You cannot use the READINDEX= and OUTINDEX= options in the same chart statement. Also, the READLIMITS and READINDEX= options are alternatives to each other. If the LIMITS= data set contains more than one set of control limits for the same *process*, you should use the READINDEX= option.

You can display distinct sets of control limits (read from a LIMITS= data set) with data for various *phases* (read from blocks of observations in the input data set) by using the READINDEXES= and READPHASES= options together. See the entry for the READPHASES= option.

For more information about multiple sets of control limits and about the keyword ALL, see "Displaying Multiple Sets of Control Limits" on page 2081.

READLIMITS

specifies that the control limits are read from a LIMITS= data set specified in the PROC SHEWHART statement.[10] The control limits for each *process* listed in the chart statement are to be read from the first observation in the LIMITS= data set where

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on the release of SAS/QC software that you are using.

- **In SAS 6.10 and later releases, the READLIMITS option is not necessary.** To read control limits as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases. Consequently, the following two XRCHART statements are equivalent:

```
proc shewhart data=Pistons limits=Diamlim;
   xrchart Diameter*Hour;
   xrchart Diameter*Hour / readlimits;
run;
```

  If the LIMITS= data set contains more than one set of control limits for the same *process*, you should use the READINDEX= option.

- **In SAS 6.09 and earlier releases, you must specify the READLIMITS option to read control limits as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the control limits are computed from the data. Consequently, the following two XRCHART statements are **not** equivalent:

```
proc shewhart data=Pistons limits=diamlim;
   xrchart Diameter*Hour;  /* limits computed from data */
   xrchart Diameter*Hour /
      readlimits;          /* limits read from DIAMLIM  */
run;
```

  The READLIMITS and READINDEX= options are alternatives to each other.

  You can use the READLIMITS and READPHASES= options together. In this case, the control limits are read as described previously, and the data plotted on the chart are those selected by the READPHASES= option.

READPHASES=*value-list* | **ALL**

READPHASE=*value-list* | **ALL**

selects blocks of consecutive observations to be read from the input data set. You can use the READPHASES= option only if

- the input data set contains a _PHASE_ variable
- the _PHASE_ variable is a character variable of no more than 128 characters

[10] For details about computing control limits from the data, see the entry for the NOREADLIMITS option.

The READPHASES= option selects those observations whose _PHASE_ value matches one of the *values* specified in the *value-list*. The block of consecutive observations identified by the *i*th *value* is referred to as the *i*th *phase*. The *values* can be up to 128 characters and must be enclosed in quotes. List the *values* in the same order that they appear as values of the variable _PHASE_ in the input data set.

With the READPHASES= option you can

- create control charts that label blocks of data corresponding to multiple time *phases*. See the PHASELEGEND, PHASEREF, and CFRAME= options.
- create *historical control charts* that display distinct sets of control limits for different *phases*. This also requires a LIMITS= data set and the READINDEXES= option.

If the subgroup variable is numeric, the values of the subgroup variable should be contiguous from one block of observations to the next. Otherwise, there might be a gap in the control chart between the last point in one phase and the first point in the next phase. If you read a data set that contains multiple observations for each subgroup, the value of _PHASE_ must be constant within the subgroup.

You can display distinct sets of control limits (read from a LIMITS= data set) with data for various *phases* by using the READINDEX= and READPHASES= options together. For example, consider the flange width data in the HISTORY= data set Flange and the LIMITS= data set Flangelim. A partial listing of Flange is given in Figure 19.136 (for a complete listing of Flange, see Figure 19.149). The complete listing of Flangelim is given in Figure 19.137.

```
proc print data=Flange;
   var _phase_ Day Sample FlangewidthX FlangewidthR FlangewidthN;
run;
```

Figure 19.136 Listing of the HISTORY= Data Set Flange

Mean Chart for Diameters

Obs	_phase_	Day	Sample	FlangewidthX	FlangewidthR	FlangewidthN
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
4	Production	11FEB90	9	0.95509	0.08378	5
5	Production	12FEB90	10	1.00348	0.09993	5
6	Production	15FEB90	11	1.02566	0.06766	5
7	Production	16FEB90	12	0.97053	0.07608	5
8	Production	17FEB90	13	0.94713	0.10170	5
9	Production	18FEB90	14	1.00377	0.04875	5
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
13	Change 1	24FEB90	18	1.02235	0.10541	5
14	Change 1	25FEB90	19	0.99950	0.11476	5
15	Change 1	26FEB90	20	0.99271	0.05395	5
16	Change 1	01MAR90	21	0.98695	0.03833	5
17	Change 1	02MAR90	22	1.00969	0.06183	5
18	Change 1	03MAR90	23	0.98791	0.05836	5
19	Change 1	04MAR90	24	1.00170	0.05243	5
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
23	Change 2	10MAR90	28	1.01463	0.05558	5
24	Change 2	11MAR90	29	0.99812	0.03648	5
25	Change 2	12MAR90	30	1.00047	0.04309	5
26	Change 2	15MAR90	31	0.99714	0.03689	5
27	Change 2	16MAR90	32	0.98642	0.04809	5
28	Change 2	17MAR90	33	0.98891	0.07777	5
29	Change 2	18MAR90	34	1.00087	0.06409	5
30	Change 2	19MAR90	35	1.00863	0.02649	5

```
proc print data=Flangelim;
   var _index_ _var_ _subgrp_ _type_ _limitn_ _alpha_ _sigmas_
       _lclx_ _mean_ _uclx_ _lclr_ _r_ _uclr_ _stddev_;
run;
```

Figure 19.137 Listing of the LIMITS= Data Set Flangelim

Mean Chart for Diameters

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_	_lclx_	_mean_
1	Change 1	Flangewidth	Sample	ESTIMATE	5	.0026998	3	0.96167	0.99924
2	Production	Flangewidth	Sample	ESTIMATE	5	.0026998	3	0.93792	0.98827
3	Start	Flangewidth	Sample	ESTIMATE	5	.0026998	3	0.87088	0.96803

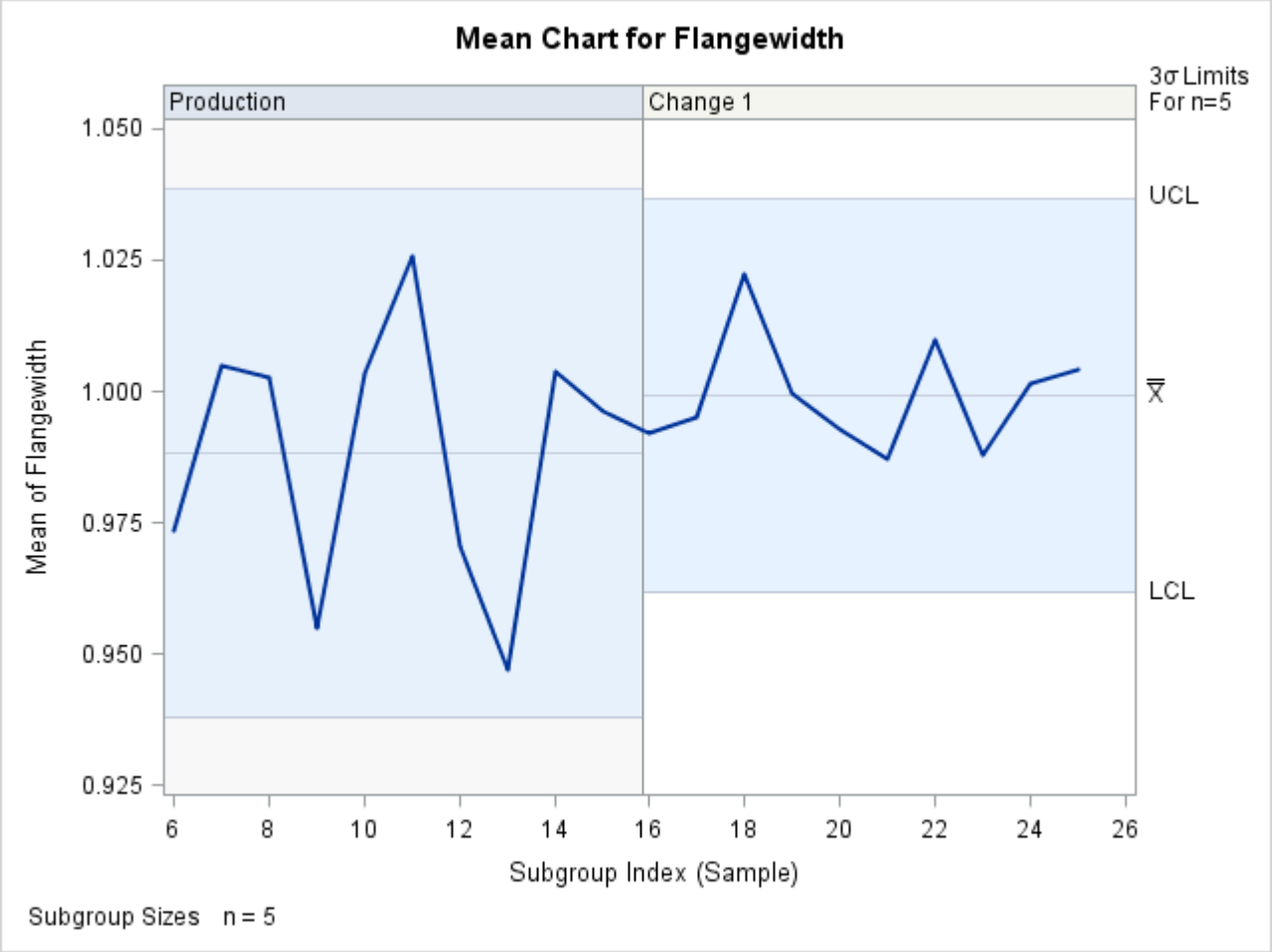
Obs	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	1.03680	0	0.06513	0.13771	0.028000
2	1.03862	0	0.08729	0.18458	0.037530
3	1.06517	0	0.16842	0.35612	0.072409

The following statements use the READINDEX= and READPHASES= options to create a historical control chart for the *Production* and *Change 1* phases:

```
ods graphics on;
proc shewhart history=Flange limits=Flangelim;
   xchart Flangewidth*Sample /
      readphases  = ('Production' 'Change 1')
      readindexes = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

The chart is displayed in Figure 19.138.

Figure 19.138 Multiple Control Limits for Multiple Phases



You can also use the keyword ALL with the READPHASES= option to match control limits to phases. For more information and examples about specifying multiple control limits, including the use of the keyword ALL, see "Displaying Multiple Sets of Control Limits" on page 2081.

REPEAT

REP

specifies that the horizontal axis of a chart that spans multiple pages be arranged so that the last subgroup position on a page is repeated as the first subgroup position on the next page. The REPEAT option facilitates cutting and pasting panels together. If a SAS DATETIME format is associated with the subgroup variable, REPEAT is used by default.

RSYMBOL='*label***'**

RSYMBOL=R | RBAR | RPM | R0

specifies a label for the central line in an *R* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
R	R	R
RBAR	$\overline{R}$	$\overline{R}$
RPM	R'	R'
R0	R_0	R0

The default keyword is RBAR. The RSYMBOL= option is available only in the IRCHART, MRCHART, RCHART, and XRCHART statements.

RTMARGIN=*value*

RTM=*value*

specifies the width of the right marginal area for the plot requested with the RTMPLOT= option. For traditional graphics, the width is specified in horizontal percent screen units. For ODS Graphics output, the width is specified in pixels. The RTMARGIN= option is available only in the IRCHART statement.

RTMPLOT=*keyword*

requests a univariate plot of the control chart statistics that is positioned in the right margin of the control chart. The *keywords* that you can specify and the associated plots are listed in the following table:

Keyword	Marginal Plot
DIGIDOT	Digidot plot
HISTOGRAM	Histogram
SKELETAL	Skeletal box-and-whisker plot
SCHEMATIC	Schematic box-and-whisker plot
SCHEMATICID	Schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	Schematic box-and-whisker plot with far outliers labeled

NOTE: Digidot plots are not available in ODS Graphics output.

The RTMPLOT= option is available only in the IRCHART statement; see Example 19.13 for an example. Refer to Hunter (1988) for a description of digidot plots, and see the entry for the BOXSTYLE= option for a description of the various box-and-whisker plots. Related options are LTMARGIN=, LTMPLOT=, and RTMARGIN=.

SEPARATE

displays primary and secondary charts on separate screens or pages. This option is useful if you are displaying line printer charts on a terminal and the number of lines on the screen limits the resolution of the chart. The SEPARATE option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

SERIFS

adds serifs to the whiskers of *skeletal box-and-whisker charts*. The SERIFS option is available only in the BOXCHART statement.

SIGMA0=*value*

specifies a known (standard) value σ_0 for the process standard deviation σ. By default, σ_0 is estimated from the data.

The SIGMA0= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

NOTE: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.

SIGMAS=*k*

specifies the width of the control limits in terms of the multiple k of the standard error of the subgroup summary statistic plotted on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are "3σ limits."

The particular subgroup summary statistic whose standard error is multiplied by k depends on the chart statement, as indicated by the following table:

Statement	Subgroup Summary Statistic
BOXCHART	Mean or median
CCHART	Number nonconforming
IRCHART	Individual measurements and moving ranges
MCHART	Median
MRCHART	Median and range
NPCHART	Number nonconforming
PCHART	Proportion nonconforming
RCHART	Range
SCHART	Standard deviation
UCHART	Number of nonconformities per unit
XCHART	Mean
XRCHART	Mean and range
XSCHART	Mean and standard deviation

For details, see the Options for Specifying Control Limits table and the "Details" subsection in the section for the particular chart statement that you are using.

Note that

- as an alternative to specifying SIGMAS=k, you can read k from the variable _SIGMAS_ in a LIMITS= data set. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.
- as an alternative to specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set), you can request probability limits by specifying ALPHA=α (or reading the variable _ALPHA_ from a LIMITS= data set by specifying the READALPHA option).

SKIPHLABELS=*n*

SKIPHLABEL=*n*

specifies the number n of consecutive tick mark labels, beginning with the second tick mark label, that are thinned (not displayed) on the horizontal (subgroup) axis. For example, specifying SKIPHLABEL=1 causes every other label to be skipped (not displayed). Specifying SKIPHLABEL=2 causes the second and third labels to be skipped, the fifth and sixth labels to be skipped, and so forth.

The default value of the SKIPHLABELS= option is the smallest value *n* for which tick mark labels do not collide. A specified *n* will be overridden to avoid collision, unless you specify SKIPHLABELS=0, which forces all tick mark labels to be displayed. To avoid both collisions and thinning, you can use the TURNHLABELS option.

SMETHOD=NOWEIGHT | MVLUE | RMSDF | MAD | MMR | MVGRANGE
specifies a method for estimating the process standard deviation, σ, as summarized by the following table:

Keyword	Method for Estimating Standard Deviation
NOWEIGHT	Estimates σ as an unweighted average of unbiased subgroup estimates of σ
MVLUE	Calculates a minimum variance linear unbiased estimate for σ
RMSDF	Calculates a root-mean square estimate for σ
MAD	Calculates a median absolute deviation estimate for σ (IRCHART only)
MMR	Calculates a median moving range estimate for σ (IRCHART only)
MVGRANGE	Estimates σ based on a moving range of subgroup means (XRCHART and XSCHART only)

For formulas, see "Methods for Estimating the Process Standard Deviation" in the section for the particular chart statement you are using.

The default keyword is NOWEIGHT. The SMETHOD= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements. You can specify SMETHOD=RMSDF only in the BOXCHART, MCHART, XCHART, SCHART, and XSCHART statements and only when used with the STDDEVIATIONS option (or only in the absence of the RANGES option with a BOXCHART statement). You can specify SMETHOD=MAD and SMETHOD=MMR only in the IRCHART statement. You can specify SMETHOD=MVGRANGE only in the XRCHART and XSCHART statements.

SPLIT='*character***'**
specifies a special *character* that is inserted into the label of a process variable or summary statistic variable and whose purpose is to split the label into two parts. The first part is used to label the vertical axis of the primary chart, and the second part is used to label the vertical axis of the secondary chart. The *character* is not displayed in either label. See Figure 19.173 for an example.

The SPLIT= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

SSYMBOL='*label***'**

SSYMBOL=S | SBAR | SPM | S0
specifies a label for the central line in an *s* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
S	S	S
SBAR	$\overline{S}$	$\overline{S}$
SPM	S'	S'
S0	S_0	S0

The default keyword is SBAR. The SSYMBOL= option is available only in the SCHART and XSCHART statements.

STARBDRADIUS=*value*

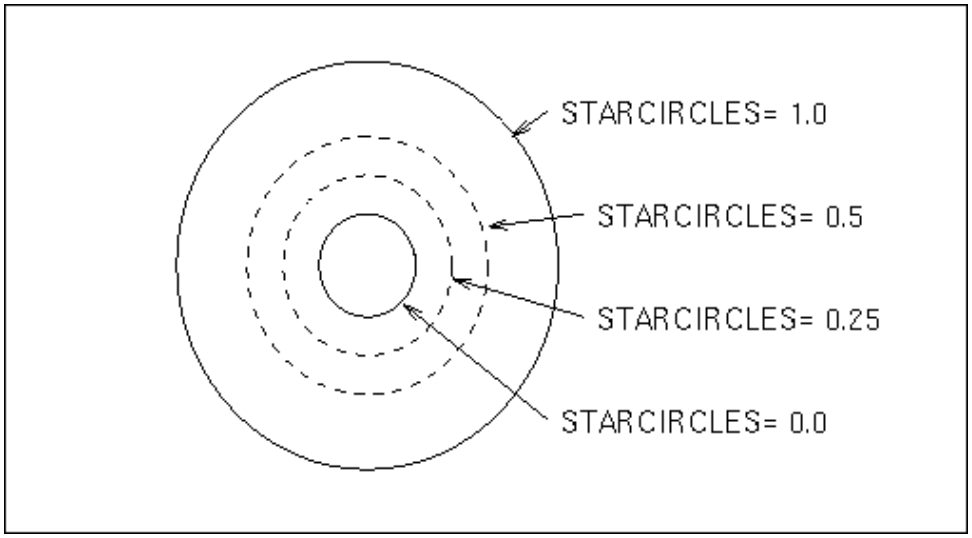
specifies the radius of an imaginary circle that is the outer bound for vertices of stars requested with the STARVERTICES= option. For traditional graphics, the radius is specified in horizontal percent screen units. For ODS Graphics output, the radius is specified in pixels. Vertices that exceed the outer bound are truncated to this value in order to prevent gross distortion of stars due to extreme values in the data. The *value* must be greater than or equal to the value specified with the STAROUTRADIUS= option. See Figure 19.140 or "Displaying Auxiliary Data with Stars" on page 2090.

STARCIRCLES=*values*

specifies reference circles that are superimposed on the stars requested with the STARVERTICES= option. All of the circles are displayed and centered at each point plotted on the primary chart. The *value* determines the diameter of the circle as follows: a *value* of zero specifies a circle with the *inner radius*, and a value of one specifies a circle with the *outer radius*. In general, a value of *h* specifies a circle with a radius equal to $inradius + h \times (outradius - inradius)$.

Figure 19.139 shows four circles specified with the STARCIRCLES= option. The values 0.0 and 1.0 correspond to the *inner circle* and *outer circle* (see the entries for the STARINRADIUS= and STAROUTRADIUS= options). The value 0.5 specifies a circle with a radius of $inradius + 0.5 \times (outradius - inradius)$ or a circle halfway between the inner circle and the outer circle. Likewise, the value 0.25 specifies a circle one-fourth of the way from the inner circle to the outer circle. Note also that the line types for the circles are specified with the LSTARCIRCLES= option. For more information, see "Displaying Auxiliary Data with Stars" on page 2090.

Figure 19.139 Circles Specified by STARCIRCLES=0.0 1.0 0.25 0.5

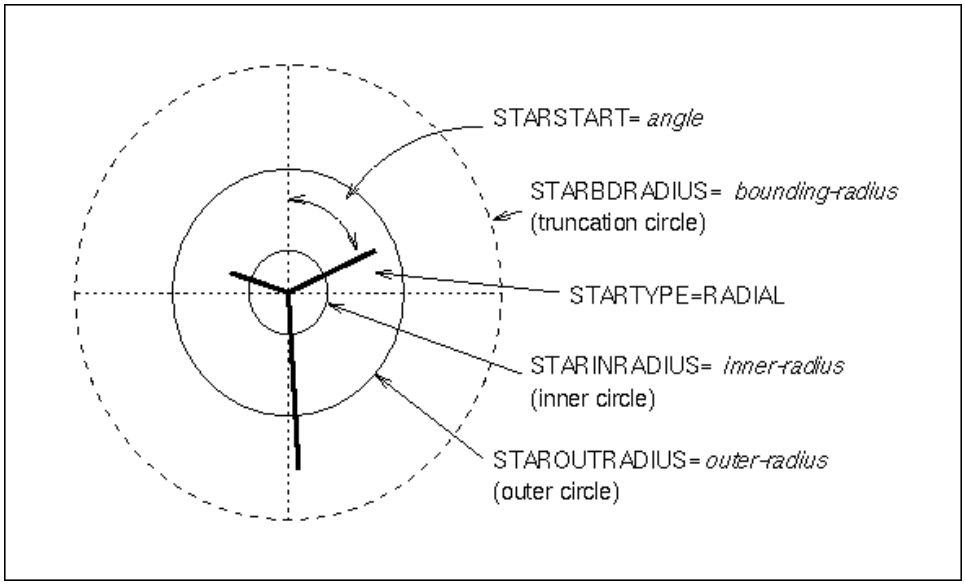


STARINRADIUS=*value*

specifies the inner radius of stars requested with the STARVERTICES= option. For traditional graphics, the radius is specified in horizontal percent screen units. For ODS Graphics output, the radius is specified in pixels. The *value* must be less than the value that is specified with the STAROUTRADIUS= option. The inner radius of a star is the distance from the center of the star to the circle that represents the lower limit of the standardized vertex variables. The lower limit can correspond to the minimum value, a multiple of standard deviations below the mean, or a lower specification limit. The default *value* is one-third of the outer radius.

Figure 19.140 illustrates five of the star options. The STARSTART= option determines the angle between the vertical axis and the first vertex. The STARINRADIUS= and STAROUTRADIUS= options specify the radii (in horizontal percent screen units) of the inner and outer circles that are associated with each star. Extremely large vertex values are truncated at the imaginary circle whose radius is specified by the STARBDRADIUS= option. The STARTYPE=RADIAL option specifies that the vertices are to be displayed as endpoints of line segments connecting each vertex to the center point. For more information, see the entries for these options or "Displaying Auxiliary Data with Stars" on page 2090.

Figure 19.140 Illustration of Star Options



STARFILL=*variable*

specifies colors for filling the interior of stars requested with the STARVERTICES= option. The STARFILL= option is analogous to the CSTARFILL= option, but the values of the STARFILL= variable are used only to group the stars. Stars in the same group are filled with the same color from the ODS style.

STARLABEL=ALL | FIRST | HIGH | LOW | OUT

specifies a method for labeling the vertices of stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword:

Keyword	Method for Labeling Star Vertices
ALL	Labels all vertices of all stars
FIRST	Labels all vertices of the leftmost star
HIGH	Labels only vertices that lie outside the outer circle
LOW	Labels only vertices that lie inside the inner circle
OUT	Labels only vertices that lie inside the inner circle or outside the outer circle

The label used for a particular vertex is the value of the variable _LABEL_ in the STARSPECS= data set. If this data set is not specified, or if the _LABEL_ variable is not provided, then the name of the vertex variable is used as the label. See "Displaying Auxiliary Data with Stars" on page 2090. By default, vertices are not labeled.

STARLEGEND=CLOCK | CLOCK0 | DEGREES | NONE

specifies the style of the legend used to identify the vertices of stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword:

Keyword	Star Vertices Legend Style
CLOCK	Identifies the vertex variables by their positions on the clock (starting with 12:00)
CLOCK0	Identifies the vertex variables by their positions on the clock (starting with 0:00)
DEGREES	Identifies the vertex variables by angles in degrees, with 0 degrees corresponding to 12 o'clock
NONE	Suppresses the legend

See "Displaying Auxiliary Data with Stars" on page 2090. The default keyword is CLOCK.

STARLEGENDLAB='*label***'**

specifies the label displayed to the left of the legend for stars requested with the STARLEGEND= option. The label can be up to 16 characters and must be enclosed in quotes. See "Displaying Auxiliary Data with Stars" on page 2090. The default label is *Vertices:*.

STAROUTRADIUS=*value*

specifies the outer radius of stars requested with the STARVERTICES= option. For traditional graphics, the radius is specified in horizontal percent screen units. For ODS Graphics output, the radius is specified in pixels. The outer radius of a star is the distance from the center of the star to the circle that represents the upper limit of the standardized vertex variables. The upper limit can correspond to the maximum value, a multiple of standard deviations above the mean, or an upper specification limit.

See Figure 19.140 "Displaying Auxiliary Data with Stars" on page 2090. For an example, see Figure 19.224. The default *value* depends on the number of subgroup positions per panel, and it is as large as possible without causing overlap of adjacent stars.

STARS=*variable*

specifies colors for the outlines of stars requested with the STARVERTICES= option. The STARS= option is analogous to the CSTARS= option, but the values of the STARS= variable are used only to group the stars. The outlines of stars in the same group are drawn with the same color from the ODS style.

STARSPECS=*value*|*SAS-data-set*

STARSPEC=*value*|*SAS-data-set*

specifies the method used to standardize the star vertex variables listed with the STARVERTICES= option. The method determines how the value of a vertex variable is transformed to determine the distance between the center of the star and the vertex. The STARSPECS= option also determines how the inner and outer radii of the star are to be interpreted.

A *value* of zero specifies standardization by the range of the variable. In this case, the distance between the center and the vertex is proportional to the difference between the variable value and the minimum variable value (taken across all subgroups). The inner radius of the star corresponds to the minimum variable value, and the outer radius of the star corresponds to the maximum variable value.

A positive STARSPECS= value requests standardization by a multiple of standard deviations above and below the mean. For example, STARSPECS=3 specifies that the inner radius of the star corresponds to three standard deviations below the mean, and the outer radius corresponds to three standard deviations above the mean. Thus, a vertex variable value exactly equal to the mean is represented by a vertex whose distance to the center of the star is halfway between the inner and outer radii.

You can request a distinct method of standardization for each vertex variable by specifying a STARSPECS= *data set*. Each observation provides standardization and related information for a distinct vertex variable. The variables read from a STARSPECS= *data set* are described in the following table:

Variable	Description
CSPOKE	Color of spokes used with STARTYPE=RADIAL and STARTYPE=SPOKE; this must be a character variable of length 8 or less
LABEL	Label for identifying the vertex when you specify STARLEGEND=FIRST or STARLEGEND=ALL; this must be a character variable of up to 16 characters
LSL	Lower specification limit
LSPOKE	Line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
NOMVAL	Nominal value substituted for missing values
SIGMAS	Multiple of standard deviations above and below the average
UBOUND	Upper bound for truncating extremely high values
USL	Upper specification limit
VAR	Name of vertex variable; this must be a character variable of length 32 or less

Only the variable _VAR_ is mandatory. If you provide the variables _LSL_ and _USL_, standardization is based on the specification limits; in this case, the variable _LSL_ corresponds to the inner radius of the star, and the variable _USL_ corresponds to the outer radius of the star. If you do not provide the variables _LSL_ and _USL_, standardization is based on the value of the variable _SIGMAS_, and if you do not provide the variable _SIGMAS_, standardization is based on the range.

See "Displaying Auxiliary Data with Stars" on page 2090. If you do not specify the STARSPECS= option, each vertex variable is standardized by its range across subgroups. In other words, the minimum corresponds to the inner radius, and the maximum corresponds to the outer radius.

STARSTART=*value*

specifies the vertex angle for the first variable in the STARVERTICES= list. Vertex angles for the remaining variables are uniformly spaced clockwise and assigned in the order listed. You can specify the *value* in the following ways:

- *Clock position:* If you specify the value as a time literal (between '0:00'T and '12:00'T), the corresponding clock position is used for the first vertex variable.
- *Degrees:* If you specify the value as a nonpositive number, the absolute value in degrees is used for the first vertex angle. Here, 0 degrees corresponds to 12:00.

The default *value* is zero, so the first vertex variable is positioned at 12:00. See Figure 19.140 or "Displaying Auxiliary Data with Stars" on page 2090.

STARTYPE=CORONA | POLYGON | RADIAL | SPOKE | WEDGE

specifies the style of the stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword.

Keyword	Star Style
CORONA	Polygon with star-vertices emanating from the inner circle
POLYGON	Closed polygon
RADIAL	Rays emanating from the center
SPOKE	Rays emanating from the inner circle
WEDGE	Closed polygon with rays from the center to each vertex

See Figure 19.140 or "Displaying Auxiliary Data with Stars" on page 2090. "Adding Reference Circles to Stars" on page 2093 describes the inner and outer circles, and "Specifying the Style of Stars" on page 2095 provides examples of each value of the STARTYPE= option. The default keyword is POLYGON.

STARVERTICES=*variable* | **(***variable-list***)**

superimposes a star (polygon) at each point on the primary chart. The star is centered at the point, and the distance between the center and each star vertex represents the standardized value of a *variable* in the STARVERTICES= list. The *variables* must be provided in the input data set.

The star display is suggested as a method for monitoring quantitative variables (such as environmental factors) that are measured simultaneously with the process variable. For examples and details, see "Displaying Auxiliary Data with Stars" on page 2090. By default, stars are not superimposed on the chart.

STDDEVIATIONS

STDDEVS

specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations. This, in turn, affects the calculation of control limits; for details, see "Methods for Estimating the Process Standard Deviation" in the section for the chart statement in which you are interested. By default, the estimate of σ is calculated from subgroup ranges, except with the BOXCHART statement, where subgroup standard deviations are used by default.

If you specify the STDDEVIATIONS option and read summary data from a HISTORY= data set, the data set must contain a subgroup standard deviation variable for each *process*. Conversely, if you omit

the STDDEVIATIONS option, the HISTORY= data set must contain a subgroup range variable for each *process* listed in the chart statement.

You should specify STDDEVIATIONS when your subgroup sample sizes are large (typically, 15 or greater). The STDDEVIATIONS option is available only in the MCHART and XCHART statements.

SUBGROUPN=*value*

SUBGROUPN=*variable*

specifies the subgroup sample sizes as a constant *value* or as the values of a *variable* in the DATA= data set. The SUBGROUPN= option is available only in the CCHART, NPCHART, PCHART, and UCHART statements.

You must specify SUBGROUPN= in the NPCHART, PCHART, and UCHART statements when your input data set is a DATA= data set. If you are using a CCHART statement, the SUBGROUPN= option is available only when your input data set is a DATA= data set. For the CCHART statement, the default value of the SUBGROUPN= option is one.

If you specify multiple *processes* in a chart statement, the SUBGROUPN= option is used with all of the *processes* listed.

SYMBOLLEGEND=LEGEND*n*

SYMBOLLEGEND=NONE

controls the legend for the levels of a *symbol-variable* (see "Displaying Stratification in Levels of a Classification Variable" on page 2073). For traditional graphics, you can specify SYMBOLLEGEND=LEGEND*n*, where *n* is the number of a LEGEND statement defined previously. You can specify SYMBOLLEGEND=NONE to suppress the default legend.

SYMBOLORDER=DATA | INTERNAL | FORMATTED

SYMORD=DATA | INTERNAL | FORMATTED

specifies the order in which symbols are assigned for levels of *symbol-variable*. The DATA keyword assigns symbols to values in the order in which values appear in the input data. This is how symbols were assigned in SAS 6.12 and earlier releases of SAS/QC software. The INTERNAL keyword assigns symbols based on sorted order of internal values of *symbol-variable* and FORMATTED assigns them based on sorted formatted values. The default value is FORMATTED.

TABLE <(EXCEPTIONS)>

TABLES <(EXCEPTIONS)>

creates a basic table of the subgroup values, the subgroup sample sizes, the subgroup summary statistics, and the upper and lower control limits. Rows of the table correspond to subgroups. The keyword **EXCEPTIONS** (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

You can request extended versions of the basic table by specifying one or more of the following options: TABLEBOX, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS. Specifying the TABLEALL option is equivalent to specifying all of these options, and it provides the most extensive table.

TABLEALL <(EXCEPTIONS)>

tabulates the information about the control chart and is equivalent to specifying all of the following options: TABLES, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS. If you specify the TABLEALL option in a BOXCHART statement, the TABLEBOX

option is also implied. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive. You can use the OUTTABLE= option to create a data set that saves the information tabulated with the TABLEALL option.

TABLEBOX < (EXCEPTIONS) >

augments the basic table created by the TABLES option with columns for the minimum, 25th percentile, median, 75th percentile, and maximum of the observations in a subgroup. The TABLEBOX option is available only in the BOXCHART statement. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLECENTRAL < (EXCEPTIONS) >

TABLEC < (EXCEPTIONS) >

augments the basic table created by the TABLES option with columns for the values of the central lines. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLEID < (EXCEPTIONS) >

augments the basic table created by the TABLES option with a column for each of the ID variables. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLELEGEND < (EXCEPTIONS) >

TABLELEG < (EXCEPTIONS) >

adds a legend to the basic table created by the TABLE option. The legend describes the tests for special causes that were requested with the TESTS= option and for which a positive signal is found for at least one subgroup. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLEOUTLIM < (EXCEPTIONS) >

TABLEOUT < (EXCEPTIONS) >

augments the basic table created by the TABLE option with columns indicating which control limits (if any) are exceeded. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLETESTS < (EXCEPTIONS) >

augments the basic table created by the TABLES option with a column that indicates which of the tests for special causes (requested with the TESTS= option) are positive. The column contains the numbers of all the tests that are positive at a particular subgroup. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TARGET=*value-list*

provides target values used to compute the capability index C_{pm}, which is saved in the OUTLIMITS= data set. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

CAUTION: You can use the TARGET= option only in conjunction with the LSL= and USL= options. For more information, see "Capability Indices" on page 1971 and "Output Data Sets" in the section for the chart statement in which you are interested. Also see the entries for the LSL= and USL= options. The TARGET= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

TEST2RESET=*variable*

TEST2RESET=*value*

enables tests for special causes to be reset in a secondary chart. The specified variable must be a character variable of length 8, or length 16 if customized tests are requested. The variable values have the same format as those of the _TESTS_ variable in a TABLE= data set. A test that is flagged by the TEST2RESET= value for a given subgroup is reset starting with that subgroup. That means a positive result for the test can include the given subgroup only if it is the first subgroup in the pattern. For example, the value "12345678" for the TEST2RESET= variable will reset all standard tests for special causes.

TEST2RUN=*run-length*

specifies the length of the pattern for Test 2 requested with the TESTS= and TESTS2= options. The values allowed for the *run-length* are 7, 8, 9, 11, 14, and 20. The form of the test for each *run-length* value is given in the following table. The default *run-length* is 9. See "Tests for Special Causes: SHEWHART Procedure" on page 2119 for more information.

Run-length	Number of Points on One Side of the Central Line
7	7 in a row
8	8 in a row
9	9 in a row
11	at least 10 out of 11 in a row
14	at least 12 out of 14 in a row
20	at least 16 out of 20 in a row

TEST3RUN=*run-length*

specifies the length of the pattern for Test 3 requested with the TESTS= and TESTS2= options. Test 3 searches for a pattern of steadily increasing or decreasing values, where the length of the pattern is at least the value given as the *run-length*. The values allowed for the *run-length* are 6, 7, and 8. The default *run-length* is 6. See "Tests for Special Causes: SHEWHART Procedure" on page 2119 for more information.

TESTACROSS

specifies that tests for special causes requested with the TESTS= or TESTS2= options be applied without regard to phases (blocks of consecutive subgroups) determined by the READPHASES= option and the variable _PHASE_ in the input data set. With varying control limits, if you specify the READPHASES= option but do not specify the TESTACROSS option, tests for special causes are applied within (but not across) phases. With constant control limits, tests are applied across phases by default, and you can use the NOTESTACROSS option to specify that they be applied only within phases. See "Tests for Special Causes: SHEWHART Procedure" on page 2119.

TESTLABBOX
requests that labels for subgroups with positive tests for special causes are positioned so they do not overlap. The labels are enclosed in boxes that are connected to the associated subgroup points with line segments.

TESTLABEL='*label***'**
TESTLABEL=(*variable***)**
TESTLABEL=TESTINDEX
TESTLABEL=SPACE
TESTLABEL=NONE
provides labels for points at which one of the tests for special causes (requested with the TESTS= or TESTS2= option) is positive. The values for the TESTLABEL= option are as follows:

- You can specify a *label* of up to 16 characters enclosed in quotes. This label is displayed at all points where a test is signaled.
- You can specify a *variable* (enclosed in parentheses) whose values are used as labels. The *variable* must be provided in the input data set, and it can be numeric or character. If the *variable* is character, its length cannot exceed 16. For each subgroup of observations at which a test is signaled, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup.
- You can specify TESTINDEX to label points with the single-digit *index* that requested the test in a TESTS= or TESTS2= list. If the test was requested with a customized *pattern* in a TESTS= or TESTS2= list, then points are labeled with the letter that you specified using the CODE= option.
- You can specify SPACE to request a label of the form *Test k*. This is slightly more legible than the default label of the form *Testk* (a description of *Testk* follows).
- You can specify NONE to suppress labeling.

If you do not use the TESTLABEL= option, the default label is of the form *Testk*, where *k* is the index of the test as requested with the TESTS= or TESTS2= options, or *k* is the CODE= *character* of the test as requested in a pattern specified with the TESTS= or TESTS2= options.

See "Tests for Special Causes: SHEWHART Procedure" on page 2119. Related options include OUTLABEL=, OUTLABEL2=, TESTFONT=, TESTHEIGHT=, and TESTLABEL*n*=.

TESTLABEL*n*='*label***'**
specifies a *label* for points at which the test for special causes requested with the *index n* in a TESTS= or TESTS2= list is positive. The *index n* can be a number from 1 to 8. The TESTLABEL*n*= option overrides a TESTLABEL= option and the default label *Test n*. The *label* that you specify with the TESTLABEL*n*= option can be up to 16 characters and must be enclosed in quotes.

See "Tests for Special Causes: SHEWHART Procedure" on page 2119. Related options are TESTFONT=, TESTHEIGHT=, and TESTLABEL=.

TESTNMETHOD=STANDARDIZE
applies the tests for special causes requested with the TESTS= and TESTS2= options to standardized test statistics when the subgroup sample sizes are not constant. This method was suggested by Nelson (1994). See "Tests for Special Causes: SHEWHART Procedure" on page 2119. By default, the tests are not applied to data with varying subgroup sample sizes.

TESTOVERLAP

applies tests for special causes (requested with the TESTS= or TESTS2= option) to overlapping patterns of points.

The TESTOVERLAP option modifies the way in which the search for a subsequent pattern is done when a pattern is encountered. If you omit the TESTOVERLAP option, the search begins with the first subgroup after the current pattern ends. If you specify the TESTOVERLAP option, the search begins with the second subgroup in the current pattern.

The following statements illustrate the use of the TESTOVERLAP option:

```
proc shewhart;
   xrchart Width*Hour / test=3;
   xrchart Width*Hour / test=3 testoverlap;
run;
```

Test 3 looks for six subgroup means in a row steadily increasing or decreasing. Suppose that the subgroup means of Width are steadily increasing for Hour=5, 6, 7, 8, 9, 10, and 11. The first XRCHART statement will signal that Test 3 is positive at Hour=10 but not at Hour=11, because the search for the next pattern begins with Hour=11. The second XRCHART statement will signal that Test 3 is positive at Hour=10 and Hour=11, because the search for the next pattern begins with Hour=6 and thus finds a second pattern ending with Hour=11. See "Tests for Special Causes: SHEWHART Procedure" on page 2119 for more information.

CAUTION: Specifying TESTOVERLAP affects the interpretation of the standard tests for special causes, because a particular point can contribute to more than one positive test. Typically, this option should not be used.

TESTRESET=*variable*

TESTRESET=*value*

enables tests for special causes to be reset in a primary chart. The specified variable must be a character variable of length 8, or length 16 if customized tests are requested. The variable values have the same format as those of the _TESTS_ variable in a TABLE= data set. A test that is flagged by the TESTRESET= value for a given subgroup is reset starting with that subgroup. That means that a positive result for the test can include the given subgroup only if it is the first subgroup in the pattern. For example, the value "12345678" for the TESTRESET= variable will reset all standard tests for special causes.

TESTS=*index-list*

TESTS=*customized-pattern-list*

requests one or more tests for special causes, which are also known as *runs tests*, *pattern tests*, and *Western Electric rules*. These tests detect particular nonrandom patterns in the points plotted on the primary control chart. The occurrence of a pattern, referred to as a *signal*, suggests the presence of a special cause of variation.

Each pattern is defined in terms of zones A, B, and C, which are constructed by dividing the interval between the control limits into six equally spaced subintervals. Zone A is the union of the subintervals immediately below the upper control limit and immediately above the lower control limit. Zone C is the union of the subintervals immediately above and below the central line. Zone B is the union of the subintervals between zones A and C. See Figure 19.178 for an illustration of test zones.

You can use the TESTS= option in three ways. First, you can specify an *index-list* to request a combination of standard tests (this is the approach most commonly used). Second, you can specify a *customized-pattern-list* to request a combination of tests based on customized patterns. Third, you can specify a list consisting of both *indexes* and *customized-patterns*. The first two approaches are described as follows.

Standard tests. The following table lists the standard tests that you can request by specifying TEST=*index-list*. The tests are indexed according to the sequence used by Nelson (1984, 1985).

Index	Pattern Description
1	One point beyond Zone A (outside the control limits)
2	Nine points in a row in Zone C or beyond on one side of the central line (see the entry for the TEST2RUN option)
3	Six points in a row steadily increasing (see the entry for the TEST3RUN option)
4	Fourteen points in a row alternating up and down
5	Two out of three points in a row in Zone A or beyond
6	Four out of five points in a row in Zone B or beyond
7	Fifteen points in a row in Zone C on either or both sides of the central line
8	Eight points in a row on either or both sides of the central line with no points in Zone C

You can specify any combination of the eight *indexes* with an explicit list or with an implicit list, as in the following example:

```
proc shewhart;
   xrchart Width*Hour / tests=1 2 3 4;
   xrchart Width*Hour / tests=1 to 4;
run;
```

The TESTS= option is available in all but the RCHART and SCHART statements. Use only tests 1, 2, 3, and 4 in the CCHART, NPCHART, PCHART, and UCHART statements. By default, the TESTS= option is not applied in any chart statement unless the control limits are 3σ limits. You can use the NO3SIGMACHECK option to request tests for special causes when you use the SIGMAS= option to specify control limits other than 3σ limits.

Customized tests. Although the standard tests that the TESTS= option supports are appropriate for the vast majority of control chart applications, there might be situations in which you want to use customized tests. You can define your own tests by specifying TESTS=*customized-pattern-list*. You can include three types of patterns in this list: *T-patterns*, *M-patterns*, and *S-patterns*.

Use a T-pattern to request a search for k out of m points in a row in the interval (a, b). The required syntax for a T-pattern is

T(K=*k* **M=***m* **LOWER=***a* **UPPER=***b* **SCHEME=***scheme* **CODE=***character* **LABEL=***'label'* **LEGEND=***'legend'***)**

The default value for SCHEME= is ONESIDED. The options for a T-pattern are summarized in the following table:

Option	Description
K=*k*	Number of points
M=*m*	Number of consecutive points
LOWER=*value*	Lower limit of interval (a, b)
UPPER=*value*	Upper limit of interval (a, b)
SCHEME=ONESIDED	One-sided scheme using (a, b)
SCHEME=TWOSIDED	Two-sided scheme using $(a, b) \cup (-b, -a)$
CODE=*character*	Identifier for test (A–H)
LABEL='*label*'	Label for points if signal
LEGEND='*legend*'	Legend used with the TABLELEGEND option

Use an M-pattern to request a search for k points in a row increasing or decreasing. The required syntax for an M-pattern is

M(K=*k* **DIR=***direction* **CODE=***character* **LABEL='***label***' LEGEND='***legend***')**

The options for an M-pattern are summarized in the following table:

Option	Description
K=*k*	Number of points
DIR=INC	Increasing pattern
DIR=DEC	Decreasing pattern
CODE=*character*	Identifier for test (A–H)
LABEL='*label*'	Label for points if signal
LEGEND='*legend*'	Legend used with the TABLELEGEND option

Use an S-pattern to request a search for a statistically significant linear trend over a window of k points. The required syntax for an S-pattern is

S(K=*k* **CLEV=***α* **FORM=***character* **CODE=***character* **LABEL='***label***' LEGEND='***legend***')**

The options for an S-pattern are summarized in the following table:

Option	Description
K=*k*	Number of points in sliding window ($k > 2$)
CLEV=*α*	Type I (false positive) error rate ($0 < \alpha \leq 0.5$)
FORM=*character*	Type of trend test (P=parametric, N=nonparametric)
CODE=*character*	Identifier for test (A–H)
LABEL='*label*'	Label for points if signal
LEGEND='*legend*'	Legend used with the TABLELEGEND option

For details on the TESTS= option, see "Tests for Special Causes: SHEWHART Procedure" on page 2119. Related options include CTEST=, CZONES=, LTEST=, TABLETESTS, TABLELEGEND, TEST2RUN=, TEST3RUN=, TESTACROSS, TESTCHAR=, TESTLABEL=, TESTLABELn=, TESTNMETHOD=, TESTOVERLAP, TESTS2=, ZONES, ZONECHAR=, and ZONELABELS.

TESTS2=*index-list*

TESTS2=*customized-pattern-list*

requests one or more tests for special causes for an *R* chart or *s* chart. The syntax for the TESTS2= option is identical to the syntax for the TESTS= option. The TESTS2= option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements. For details on the TESTS2=

option, see "Tests for Special Causes: SHEWHART Procedure" on page 2119. Related options include CTEST=, CZONES=, LTEST=, TABLETESTS, TABLELEGEND, TEST2RUN=, TEST3RUN=, TESTACROSS, TESTCHAR=, TESTLABEL=, TESTLABELn=, TESTNMETHOD=, TESTOVERLAP, TESTS2=, ZONES, ZONECHAR=, and ZONELABELS.

TOTPANELS=*n*

specifies the total number of panels to be used to display the chart. This option overrides the NPANEL= option.

TRENDVAR=*variable* | **(***variable-list***)**

specifies a list of trend variables, one for each *process* listed in the chart statement. The TRENDVAR= option is available only in the BOXCHART, MCHART, and XCHART statements and only when your input data set is a DATA= or HISTORY= data set.

The values of the trend variables are subtracted from the values of the corresponding process variables (if you read a DATA= data set) or subgroup mean variables (if you read a HISTORY= data set). The chart is then created for the residuals (differences), and the trend values are plotted in a secondary chart. If you specify a single trend variable and two or more *processes*, the trend variable is used with each *process*.

The TRENDVAR= option does not apply if you are reading a TABLE= data set. In this case, the procedure produces a trend chart only if the variable _TREND_ is included in the TABLE= data set.

For more details, see "Displaying Trends in Process Data" on page 2100. Related options include NOTRENDCONNECT, SEPARATE, SPLIT=, WTREND=, and YPCT1=.

TYPE=*value*

specifies the *value* of the _TYPE_ variable in the OUTLIMITS= data set, which in turn indicates whether certain parameter variables in this data set represent estimates or standard (known) values.

If you are using a chart statement that creates a variables chart, _TYPE_ is a bookkeeping variable that indicates whether the values of the variables _MEAN_ and _STDDEV_ in the OUTLIMITS= data set are estimates or standard values of the process mean μ and standard deviation σ. The following table summarizes the *values* that you can specify:

Value	_MEAN_	_STDDEV_
ESTIMATE	Estimate	Estimate
STDMU	Standard	Estimate
STDSIGMA	Estimate	Standard
STANDARD	Standard	Standard

The default *value* is ESTIMATE, unless you specify standard values for μ or σ with the MU0= or SIGMA0= options.

For PCHART and NPCHART statements, the *value* you specify for the TYPE= option can be either ESTIMATE or STANDARD, indicating that the value of the variable _P_ in the OUTLIMITS= data set is an estimate or standard value of the proportion p of nonconforming items. The default *value* is ESTIMATE, unless you specify a standard value for p with the P0= option.

For UCHART and CCHART statements, the *value* you specify for the TYPE= option can be either ESTIMATE or STANDARD, indicating that the value of the variable _U_ in the OUTLIMITS= data set is an estimate or standard value of the average number u of nonconformities per unit. The default *value* is ESTIMATE, unless you specify a standard value for u with the U0= option.

U0=*value*

specifies a known (standard) value u_0 for the average number u of nonconformities per unit produced by the process. By default, u_0 is estimated from the data. The U0= option is available only in the CCHART and UCHART statements.

NOTE: As an alternative to specifying the U0= option, you can read a predetermined value for u_0 from the variable _U_ in a LIMITS= data set. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.

UCLLABEL='*label*'

specifies a label for the upper control limit in the primary chart. The label can be up to 16 characters. Enclose the label in quotes. The default label is of the form *UCL=value* if the control limit has a fixed value; otherwise, the default label is *UCL*. Related options are UCLLABEL2=, LCLLABEL=, and LCLLABEL2=.

UCLLABEL2='*label*'

specifies a label for the upper control limit in the secondary chart. The label can be up to 16 characters. Enclose the label in quotes. The default label is of the form *UCL=value* if the control limit has a fixed value; otherwise, the default label is *UCL*. This option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements. Related options are LCLLABEL2=. LCLLABEL=, and UCLLABEL=,

USL=*value-list*

provides upper specification limits used to compute capability indices. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

The SHEWHART procedure uses the specification limits to compute capability indices, and it saves the limits and indices in the OUTLIMITS= data set. For more information, see "Capability Indices" on page 1971 and "Output Data Sets" in the section for the chart statement in which you are interested. A related option is LSL=. The USL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

USYMBOL='*label*'

USYMBOL=U | UBAR | UPM | UPM2 | U0

specifies a label for the central line in a *u* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
U	U	U
UBAR	$\overline{U}$	$\overline{U}$
UPM	U'	U'
UPM2	U''	U''
U0	U_0	U0

The default keyword is UBAR. The USYMBOL= option is available only in the UCHART statement.

VAXIS=*value-list*

VAXIS=AXIS*n*

specifies major tick mark values for the vertical axis of a primary chart. The *values* must be listed in increasing order, must be evenly spaced, and must span the range of summary statistics and control limits displayed in the chart. You can specify the *values* with an explicit list or with an implicit list, as shown in the following example:

```
proc shewhart;
   xrchart Width*Hour / vaxis=0 2 4 6 8;
   xrchart Width*Hour / vaxis=0 to 8 by 2;
run;
```

If you are producing traditional graphics, you can also specify a previously defined AXIS statement with the VAXIS= option. Related options are HAXIS= and VAXIS2=.

VAXIS2=*value-list*

VAXIS2=AXIS*n*

specifies major tick mark values for the vertical axis of a secondary chart. The specifications and restrictions are the same as for the VAXIS= option. The VAXIS2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option. Related options are HAXIS= and VAXIS=.

VFORMAT=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of a primary chart.

VFORMAT2=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of a secondary chart.

VOFFSET=*value*

specifies the length of the offset at each end of the vertical axis. For traditional graphics, the offset is specified in percent screen units. For ODS Graphics output, the offset is specified in pixels.

VREF=*value-list*

VREF=*SAS-data-set*

draws reference lines perpendicular to the vertical axis on the primary chart. Reference line values can be expressed as simple values or as multiples of the standard error of the subgroup summary statistic. You can use this option in the following ways:

- Specify the *values* for the lines with a VREF= list. Examples of the VREF= option follow:

  ```
  vref=20
  vref=20 40 80
  vref=(2.5 sigma)
  vref=20 (1.5 2.0 2.5 sigma) 80
  ```

  Values expressed as multiples of σ must be enclosed in parentheses with the SIGMA keyword.

- Specify the values for the lines as the values of a numeric variable named _REF_ in a VREF= data set. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be

displayed in charts for different *processes* specified in the chart statement, you must include a character variable of length 32 or less named _VAR_, whose values are the *processes*. If you do not include the variable _VAR_, all of the lines are displayed in all of the charts. If you want to display reference lines whose values are multiples of σ, you must include a character variable named _TYPE_, whose values are "VALUES" or "SIGMAS." The value of _TYPE_ indicates whether the reference line value is expressed as a simple value or as a multiple of σ.

Each observation in the VREF= data set corresponds to a reference line. If BY variables are used in the input data set (DATA=, HISTORY=, or TABLE=), the same BY variable structure must be used in the VREF= data set unless you specify the NOBYREF option.

This option can be used to add warning limits to be displayed on a chart.

Related options are CVREF=, LVREF=, NOBYREF, VREFCHAR=, VREFLABELS=, and VREFLABPOS=.

VREF2=*value-list*

VREF2=*SAS-data-set*

draws reference lines perpendicular to the vertical axis on the secondary chart. The conventions for specifying the VREF2= option are identical to those for specifying the VREF= option. Related options are CVREF=, LVREF=, NOBYREF, VREFCHAR=, VREF2LABELS=, and VREFLABPOS=.

The VREF2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

VREF2LABELS='*label1*' **...** '*labeln*'

VREF2LAB='*label1*' **...** '*labeln*'

specifies labels for the reference lines requested by the VREF2= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. The VREF2LABELS= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

VREFLABELS='*label1*' **...** '*labeln*'

specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

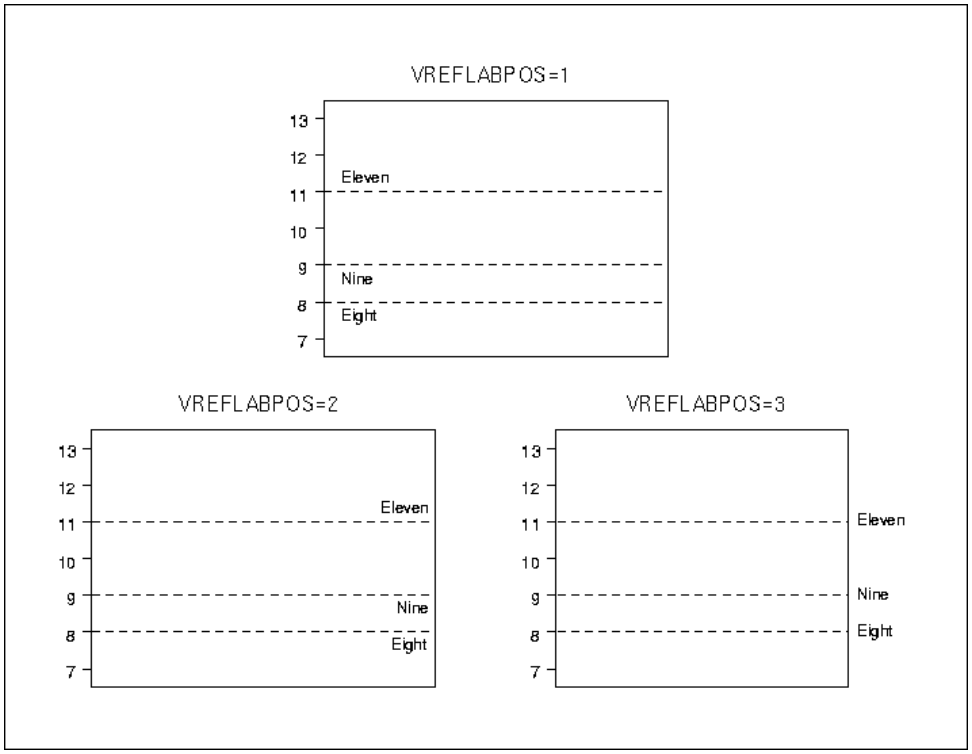
specifies the horizontal position of the VREFLABELS= and VREF2LABELS= labels, as described in the following table. By default, $n = 1$.

n	**Label Position**
1	Left-justified in subplot area
2	Right-justified in subplot area
3	Left-justified in right margin

Figure 19.141 illustrates label positions for values of the VREFLABPOS= option when the VREF= and VREFLABELS= options are as follows:

```
vref       = 8 9 11
vreflabels = 'Eight' 'Nine' 'Eleven'
```

Figure 19.141 Positions for Reference Line Labels



VZERO
forces the origin to be included in the vertical axis for a primary chart.

VZERO2
forces the origin to be included in the vertical axis for a secondary chart.

WESTGARD=*index-list*
requests that one or more of the Westgard rules be applied. The Westgard rules are tests for special causes that were developed specifically for use in healthcare laboratories. Westgard (2002) describes the rules and their proper use in detail.

The Westgard rules are similar to the Western Electric rules that are implemented by the TESTS= option. They detect unusual patterns of points plotted on the primary control chart. The patterns are defined in terms of the zones A, B, and C that are illustrated in Figure 19.178. The occurrence of one or more of these patterns suggests the presence of a special cause of variation.

Table 19.93 lists the Westgard tests that you can request.

Table 19.93 Westgard Rules

Index	Notation	Pattern Description
1	1:2s	One point in Zone A or beyond
2	1:3s	One point beyond Zone A (outside the control limits)
3	2:2s	Two points in a row in Zone A or beyond on the same side of the central line
4	R:4s	At least one point in Zone A or beyond on each side of the central line
5	4:1s	Four points in a row in Zone B or beyond on the same side of the central line
6	10x	Ten points in a row on the same side of the central line

WHISKERPERCENTILE=*pctl*

specifies that the whiskers of the box-and-whisker plots be drawn to the *pctl* and 100 − *pctl* percentiles. For example, if you specify WHISKERPERCENTILE=10 the whiskers are drawn to the 10th and 90th percentiles. Observations that lie beyond the whiskers are outliers, and there are no far outliers. This option in available only in the BOXCHART statement.

XSYMBOL='*label***'**

XSYMBOL=*keyword*

specifies a label for the central line in an $\bar{X}$ chart or a median chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the *keywords* listed in the following table. Each *keyword* requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
MBAR	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$
MTIL	$\tilde{\mathrm{M}}$	$\tilde{\mathrm{M}}$
MU	μ	MU
MU0	μ_0	MU0
XBAR	$\overline{\mathrm{X}}$	$\overline{\mathrm{X}}$
XBAR2	$\overline{\overline{\mathrm{X}}}$	$\overline{\overline{\mathrm{X}}}$
XBARPM	$\overline{\mathrm{X}}'$	$\overline{\mathrm{X}}'$
XBAR0	$\overline{\mathrm{X}}_0$	$\overline{\mathrm{X}}0$
XBAR0PM	$\overline{\mathrm{X}}_0'$	$\overline{\mathrm{X}}0'$

For the IRCHART statement, the default *keyword* is XBAR. For the MCHART and MRCHART statements, the default *keyword* is MBAR. For all other chart statements, the default *keyword* is XBAR2. The XSYMBOL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, XCHART, XRCHART, and XSCHART statements.

YPCT1=*value*
specifies a percent (ranging from 0 to 100) that determines the length of the vertical axis for the primary chart in proportion to the sum of the lengths of the vertical axes for the primary and secondary charts. For example, you can specify YPCT1=50 in an XRCHART statement to request that the vertical axes for the $\bar{X}$ and R charts have the same length. The default *value* is 60. The YPCT1= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

YSCALE=PERCENT
scales the vertical axis on a p chart in percent units. The YSCALE= option is available only in the PCHART statement.

ZEROSTD
ZEROSTD=NOLIMITS
specifies that a control chart is to be constructed and displayed regardless of whether the estimated process standard deviation $\hat{\sigma}$ is zero. When $\hat{\sigma}$ is zero, the control limits are degenerate (collapsed around the central line), and the chart simply serves as a placeholder, particularly when a series of charts is to be created. Specify ZEROSTD=NOLIMITS to suppress the display of the degenerate limits. By default, a chart is not displayed when $\hat{\sigma}$ is zero.

ZONE2LABELS
adds the labels A, B, and C to zone lines requested with the ZONES2 or ZONE2VALUES options. The ZONE2LABELS option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

ZONE2VALUES
labels R or s chart zones lines with their values. If the ZONE2VALUES option is specified the ZONES2 option is not required.

ZONELABELS
adds the labels A, B, and C to zone lines requested with the ZONES or ZONEVALUES options. The ZONELABELS option is not available in the RCHART or SCHART statements.

ZONES
adds lines to a primary chart that delineate zones A, B, and C for standard tests requested with the TESTS= option. Related options are CZONES= and ZONELABELS. The ZONES option is not available in the RCHART or SCHART statements.

ZONES2
adds lines to an R or s chart that delineate zones A, B, and C for tests requested with the TESTS2= option. Related options are CZONES= and ZONE2LABELS. The ZONES2 option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

ZONEVALPOS=*n*
specifies the horizontal position of the ZONEVALUES= and ZONE2VALUES= labels, as described in the following table. By default, $n = 1$.

n	**Label Position**
1	Left-justified in subplot area
2	Right-justified in subplot area
3	Left-justified in right margin

ZONEVALUES
labels the primary chart zones lines with their values. If the ZONEVALUES option is specified, the ZONES option is not required.

Options for ODS Graphics

BLOCKREFTRANSPARENCY=*value*

PHASEREFTRANSPARENCY=*value*

REFFILLTRANSPARENCY=*value*
specifies the wall fill transparency for blocks and phases when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default wall fill transparency is 0.85.

BOXTRANSPARENCY=*value*
specifies the box fill transparency for box-and-whisker charts when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default box fill transparency is 0.25.

INFILLTRANSPARENCY=*value*
specifies the control limit infill transparency when transparency is used in ODS Graphics output. The *value* be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default control limit infill transparency is 0.75.

MARKERDISPLAY=OOC | UPPER | LOWER | RUNSTEST
specifies a subset of subgroups to be plotted with markers on a primary chart that is not an *R* or *s* chart. You can select the following subsets of subgroups:

Keyword	Subgroups Plotted with Markers
OOC	Specifies subgroups outside the control limits
UPPER	Specifies subgroups above the upper control limit
LOWER	Specifies subgroups below the lower control limit
RUNSTEST	Specifies subgroups that signal positive tests for special causes

If you specify a *symbol-variable*, subgroups that are associated with the same *symbol-variable* value are plotted with the same marker. The MARKERDISPLAY= option overrides the MARKERS option.

You can use the MARKERLABEL= option to label the selected subgroups. You can use the MARKERDISPLAY2= option to select subgroups to be plotted with markers on an *R* or *s* chart.

The MARKERDISPLAY= option is ignored when it is specified in a BOXCHART statement.

MARKERDISPLAY2=OOC | UPPER | LOWER | RUNSTEST
specifies a subset of subgroups to be plotted with markers on an *R* or *s* chart. You can select the following subsets of subgroups:

Keyword	Subgroups Plotted with Markers
OOC	Specifies subgroups outside the control limits
UPPER	Specifies subgroups above the upper control limit
LOWER	Specifies subgroups below the lower control limit
RUNSTEST	Specifies subgroups that signal positive tests for special causes

If you specify a *symbol-variable*, subgroups that are associated with the same *symbol-variable* value are plotted with the same marker. The MARKERDISPLAY2= option overrides the MARKERS option.

You can use the MARKERLABEL2= option to label the selected subgroups.

MARKERLABEL=(*variable*)
specifies a variable whose values provide labels for the subgroups that are plotted with markers on a primary chart that is not an *R* or *s* chart.

The MARKERMISSINGGROUP= and MARKERDISPLAY= options provide methods of specifying which subgroups are plotted with markers. The ALLLABEL= and OUTLABEL= options provide other alternatives for labeling subgroups.

The MARKERLABEL= option is ignored when it is specified in a BOXCHART statement.

MARKERLABEL2=(*variable*)
specifies a variable whose values provide labels for the subgroups that are plotted with markers on an *R* or *s* chart.

The MARKERMISSINGGROUP= and MARKERDISPLAY2= options provide methods of specifying which subgroups are plotted with markers. The ALLLABEL2= and OUTLABEL2= options provide other alternatives for labeling subgroups.

MARKERMISSINGGROUP=TRUE | FALSE
specifies whether subgroups with missing *symbol-variable* values are plotted with a unique marker. The default is MARKERMISSINGGROUP=TRUE. This option is ignored when it is specified in a BOXCHART statement.

You can specify MARKERMISSINGGROUP=FALSE to suppress markers for subgroups with missing *symbol-variable* values. Always use a connecting line or needles to ensure that all the data are represented on the chart.

By suppressing markers for missing *symbol-variable* values, you can use markers to emphasize important data in the chart. You can specify MARKERMISSINGGROUP=FALSE and assign nonmissing values to the *symbol-variable* for the subgroups that you want to emphasize. Assign missing values to the *symbol-variable* for the remaining subgroups.

By default, the *symbol-variable* values are summarized in the symbol legend as marker-label entries. The legend is useful for identifying these points when you have several long *symbol-variable* values (labels). Alternatively, use the MARKERLABEL= option to label the markers directly on the chart. If the symbol legend contains redundant information, you can suppress it by specifying the SYMBOLLEGEND=NONE option.

For example, to render only specific subgroups in a chart with the default marker, do the following:

- Create a variable to be used as a *symbol-variable*.
- Assign a constant group value to all the points that you want to represent with a marker.

- Assign missing values to suppress the markers at the other points.
- Assign ODS style element attributes to the reserved, SAS macro variables &_COLOR and &_CONTRAST.

Then modify the following code snippet to produce an appropriate graph:

```
%let _color    = GraphDataDefault:color;
%let _contrast = GraphDataDefault:contrastcolor;

proc shewhart;
   xchart x*i = constant / markermissinggroup = false
                           symbollegend = none;
   run;

%let _color    = ;  * reset;
%let _contrast = ;  * reset;
%let _symbol   = ;  * reset;
```

For more examples, see the sections "ODS Graphics Template" on page 1968 and "Displaying Stratification in Levels of a Classification Variable" on page 2073.

MARKERS
plots subgroup points with markers. By default, subgroup points are plotted with markers only by the BOXCHART statement. On other types of charts, subgroup points are connected by line segments and are not plotted with markers by default.

NOBLOCKREF

NOPHASEREF

NOREF
suppresses block and phase reference lines from ODS Graphics output. By default, block and phase reference lines are drawn when ODS Graphics is in effect.

NOBLOCKREFFILL

NOPHASEREFFILL

NOREFFILL
suppresses the block and phase wall fills from ODS Graphics output. By default, block and phase walls are filled when ODS Graphics is in effect.

NOBOXFILLLEGEND

NOFILLLEGEND

NOSTARFILLLEGEND
suppresses the legend for the levels of a BOXFILL= or STARFILL= variable in ODS Graphics output.

NOTRANSPARENCY
disables transparency in ODS Graphics output, so that all graph features are opaque. By default, transparency is enabled when ODS Graphics is in effect.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string*'
adds a footnote to ODS Graphics output. If you specify the FOOTNOTE (or FOOTNOTE1) keyword, the value of SAS FOOTNOTE statement is used the as the graph footnote. If you specify a quoted string, that is used as the footnote. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

ODSFOOTNOTE2=FOOTNOTE2 | '*string*'
adds a secondary footnote to ODS Graphics output. If you specify the FOOTNOTE2 keyword, the value of SAS FOOTNOTE2 statement is used as the secondary graph footnote. If you specify a quoted string, that is used as the secondary footnote. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

ODSLEGENDEXPAND
specifies that legend entries contain all levels observed in the data. By default, a legend shows only the levels used on the current page.

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | LABELFMT | '*string*'
specifies a title for ODS Graphics output.

TITLE (or TITLE1)	uses the value of SAS TITLE statement as the graph title.
NONE	suppresses all titles from the graph.
DEFAULT	uses the default ODS Graphics title (a descriptive title consisting of the plot type and the process variable name.)
LABELFMT	uses the default ODS Graphics title with the variable label instead of the variable name.

If you specify a quoted string, that is used as the graph title. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

ODSTITLE2=TITLE2 | '*string*'
specifies a secondary title for ODS Graphics output. If you specify the TITLE2 keyword, the value of SAS TITLE2 statement is used as the secondary graph title. If you specify a quoted string, that is used as the secondary title. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

OUTFILLTRANSPARENCY=*value*
specifies the control limit outfill transparency when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default control limit outfill transparency is 0.75.

OUTHIGHURL=*variable*
specifies a variable whose values are URLs to be associated with outlier points above the upper fence on a schematic box chart when ODS Graphics output is directed into HTML.

OUTLOWURL=*variable*
specifies a variable whose values are URLs to be associated with outlier points below the lower fence on a schematic box chart when ODS Graphics output is directed into HTML.

OVERLAY2URL=(*variable-list*)
specifies variables whose values are URLs to be associated with points on secondary chart overlays. These URLs are associated with points on an overlay plot when ODS Graphics output is directed into HTML. Variables in the OVERLAY2URL= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2URL= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYURL=(*variable-list*)
specifies variables whose values are URLs to be associated with points on primary chart overlays. These URLs are associated with points on an overlay plot when ODS Graphics output is directed into HTML. Variables in the OVERLAYURL= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYURL= variable should be the same for each observation with a given value of the subgroup variable.

PHASEBOXLABELS
draws phase labels as titles along the top of phase boxes.

PHASEPOS=*n*
specifies the vertical position of the phase legend. Values of *n* and the corresponding positions are as follows. By default, PHASEPOS=1.

n	**Legend Position**
1	Top of chart, offset from axis frame
2	Top of chart, immediately above axis frame
3	Bottom of chart, immediately above horizontal axis
4	Bottom of chart, below horizontal axis label

PHASEREFLEVEL=INNER | OUTER | NONE
enables you to associate phase reference lines (block reference lines) with either the innermost or the outermost level. The default value is INNER.

POINTSURL=*variable*
specifies a variable whose values are URLs to be associated with points on a box chart when the BOXSTYLE= value is POINTS, POINTSJOIN, POINTSBOX, POINTSID, or POINTSJOINID. These URLs are associated with points on a box chart when ODS Graphics output is directed into HTML.

SIMULATEQCFONT
draws the central line labels using a simulated software font rather than a hardware font.

STARTRANSPARENCY=*value*
specifies the star fill transparency when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default star fill transparency is 0.25.

URL=*variable*
specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with subgroup points on a primary control chart when ODS Graphics output is directed into HTML. The value of the URL= variable should be the same for each observation with a given value of the subgroup variable.

URL2=*variable*
specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with subgroup points on a secondary control chart when ODS Graphics output is directed into HTML. The value of the URL2= variable should be the same for each observation with a given value of the subgroup variable.

WBOXES=*n*
specifies the width in pixels for the outlines of the box-and-whisker plots created with the BOXCHART statement in ODS Graphics output.

Options for Traditional Graphics

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*
specifies an ANNOTATE= type data set, as described in *SAS/GRAPH: Reference*, that enhances a primary chart. The ANNOTATE= data set specified in a chart statement enhances all charts created by that particular statement. You can also specify an ANNOTATE= data set in the PROC SHEWHART statement to enhance all primary charts created by the procedure.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*
specifies an ANNOTATE= type data set, as described in *SAS/GRAPH: Reference*, that enhances a secondary chart. The ANNOTATE2= data set specified in a chart statement enhances all charts created by that particular statement. You can also specify an ANNOTATE2= data set in the PROC SHEWHART statement to enhance all secondary charts created by the procedure.

This option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

BILEVEL

arranges the Shewhart chart in two levels (rather than the default of one level) so that twice as much data can be displayed on a page or screen. The second level is a continuation of the first level, and this arrangement is continued on subsequent pages until all the subgroups are displayed. You use the NPANELPOS= option to control the number of subgroup positions in each level. If you specify the BILEVEL option in a chart statement that produces primary and secondary charts, you must also specify the SEPARATE option.

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement.

CBLOCKLAB=*color* | **(***color-list***)**

specifies fill colors for the frames that enclose the *block-variable* labels in a block legend. By default, these areas are not filled. Colors in the CBLOCKLAB= list are matched with *block-variables* in the order in which they appear in the chart statement. Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, and CBLOCKVAR=.

CBLOCKVAR=*variable* | **(***variable-list***)**

specifies variables whose values are colors for filling the background of the legend associated with *block-variables*. Each CBLOCKVAR= variable must be a character variable of no more than eight characters in the input data set (a DATA=, HISTORY=, or TABLE= data set). A list of CBLOCKVAR= variables must be enclosed in parentheses. You can use the BLOCKVAR= option to specify that the block variable legend be filled with different colors from the ODS style.

The procedure matches the CBLOCKVAR= variables with *block-variables* in the order specified. That is, each block legend will be filled with the color value of the CBLOCKVAR= variable of the first observation in each block. In general, values of the *i*th CBLOCKVAR= variable are used to fill the block of the legend corresponding to the *i*th *block-variable*. For examples of the CBLOCKVAR= option, see Figure 19.146 and Figure 19.147.

By default, fill colors are not used for the *block-variable* legend. The CBLOCKVAR= option is available only when *block-variables* are used in the chart statement.

CBOXES=*color*

CBOXES=(*variable***)**

specifies the colors for the outlines of the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify CBOXES=*color* to provide a single outline color for all the box-and-whisker plots.
- You can specify CBOXES=(*variable*) to provide a distinct outline color for *each* box-and-whisker plot as the value of the *variable*. The *variable* must be a character variable of length 8 less in the input data set, and its values must be valid SAS/GRAPH color names. The outline color of the plot displayed for a particular subgroup is the value of the *variable* in the observations

corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all the observations in a given subgroup.

You can use the BOXES= option to group boxes to be drawn with different colors from the ODS style.

The CBOXES= option is available only in the BOXCHART statement.

CBOXFILL=*color*

CBOXFILL=(*variable***)**

specifies the interior fill colors for the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify CBOXFILL=*color* to provide a single color for all of the box-and-whisker plots.
- You can specify CBOXFILL=(*variable*) to provide a distinct color for *each* box-and-whisker plot as the value of the *variable*. The *variable* must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH color names (or the value *EMPTY*, which you can use to suppress color filling). The interior color of the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all the observations in a given subgroup.

You can use the BOXFILL= option to group boxes to be filled with different colors from the ODS style. By default, all boxes are filled with a single color from the ODS style. The CBOXFILL= option is available only in the BOXCHART statement.

CCLIP=*color*

specifies a color for the plotting symbol that is specified with the CLIPSYMBOL= option to mark clipped points. The default color is the color specified in the COLOR= option in the SYMBOL1 statement.

CCONNECT=*color*

specifies the color for the line segments connecting points on the chart. The default color is the color specified in the COLOR= option in the SYMBOL1 statement. This option is not applicable in the BOXCHART statement unless you also specify the BOXCONNECT option.

CCOVERLAY=(*color-list***)**

specifies the colors for the line segments connecting points on primary chart overlays. Colors in the CCOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points on an overlay.

CCOVERLAY2=(*color-list***)**

specifies the colors for the line segments connecting points on secondary chart overlays. Colors in the CCOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points on an overlay.

CFRAME=*color*

CFRAME=(*color-list***)**

specifies the colors for filling the rectangle enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option.

You can specify a single *color* to fill the entire area. Alternatively, if you are displaying phases (blocks) of data read with the READPHASES= option, you can specify a *color-list* with the CFRAME= option to fill the sub-rectangles of the framed area corresponding to the phases. The colors, in order of specification, are applied to the sub-rectangles starting from left to right. You can use the value *EMPTY* in the *color-list* to avoid filling a particular sub-rectangle. If the number of colors is less than the number of phases, the colors are applied cyclically. The colors are also used for phase legends requested with the PHASELEGEND option.

CGRID=*color*

specifies the color for the grid requested by the ENDGRID or GRID option. By default, the grid is the same color as the axes.

CHREF=*color*

specifies the color for the lines requested by the HREF= and HREF2= options.

CLABEL=*color*

specifies the color for labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options.

CLIMITS=*color*

specifies the color for the control limits, the central line, and the labels for these lines.

CLIPLEGPOS=TOP | BOTTOM

specifies the position for the legend that indicates the number of clipped points when the CLIPFACTOR= option is used. The keywords TOP and BOTTOM position the legend at the top or bottom of the chart, respectively. Do not specify CLIPLEGPOS=TOP together with the PHASELEGEND option or the BLOCKPOS=1 or BLOCKPOS=2 options. By default, CLIPLEGPOS=BOTTOM.

CLIPSYMBOL=*symbol*

specifies a plot symbol used to identify clipped points on the chart and in the legend when the CLIPFACTOR= option is used. You should use this option in conjunction with the CLIPFACTOR= option. The default *symbol* is CLIPSYMBOL=SQUARE.

CLIPSYMBOLHT=*value*

specifies the height for the symbol marker used to identify clipped points on the chart when the CLIPFACTOR= option is used. The default is the height specified with the H= option in the SYMBOL statement.

For general information about clipping options, refer to "Clipping Extreme Points" on page 2105.

CNEEDLES=*color*

requests that points are to be connected to the central line with vertical line segments (needles) and specifies the color of the needles. You can use needles to visually represent the process as a series of shocks or vertical displacements away from a constant mean. See Figure 19.168 for an example. The CNEEDLES= option is available in all chart statements except the BOXCHART statement.

COUTFILL=*color*

specifies the fill color for the areas outside the control limits that lie between the connected points and the control limits and are bounded by connecting lines. This option is useful for highlighting out-of-control points. See Figure 19.203 for an example. By default, these areas are not filled. You can use the OUTFILL option to fill this area with an appropriate color from the ODS style. Note that you can use the CINFILL= option to fill the area inside the control limits.

COVERLAY=(*color-list***)**

specifies the colors used to plot primary chart overlay variables. Colors in the COVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

COVERLAY2=(*color-list***)**

specifies the colors used to plot secondary chart overlay variables. Colors in the COVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

COVERLAYCLIP=*color*

specifies the color used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

CPHASELEG=*color*

specifies a text color for the phase labels requested with the PHASELEGEND option. By default, if you specify a list of fill colors with the CFRAME= option, these colors are used for the corresponding phase labels, otherwise, the CTEXT= color is used for the phase labels.

CSTARCIRCLES=*color*

specifies a color for the circles requested with the STARCIRCLES= option. See "Displaying Auxiliary Data with Stars" on page 2090. By default, the color specified with the CSTARS= option is used.

CSTARFILL=*color*

CSTARFILL=(*variable***)**

specifies a color or colors for filling the interior of stars requested with the STARVERTICES= option. You can use one of the following approaches:

- Specify a single color to be used for all stars with CSTARFILL=*color*.
- Specify a distinct color for *each* star (or subsets of stars) by providing the colors as values of a variable specified with CSTARFILL=(*variable*). The variable must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH colors or the value *EMPTY*. The color for the star positioned at the *i*th subgroup on the chart is the value of the CSTARFILL= *variable* in the observations corresponding to the *i*th subgroup. Note that if there are multiple observations per subgroup in the input data set (for instance, if you are using the XRCHART statement in the SHEWHART procedure to analyze observations from a DATA= input data set), the values of the CSTARFILL= *variable* should be identical for all the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 2090.

You can use the STARFILL= option to group stars to be filled with different colors from the ODS style. By default, all stars are filled with a single color from the ODS style.

CSTARS=*color*

CSTARS=(*variable***)**

specifies a color or colors for the outlines of stars requested with the STARVERTICES= option.

You can use one of the following approaches:

- You can specify a single color to be used for all the stars on the chart with CSTARS=*color*.
- You can specify a distinct outline color for *each* star (or subsets of stars) by providing the colors as values of a variable specified with CSTARS=(*variable*). The variable must be a character variable of length 8 or less in the input data set. The outline color for the star positioned at the *i*th subgroup on the chart is the value of the CSTARS=*variable* in the observations corresponding to the *i*th subgroup. Note that if there are multiple observations per subgroup in the input data set (for instance, if you are using the XRCHART statement in the SHEWHART procedure to analyze observations from a DATA= input data set), the values of the CSTARS= *variable* should be identical for all the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 2090.

You can use the STARS= option to group stars to be drawn with different colors from the ODS style. By default, all stars are drawn with a single color from the ODS style.

CTESTLABBOX=*color*

specifies the color for boxes enclosing labels for positive tests for special causes requested with the TESTLABBOX option. If you use the CTESTLABBOX= option, you do not need to specify the TESTLABBOX option.

CTESTS=*color* | *test-color-list*

CTEST=*color* | *test-color-list*

specifies colors for labels indicating points where a test is positive.

- You can specify the *color* for the labels used to identify points at which tests for special causes specified in the TESTS= option are positive. For Tests 2 through 8, this color is also used for the line segments that connect patterns of points for which a test is positive.
- You can specify the *test-color-list* to enable different colors to be used for the labels and highlighted line segments associated with different tests for special causes. Any positive tests for which no specific CTESTS= value is specified are displayed using the general CTESTS= color. A non-default general CTESTS= color can be specified using the CTESTS=*color* syntax.

The following options request the standard tests for special causes 1 through 4 and one user-defined test designated B.

```
TESTS  = 1 to 4 M(K=4 DIR=DEC Code=B);
CTESTS = green;
CTESTS = (1 purple 3 yellow B blue);
```

Test 1 will be displayed in purple, Test 3 in yellow, and Test B in blue. Tests 2 and 4 will be displayed in green, the general CTESTS= color.

CTESTSYMBOL=*color*

CTESTSYM=*color*

specifies the color of the symbol used to plot subgroups with positive tests for special causes.

CTEXT=*color*

specifies the color for tick mark values and axis labels. This color is also used for the sample size legend and for the control limit legend. The default color is the color specified in the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for reference lines requested by the VREF= and VREF2= options.

CZONES=*color*

requests lines marking zones A, B, and C for the tests for special causes (see the TESTS= option) and specifies the *color* for these lines. This color is also used for labels requested with the ZONELABELS option.

DESCRIPTION='*string*'

DES='*string*'

specifies a description, up to 256 characters long, for the GRSEG catalog entry for the primary chart. The default *string* is the variable name. A related option is NAME=.

DESCRIPTION2='*string*'

DES2='*string*'

specifies a description, up to 256 characters long, for the GRSEG catalog entry for the secondary chart. The default *string* is the variable name. The DESCRIPTION2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements, and it is used in conjunction with the SEPARATE option. A related option is NAME2=.

ENDGRID

adds a grid to the rightmost portion of the chart, beginning with the first labeled major tick mark position that follows the last plotted point. This grid is useful in situations where you want to add points by hand after the chart is created. You can use the HAXIS= option to force space to be added to the horizontal axis.

FONT=*font*

specifies a software font for labels and legends. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the GOPTIONS statement. Hardware characters are used by default.

HEIGHT=*value*

specifies the height (in vertical screen percent units) of the text for axis labels and legends. This *value* takes precedence over the HTEXT= value specified in the GOPTIONS statement. This option is recommended for use with software fonts specified with the FONT= option or with the FTEXT= option in the GOPTIONS statement. Related options are LABELHEIGHT= and TESTHEIGHT=.

HMINOR=*n*

HM=*n*

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0.

HTML=*variable*

specifies a variable whose values create links associated with subgroup points on a primary control chart when traditional graphics output is directed into HTML. You can specify a character variable or formatted numeric variable. The value of the HTML= variable should be the same for each observation with a given value of the subgroup variable. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2183 for more information.

HTML2=*variable*

specifies a variable whose values create links associated with subgroup points on a secondary control chart when traditional graphics output is directed into HTML. You can specify a character variable or formatted numeric variable. The value of the HTML2= variable should be the same for each observation with a given value of the subgroup variable. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2183 for more information.

HTML_LEGEND=*variable*

specifies HTML links as values of the specified character variable (or formatted values of a numeric variable). These links are associated with symbols in the legend for the levels of a *symbol-variable*. The value of the HTML_LEGEND= variable should be the same for each observation with a given value of *symbol-variable*.

IDCOLOR=*color*

specifies the color of the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement when you use one of the following options: BOXSTYLE=SCHEMATIC, BOXSTYLE=SCHEMATICID, and BOXSTYLE=SCHEMATICIDFAR. The default *color* is the color specified with the CBOXES= option. The IDCOLOR= option is available only in the BOXCHART statement.

IDCTEXT=*color*

specifies the color for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default is the color specified with the CTEXT= option.

IDFONT=*font*

specifies the font for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default *font* is SIMPLEX.

IDHEIGHT=*value*

specifies the height for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default is the height specified with the HTEXT= option in the GOPTIONS statement.

IDSYMBOL=*symbol*
specifies the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement when you use one of the following options: BOXSTYLE=SCHEMATIC, BOXSTYLE=SCHEMATICID, and BOXSTYLE=SCHEMATICIDFAR. The default *symbol* is SQUARE. The IDSYMBOL= option is available only in the BOXCHART statement.

IDSYMBOLHEIGHT=*value*
specifies the height of the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement. This option is available only in the BOXCHART statement.

LABELANGLE=*angle*
specifies the angle at which labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options are drawn. A positive angle rotates the labels counterclockwise; a negative angle rotates them clockwise. By default, labels are oriented horizontally.

LABELFONT=*font*
TESTFONT=*font*
specifies a software font for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. Hardware characters are used by default.

LABELHEIGHT=*value*
TESTHEIGHT=*value*
specifies the height (in vertical percent screen units) for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. The default height is the height specified with the HEIGHT= option or the HTEXT= option in the GOPTIONS statement.

LBOXES=*linetype*
LBOXES=(*variable***)**
specifies the line types for the outlines of the box-and-whisker plots created with the BOXCHART statement. eou can use one of the following approaches:

- You can specify LBOXES=*linetype* to provide a single *linetype* for all of the box-and-whisker plots.
- You can specify LBOXES=(*variable*) to provide a distinct line type for *each* box-and-whisker plot. The *variable* must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH *linetype* values (numbers ranging from 1 to 46). The line type for the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all of the observations in a given subgroup.

The default value is 1, which produces solid lines. The LBOXES= option is available only in the BOXCHART statement.

LENDGRID=*n*
specifies the line type for the grid requested with the ENDGRID option. The default is $n = 1$, which produces a solid line. If you use the LENDGRID= option, you do not need to specify the ENDGRID option.

LGRID=*n*

specifies the line type for the grid requested with the GRID option. The default is *n* = 1, which produces a solid line. If you use the LGRID= option, you do not need to specify the GRID option.

LHREF=*linetype*

LH=*linetype*

specifies the line type for reference lines requested with the HREF= and HREF2= options. The default is 2, which produces a dashed line.

LLIMITS=*linetype*

specifies the line type for control limits. The default is 4, which produces a dashed line.

LOVERLAY=(*linetypes***)**

specifies line types for the line segments connecting points on primary chart overlays. Line types in the LOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

LOVERLAY2=(*linetypes***)**

specifies line types for the line segments connecting points on secondary chart overlays. Line types in the LOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

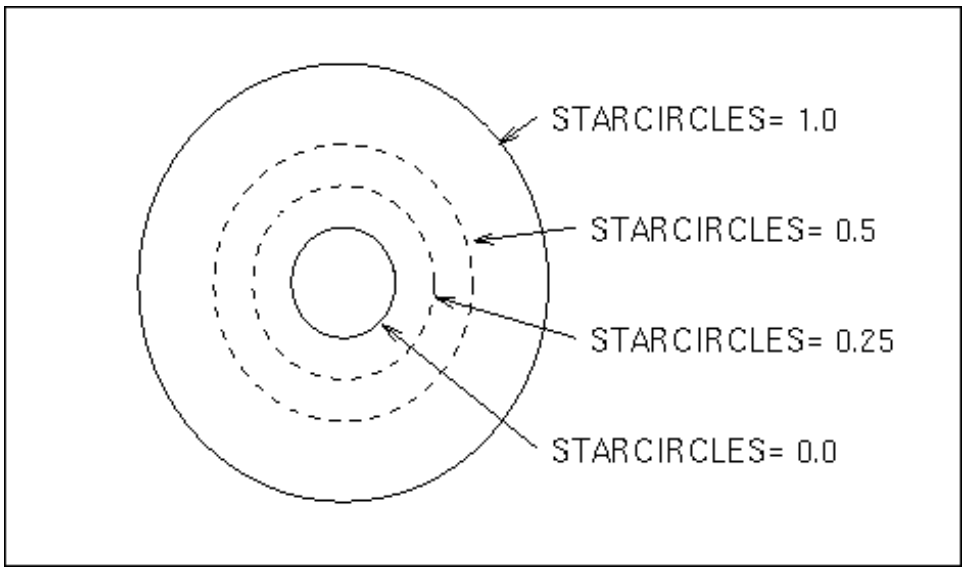
LSTARCIRCLES=*linetypes*

specifies one or more line types for the circles requested with the STARCIRCLES= option. The number of line types should match the number of circles requested, and the line types are paired with the circles in the order specified. The default *linetype* is 1, which produces a solid line.

Figure 19.142 illustrates circles displayed by the following LSTARCIRCLES= and STARCIRCLES= options:

```
starcircles  = 0.0 1.0 0.25 0.5
lstarcircles = 1 1 2 2
```

Figure 19.142 Line Types for Reference Circles



LSTARS=*linetype*

LSTARS=(*variable***)**

specifies the line types for the outlines of stars requested with the STARVERTICES= option. You can use one of the following approaches:

- You can specify LSTARS=*linetype* to provide a single line type for all of the stars.
- You can specify LSTARS=(*variable*) to provide a distinct line type for *each* star. The variable must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH line types. The line type for the star positioned at a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the *variable* values should be identical for all of the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 2090. The default *linetype* is 1, which produces a solid line.

LTESTS=*linetype*

LTEST=*linetype*

specifies the line type for the line segments that connect patterns of points for which a test for special causes (requested with the TESTS= option) is positive. The default is 1, which produces a solid line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for reference lines requested by the VREF= and VREF2= options. The default is 2, which produces a dashed line.

LZONES=*n*

specifies the line type for lines that delineate zones A, B, and C for standard tests requested with the TESTS= and/or TESTS2= options. The default is $n = 2$, which produces a dashed line.

NAME='*string*'

specifies the name of the GRSEG catalog entry for the primary chart, and the name of the graphics output file if one is created. The name can be up to 256 characters long, but the GRSEG name is truncated to eight characters. The default name is 'SHEWHART'. A related option is DESCRIPTION=.

NAME2='*string*'

specifies the name of the GRSEG catalog entry for the secondary chart, and the name of the graphics output file if one is created. The name can be up to 256 characters long, but the GRSEG name is truncated to eight characters. The default name is 'SHEWHART'. The NAME2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements, and it is used in conjunction with the SEPARATE option. A related option is DESCRIPTION2=.

NOFRAME

suppresses the default frame drawn around the chart.

NOLIMITSFRAME

suppresses the default frame for the control limit information that is displayed across the top of the chart when multiple sets of control limits with distinct multiples of σ and nominal control limit sample sizes are read from a LIMITS= data set.

NOPHASEFRAME
suppresses the default frame for the legend requested by the PHASELEGEND option.

NOVANGLE
requests vertical axis labels that are strung out vertically. By default, the labels are drawn at an angle of 90 degrees if a software font is used.

NOVLABEL
suppresses the label for the primary vertical axis. Use the NOVLABEL option when the meaning of the primary vertical axis is evident from the tick mark labels.

NOV2LABEL
suppresses the label for the secondary vertical axis. Use the NOV2LABEL option when the meaning of the secondary vertical axis is evident from the tick mark labels.

OUTHIGHHTML=*variable*
specifies a variable whose values create links to be associated with outlier points above the upper fence on a schematic box chart when traditional graphics output is directed into HTML.

OUTLOWHTML=*variable*
specifies a variable whose values create links to be associated with outlier points below the lower fence on a schematic box chart when traditional graphics output is directed into HTML.

OVERLAY2HTML=(*variable-list***)**
specifies variables whose values create links to be associated with points on secondary chart overlays. These links are associated with points on an overlay plot when traditional graphics output is directed into HTML. Variables in the OVERLAY2HTML= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2HTML= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAY2SYM=(*symbol-list***)**
specifies symbols used to plot overlays on a secondary control chart. Symbols in the OVERLAY2SYM= list are matched with variables in the corresponding positions in the OVERLAY2= list.

OVERLAY2SYMHT=(*value-list***)**
specifies the heights of symbols used to plot overlays on a secondary control chart. Heights in the OVERLAY2SYMHT= list are matched with variables in the corresponding positions in the OVERLAY2= list.

OVERLAYCLIPSYM=*symbol*
specifies the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYCLIPSYMHT=*value*
specifies the height for the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYHTML=(*variable-list***)**
specifies variables whose values create links to be associated with points on primary chart overlays. These links are associated with points on an overlay plot when traditional graphics output is directed into HTML. Variables in the OVERLAYHTML= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYHTML= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYSYM=(*symbol-list*)
specifies symbols used to plot overlays on the primary control chart. Symbols in the OVERLAYSYM= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYSYMHT=(*value-list*)
specifies the heights of symbols used to plot overlays on the primary control chart. Heights in the OVERLAYSYMHT= list are matched with variables in the corresponding positions in the OVERLAY= list.

POINTSHTML=*variable*
specifies a variable whose values create links to be associated with points on a box chart when the BOXSTYLE= value is POINTS, POINTSJOIN, POINTSBOX, POINTSID, or POINTSJOINID. These URLs are associated with points on a box chart when traditional graphics output is directed into HTML.

TESTFONT=*font*
LABELFONT=*font*
specifies a software font for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. Hardware characters are used by default.

TESTHEIGHT=*value*
LABELHEIGHT=*value*
specifies the height (in vertical percent screen units) for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. The default height is the height specified with the HEIGHT= option or the HTEXT= option in the GOPTIONS statement.

TESTSYMBOL=*symbol*
TESTSYM=*symbol*
specifies the symbol for plotting subgroups with positive tests for special causes.

TESTSYMBOLHT=*value*
TESTSYMHT=*value*
specifies the height of the symbol used to plot subgroups with positive tests for special causes.

TURNALL
TURNOUT
turns the labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options so that they are strung out vertically. By default, labels are arranged horizontally.

TURNHLABELS
TURNHLABEL
turns the major tick mark labels for the horizontal (subgroup) axis so that they are strung out vertically. By default, labels are arranged horizontally.

If you are producing traditional graphics with the NOGSTYLE option in effect, you should specify a font (with the FONT= option) in conjunction with the TURNHLABELS option. Otherwise, the labels might be displayed with a mixture of hardware and software fonts.

NOTE: Turning the labels vertically might leave insufficient room on the screen or page for a chart.

VMINOR=*n*

VM=*n*

specifies the number of minor tick marks between each major tick mark on the vertical axis. No values are printed on the minor tick marks. By default, VMINOR=0.

WAXIS=*n*

specifies the width in pixels for the axis and frame lines. By default, *n* = 1.

WEBOUT=*SAS-data-set*

produces an output data set containing all the data in an OUTTABLE= data set plus graphics coordinates for points (subgroup summary statistics) that are displayed on a control chart. You can use an WEBOUT= data set to facilitate the development of web-based applications. See "Interactive Control Charts: SHEWHART Procedure" on page 2183 for details.

WGRID=*n*

specifies the width in pixels for grid lines requested with the ENDGRID and GRID options. By default, *n* = 1.

WLIMITS=*n*

specifies the width in pixels for the control limits and central line. By default, *n* = 1.

WNEEDLES=*n*

specifies the width in pixels of needles connecting plotted points to the central line, as requested with the NEEDLES option. If you use the WNEEDLES= option, you do not need to specify the NEEDLES option. By default, *n* = 1.

WOVERLAY=(*value-list***)**

specifies the widths in pixels for the line segments connecting points on primary chart overlay plots. Widths in the WOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

WOVERLAY2=(*value-list***)**

specifies the widths in pixels for the line segments connecting points on secondary chart overlay plots. Widths in the WOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

WSTARCIRCLES=*n*

specifies the width in pixels of the outline of circles requested by the STARCIRCLES= option. See "Displaying Auxiliary Data with Stars" on page 2090. By default, *n* = 1.

WSTARS=*n*

specifies the width in pixels of the outline of stars requested by the STARVERTICES= option. See "Displaying Auxiliary Data with Stars" on page 2090. By default, *n* = 1.

WTESTS=*n*

WTEST=*n*

specifies the width in pixels of the line segments that connect patterns of points for which a test for special causes (requested with the TESTS= or TESTS2= option) is positive. By default, *n* = 1.

WTREND=*n*

specifies the width in pixels of the line segments that connect points on trend charts requested with the TRENDVAR= option. By default, *n* = 1. The WTREND= option is available in the BOXCHART, MCHART, and XCHART statements.

Options for Legacy Line Printer Charts

CLIPCHAR='*character***'**

specifies a plot character that identifies clipped points, as requested with the CLIPFACTOR= option. Specifying the CLIPCHAR= option is recommended when the CLIPFACTOR= option is used. The default character is an asterisk (*).

CONNECTCHAR='*character***'**

CCHAR='*character***'**

specifies the character used to form line segments that connect points on a chart. The default character is a plus (+) sign.

HREFCHAR='*character***'**

specifies the character used for reference lines requested by the HREF= and HREF2= options on line printer charts. The default is the vertical bar (|).

SYMBOLCHARS='*character-list***'**

specifies a list of characters used to mark the points plotted on line printer charts when a *symbol-variable* is used. See "Displaying Stratification in Levels of a Classification Variable" on page 2073.

Each character is associated with a level (unique value) of the *symbol-variable* and is used to mark points associated with that value. For example, consider the following statements:

```
proc shewhart;
   xrchart Gap*Shift=Machine / symbolchars='12345';
run;
```

Here the *symbol-variable* is Machine. The $\bar{X}$ and *R* charts use a '1' to mark points associated with the first unique value of Machine, a '2' to mark points associated with the second unique value of Machine, and so on.

If the number of levels of the *symbol-variable* exceeds the number of *characters*, the last character listed is used for points associated with the additional values. Thus, in the preceding example, if there are six levels of Machine, points with the fifth and six values are indicated by '5'.

The default *character-list* is ABCDEFGHIJKLMNOPQRSTUVWXYZ*. Thus, the procedure uses 'A' for the first unique value of the *symbol-variable*, 'B' for the second unique value, and so on. An asterisk is used for points associated with the 27th and subsequent levels when the *symbol-variable* has more than 26 levels.

TESTCHAR='*character***'**

specifies the character for the line segments that connect any sequence of points for which a test for special causes (requested with the TESTS= or TESTS2= option) is positive. The default *character* is the number of the test (with values 1 to 8).

VREFCHAR='*character*'
specifies the character used for reference lines requested by the VREF= and VREF2= options on line printer charts. The default is the hyphen (-).

ZONECHAR='*character*'
specifies the character used to form the zone lines requested by the ZONES option. See the entry for the TESTS= option for a description of the zones. You do not need to specify the ZONES option if you specify the ZONECHAR= option. By default, the line between Zone A and Zone B uses the character 'B', and the line between Zone B and Zone C uses the character 'C'. Related options are TESTS=, TESTS2=, ZONES, and ZONES2.

Graphical Enhancements: SHEWHART Procedure

Overview: Graphical Enhancements

This section provides details on the following topics:

- displaying process data stratified into levels using a *symbol-variable*
- displaying process data stratified into blocks using *block-variables*
- displaying process data stratified into time phases using the READPHASES= option
- displaying multiple sets of control limits using the READPHASES= and READINDEXES= options
- displaying multivariate process data using star charts
- displaying trends in process data
- clipping extreme points to create more readable charts
- labeling axes
- selecting subgroups for computation and display

The options described in this section can be specified in all the chart statements available in the SHEWHART procedure.

Displaying Stratified Process Data

If the data for a Shewhart chart can be classified by factors relevant to the process (for instance, machines or operators), displaying the classification on the chart can facilitate the identification of special or common causes of variation that are related to the factors. Kume (1985) refers to this type of classification as "stratification" and describes various ways to create stratified control charts.

There are important differences between stratification and subgrouping. The data must always be classified into subgroups before a control chart can be produced. Subgrouping affects how control limits are computed from the data as well as the outcome of tests for special causes (see "Tests for Special Causes: SHEWHART Procedure" on page 2119). The values of the *subgroup-variable* specified in the chart statement classify the data into subgroups. In contrast, stratification is optional and involves classification variables other than the *subgroup-variable*. Displaying stratification influences how the chart is interpreted, but it does not affect control limits or tests for special causes.

This section describes three types of variables that you can specify to create stratified control charts.

- A *symbol-variable* stratifies data into levels of a classification variable.
- The *block-variables* stratify data into blocks of consecutive observations.
- A _PHASE_ variable stratifies data into *time phases*.

You can specify any combination of these three variables. You should be careful, however, because it is possible to generate confusing charts by overusing these methods.

The data for the examples in this section consist of diameter measurements for a part produced on one of three different machines. Three subgroups, each consisting of six parts, are sampled each day, corresponding to three shifts worked each day. The data are provided in the data set Parts, which is created by the following statements:

```
data Parts;
   length Machine $ 4;
   input Sample Machine $ Day Shift DiamX DiamS;
   DiamN=6;
   datalines;
 1  A386  01  1  4.32  0.39
 2  A386  01  2  4.49  0.35
 3  A386  01  3  4.44  0.44
 4  A386  02  1  4.45  0.17
 5  A386  02  2  4.21  0.53
 6  A386  02  3  4.56  0.26
 7  A386  03  1  4.63  0.39
 8  A386  03  2  4.38  0.47
 9  A386  03  3  4.47  0.40
10  A455  04  1  4.42  0.37
11  A455  04  2  4.45  0.32
12  A455  04  3  4.62  0.36
13  A455  05  1  4.33  0.31
14  A455  05  2  4.29  0.33
15  A455  05  3  4.17  0.25
16  C334  08  1  4.15  0.28
17  C334  08  2  4.21  0.33
18  C334  08  3  4.16  0.19
19  C334  09  1  4.14  0.13
20  C334  09  2  4.11  0.19
21  C334  09  3  4.10  0.27
22  C334  10  1  3.99  0.14
23  C334  10  2  4.24  0.16
24  C334  10  3  4.23  0.14
```

```
25  A386  11  1  4.27  0.28
26  A386  11  2  4.70  0.45
27  A386  11  3  4.51  0.45
28  A386  12  1  4.34  0.16
29  A386  12  2  4.38  0.29
30  A386  12  3  4.28  0.24
31  A386  15  1  4.47  0.26
32  A386  15  2  4.31  0.46
33  A386  15  3  4.52  0.33
;
```

Displaying Stratification in Levels of a Classification Variable

NOTE: See *Stratifying Data with a Classification Variable* in the SAS/QC Sample Library.

To display process data stratified into levels of a classification variable, specify the name of this variable after an equal sign (=) immediately following the *subgroup-variable* in the chart statement. The classification variable, referred to as the *symbol-variable*, must be a variable in the input data set (a DATA=, HISTORY=, or TABLE= data set). The subgroup summary statistics are classified into groups according to the levels of the *symbol-variable* and are identified on the chart with unique plotting symbols.

When you produce traditional graphics output, you can specify the symbols with SYMBOL statements. It is recommended that you place the SYMBOL statements before the PROC SHEWHART statement. If you omit the SYMBOL statements, the procedure uses the default symbol (+) for all levels of the *symbol-variable* but plots the points for each level in a distinct color. The following example illustrates the use of a *symbol-variable* to stratify the points on an $\bar{X}$ chart according to the machine that produced the parts in each subgroup:

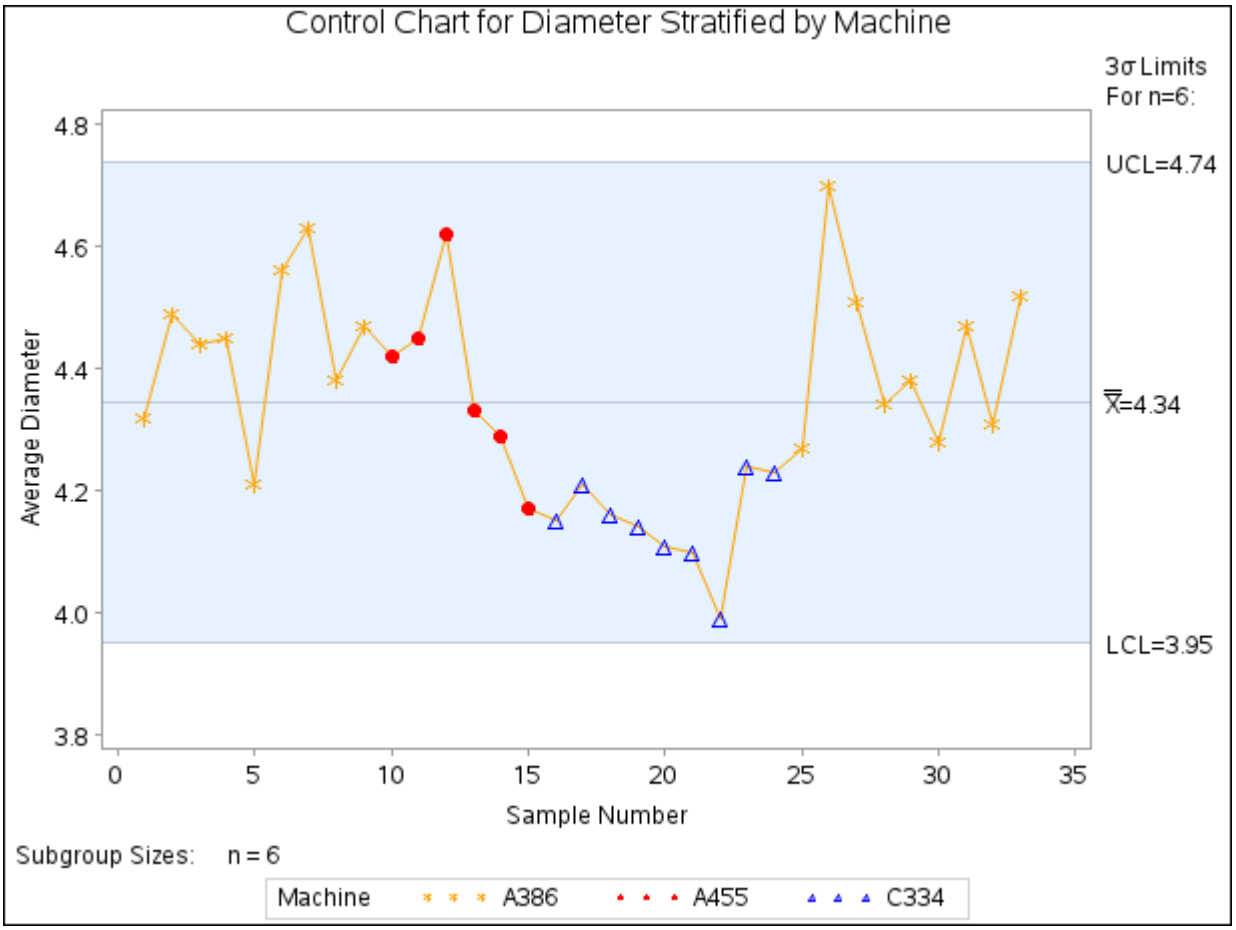
```
ods graphics off;
symbol1 c=orange value=star      h=3.0 pct;
symbol2 c=red    value=dot       h=3.0 pct;
symbol3 c=blue   value=triangle  h=3.0 pct;
title 'Control Chart for Diameter Stratified by Machine';
proc shewhart history=Parts;
   xchart Diam*Sample=Machine / stddeviations
                                symbollegend = legend1;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter' ;
   legend1 frame label=('Machine');
run;
```

The symbols are specified with the SYMBOL1, SYMBOL2, and SYMBOL3 statements. The SYMBOLLEGEND= option requests a customized legend for the symbols. For more information about the LEGEND and SYMBOL statements, refer to *SAS/GRAPH: Reference*. The $\bar{X}$ chart, shown in Figure 19.143, reveals an effect due to Machine. In particular, Machine C334 is associated with a run of parts whose diameters are systematically below average, suggesting that this machine might require adjustment.

For line printer charts, you can use the SYMBOLCHARS= option to specify the characters that identify the stratification of the points. For details, see the entry for the SYMBOLCHARS= option in "Dictionary of Options: SHEWHART Procedure" on page 1993.

In this example, Machine A386 is associated with two different blocks of observations that are identified with a common symbol. However, a *symbol-variable* is particularly useful for situations where the stratification is not necessarily chronological or associated with blocks of consecutive groups of observations.

Figure 19.143 Control Chart Stratified into Levels Using Symbols



Displaying Stratification in Blocks of Observations

NOTE: See *Using Block Variables to Stratify Data* in the SAS/QC Sample Library.

To display process data stratified into blocks of consecutive observations, specify one or more *block-variables* in parentheses after the *subgroup-variable* in the chart statement. The procedure displays a legend identifying blocks of consecutive observations with identical values of the *block-variables*. The legend displays one track of values for each *block-variable*. The values are the formatted values of the *block-variable*. For example, Figure 19.144 displays a legend with a single track for Machine, while Figure 19.145 displays a legend with two tracks corresponding to Machine and Day. You can label the tracks themselves by using the LABEL statement to associate labels with the corresponding *block-variables*; see Figure 19.146 for an illustration.

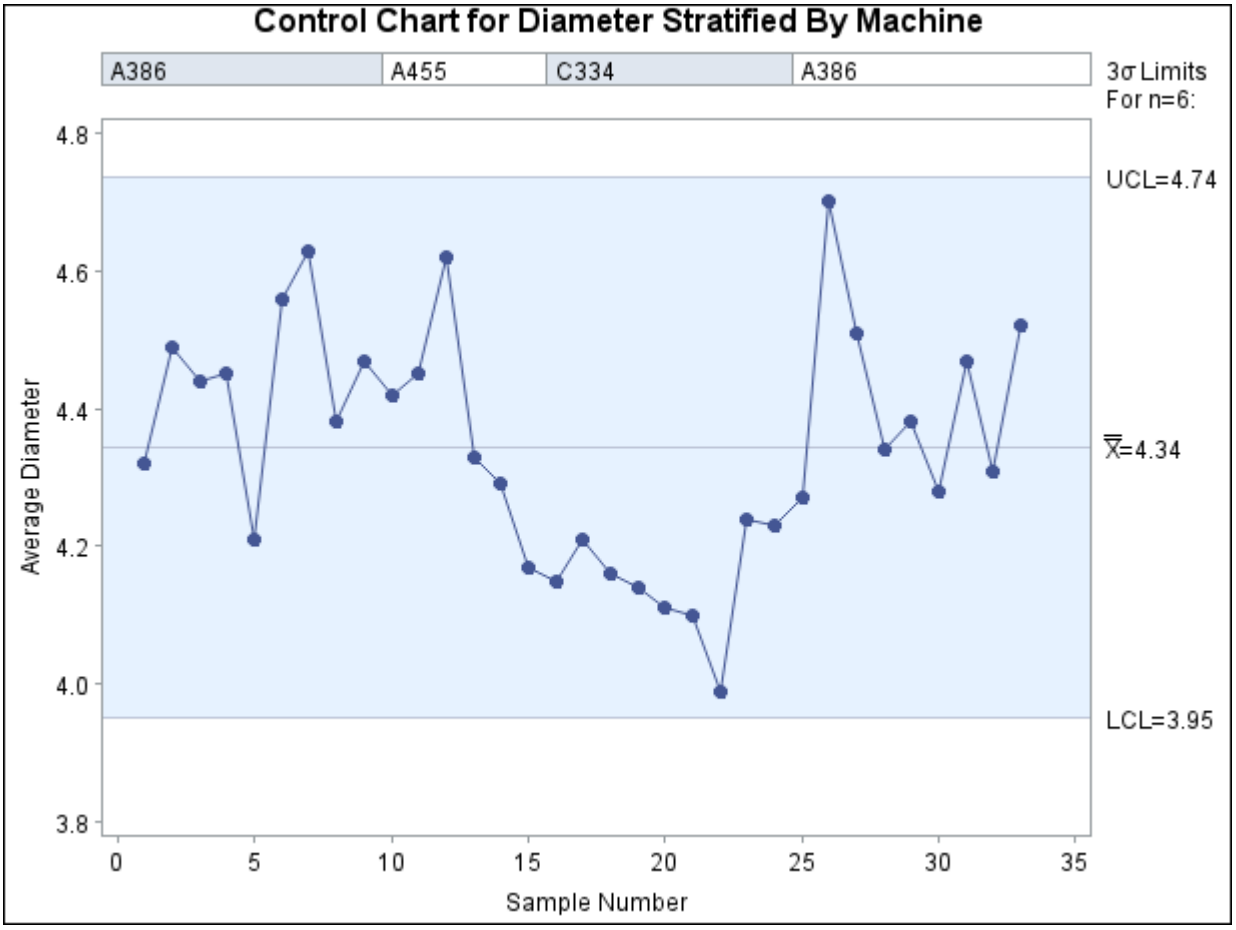
By default, the legend is placed above the chart as in Figure 19.144. You can control the position of the legend with the BLOCKPOS= option and the position of the legend labels with the BLOCKLABELPOS= option. See the entries in "Dictionary of Options: SHEWHART Procedure" on page 1993 as well as the following examples.

The *block-variables* must be variables in the input data set (a DATA=, HISTORY=, or TABLE= data set). If the input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the values of a *block-variable* must be the same for all observations with the same value of the *subgroup-variable*. In other words, subgroups must be nested within groups determined by *block-variables*. The following statements create an $\bar{X}$ chart for the data in Parts stratified by the *block-variable* Machine. The chart is shown in Figure 19.144.

```
symbol v=dot h=3.0 pct;
title 'Control Chart for Diameter Stratified By Machine';
proc shewhart history=Parts;
   xchart Diam*Sample (Machine) / stddeviations
                                  nolegend ;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter' ;
run;
```

The unique consecutive values of Machine ('A386', 'A455', 'C334', and 'A386') are displayed in a track above the chart, and they indicate the same relationship between part diameter and machine as the previous example. Note that the track is not labeled (as in Figure 19.146), because no label is associated with Machine. A LABEL statement is used to provide labels for the axes.

Figure 19.144 Stratified Control Chart Using a Single Block Variable

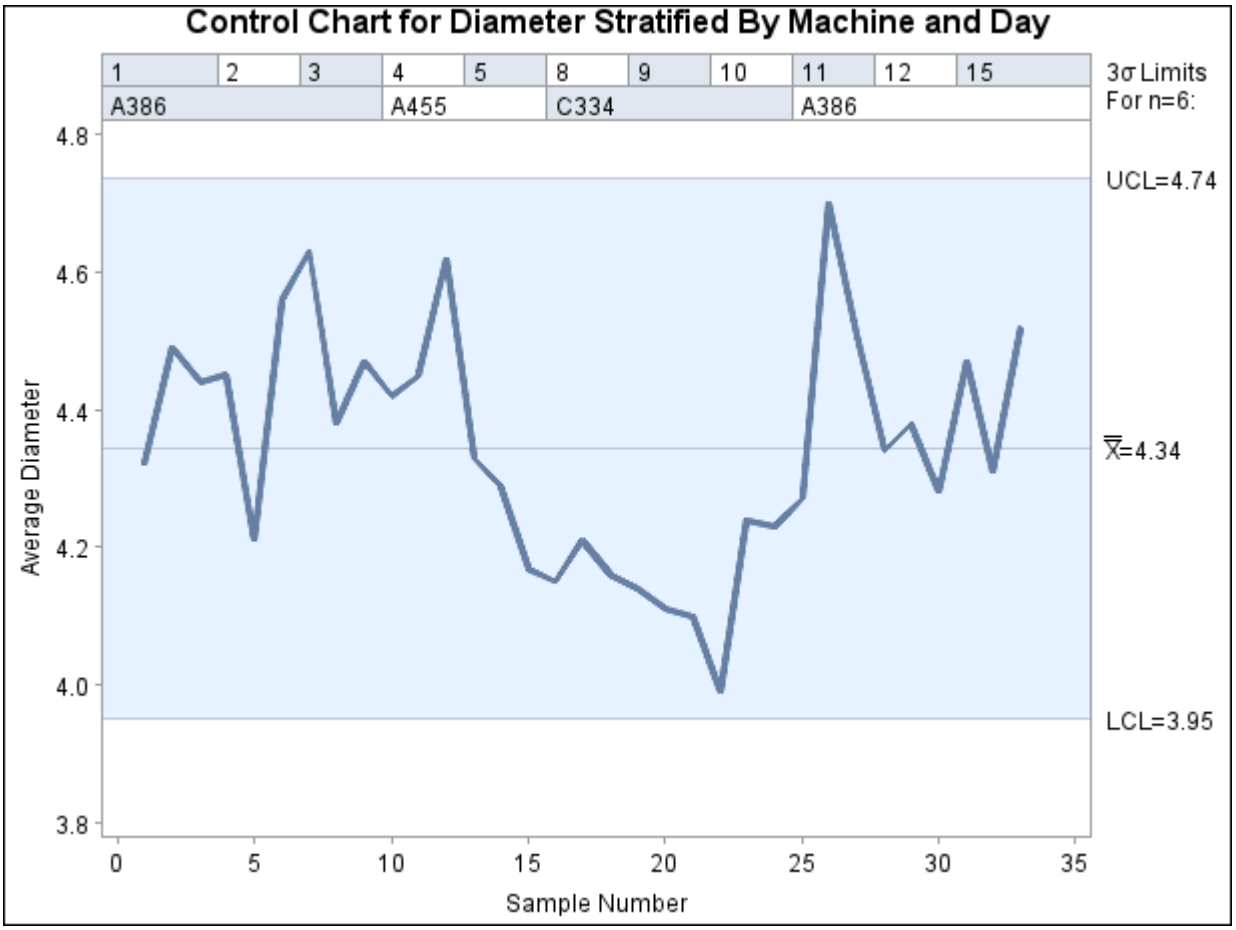


Multiple block variables. You can use multiple *block-variables* to study more than one classification factor with the same chart. The following statements create an $\bar{X}$ chart for the data in Parts, with Machine and Day as *block-variables*:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=Parts;
   xchart Diam*Sample (Machine Day) / stddeviations
                                      nolegend
                                      blockpos = 2;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter' ;
run;
```

The chart is displayed in Figure 19.145. Specifying BLOCKPOS=2 displays the *block-variable* legend immediately above the chart, without the gap shown in Figure 19.144. The NOLEGEND option suppresses the sample size legend that appears in the lower left of Figure 19.144.

Figure 19.145 Stratified Control Chart Using Multiple Block Variables

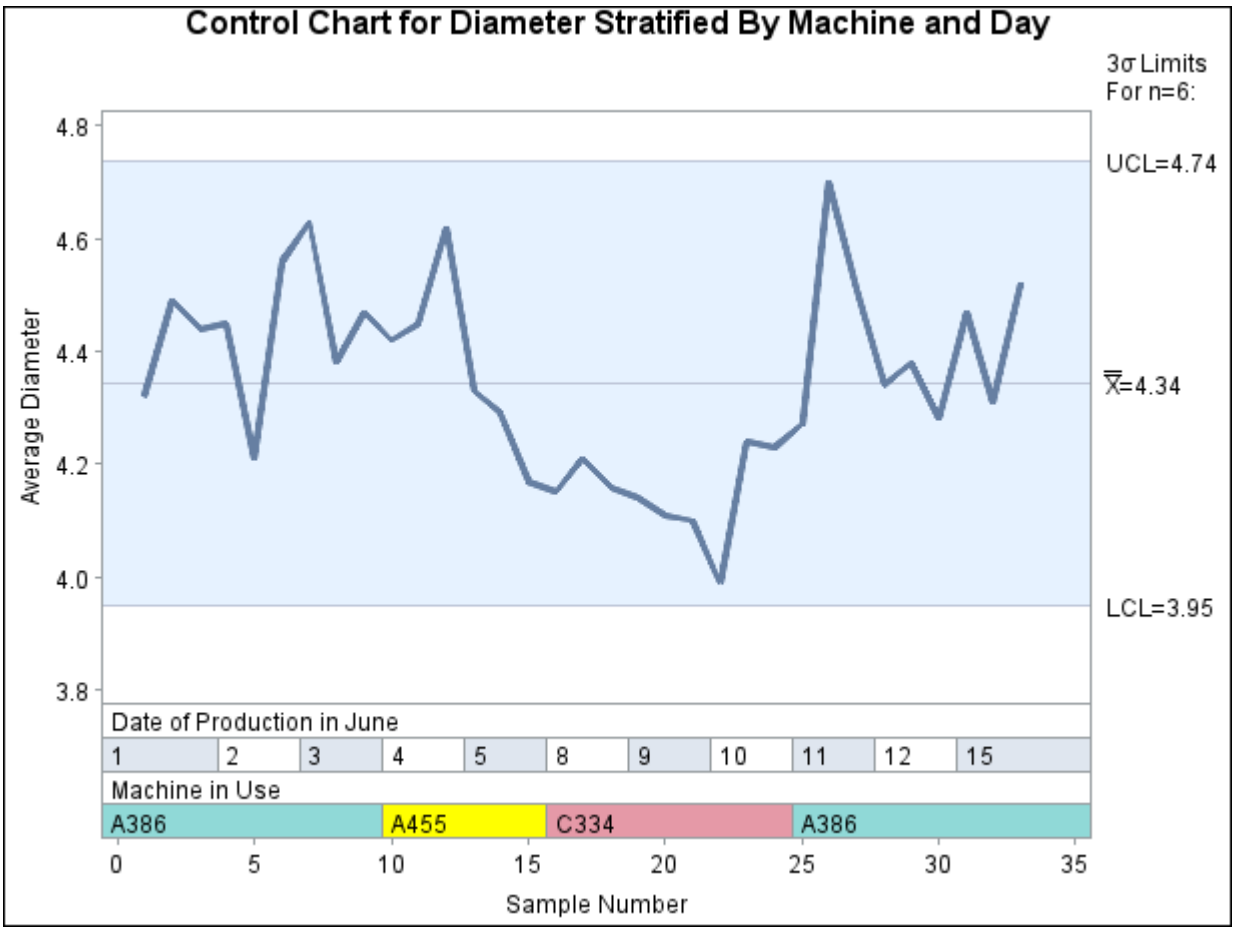


Color fills for legend. You can use the CBLOCKVAR= option to fill the legend track sections with colors corresponding to the values of the *block-variables*. Provide the colors as values of variables specified with the CBLOCKVAR= option. The procedure matches the color variables with the *block-variables* in the order specified. Each section is filled with the color for the first observation in the block. For example, the

following statements produce an $\bar{X}$ chart using a color variable named CMachine to fill the legend for the *block-variable* Machine:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=Parts2;
   xchart Diam*Sample (Machine Day) / stddeviations
                                      nolegend
                                      blockpos  = 3
                                      cblockvar = CMachine;
   label Sample  = 'Sample Number'
         DiamX   = 'Average Diameter'
         Day     = 'Date of Production in June'
         Machine = 'Machine in Use';
run;
```

Figure 19.146 Color Fill for *Block-Variable* Legend



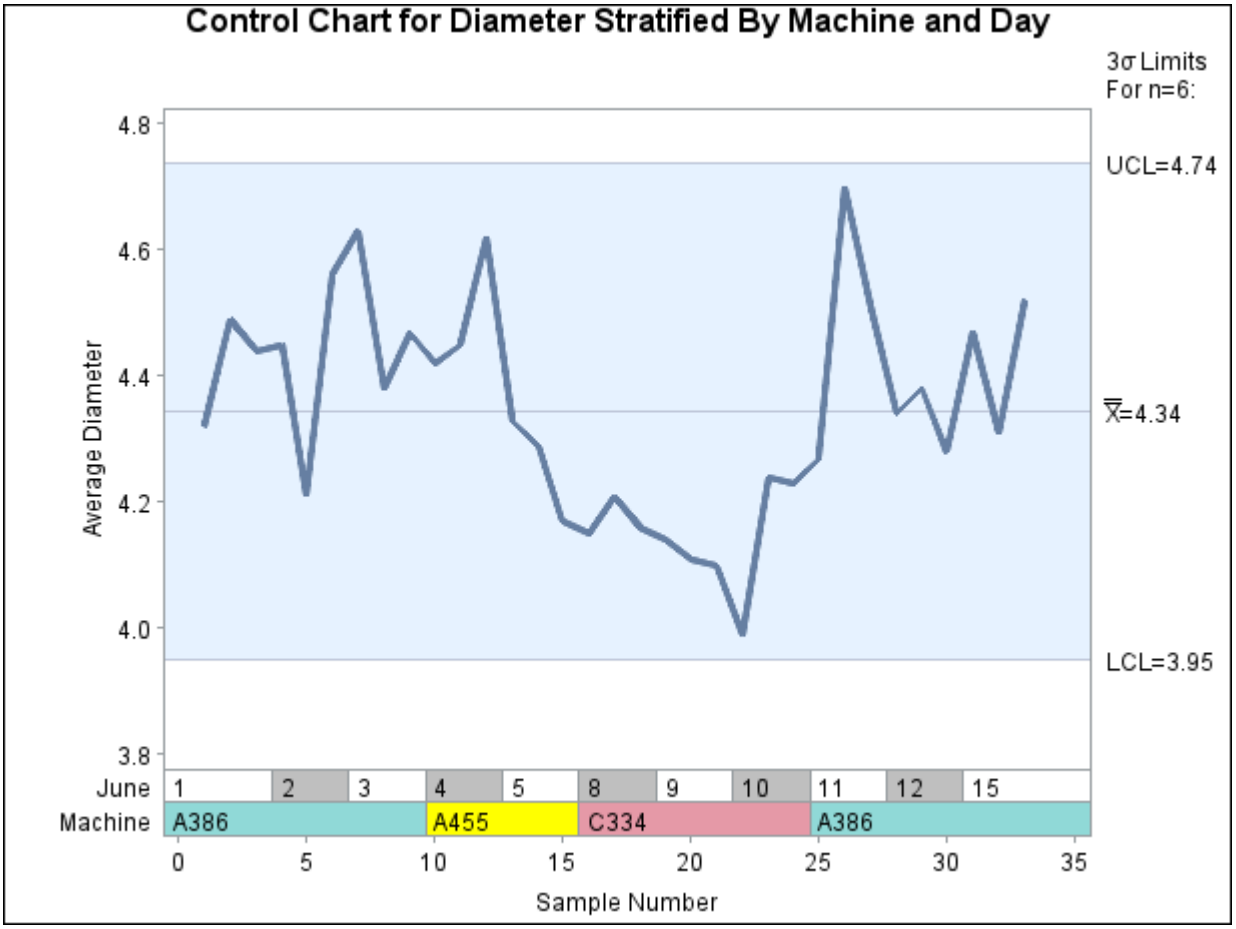
The sections for Machine A386, Machine A455, and Machine C334 are filled with the colors specified as values of CMachine. The legend track for Day is filled with the default alternating colors from the ODS style, because a second color variable was not specified with the CBLOCKVAR= option. Specifying BLOCKPOS=3 positions the legend at the bottom of the chart and facilitates comparison with the subgroup axis. The LABEL statement is used to label the tracks with the labels associated with the *block-variables*.

The following statements produce an $\bar{X}$ chart in which both legend tracks are filled:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=Parts3;
   xchart Diam*Sample (Machine Day) /
      stddeviations
      nolegend
      ltmargin      = 5
      blockpos      = 3
      blocklabelpos = left
      cblockvar     = (CMachine CDay);
   label Sample     = 'Sample Number'
         DiamX      = 'Average Diameter'
         Day        = 'June'
         Machine    = 'Machine';
run;
```

The chart is displayed in Figure 19.147. The color values of CMachine are used to fill the track for Machine, and the color values of CDay are used to fill the track for Day. Specifying BLOCKLABELPOS=LEFT displays the block variable labels to the left of the block legend. The LTMARGIN= option provides extra space in the left margin to accommodate the label *Machine*.

Figure 19.147 Stratified Control Chart Using Multiple Block Variables



Displaying Stratification in Phases

NOTE: See *Displaying Stratification in Phases* in the SAS/QC Sample Library.

The preceding section describes the use of *block-variables* to display blocks of consecutive observations that correspond to changes in factors such as machines, shifts, and raw materials. This section describes the use of a *_PHASE_ variable* to display phases of consecutive observations (as in Figure 19.148). Although the terms *block* and *phase* have similar meanings, there are differences in the two methods:

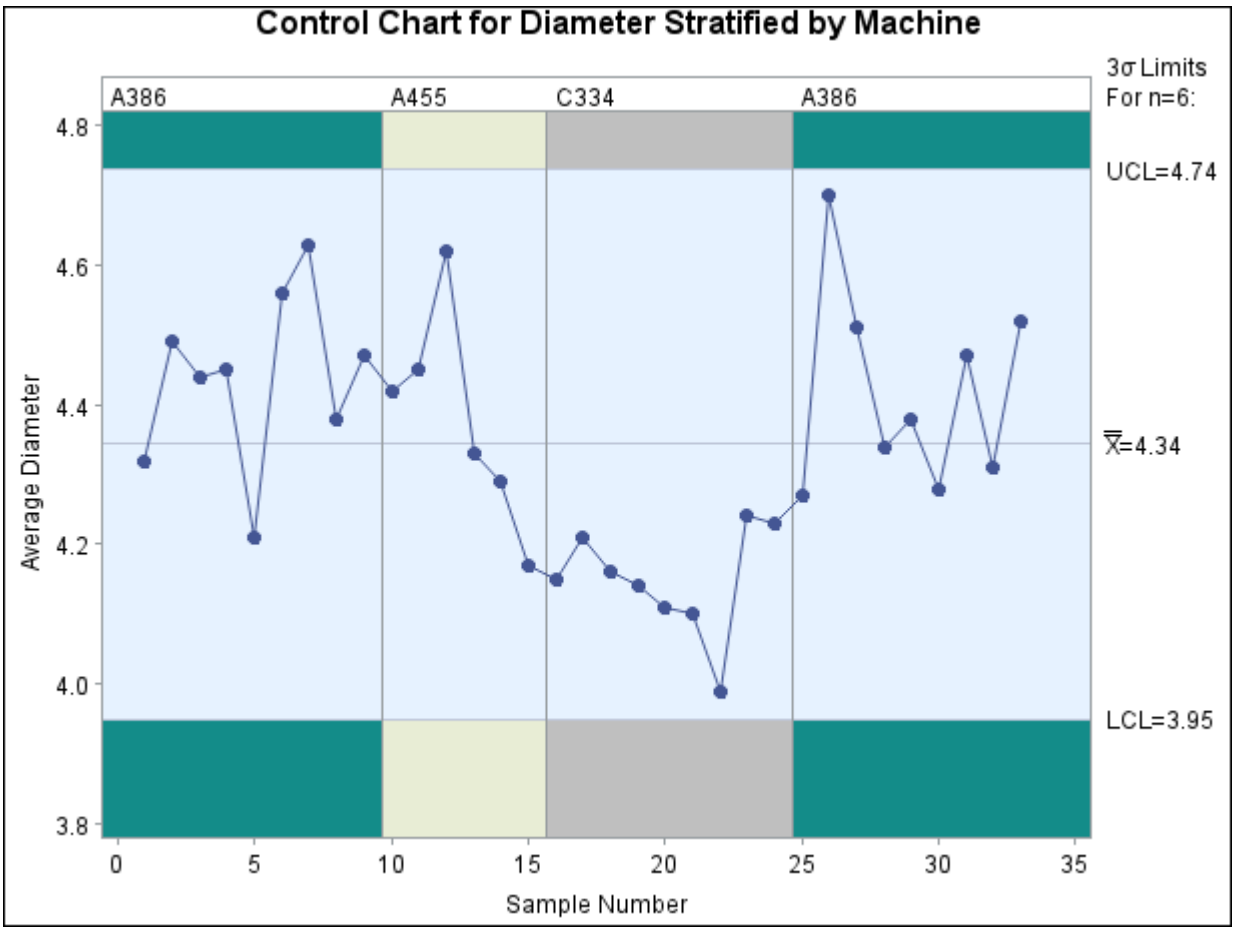
- You can provide only one Variable_PHASE_ variable, whereas you can specify multiple *block-variables*.
- You can display distinct control limits for each phase (see "Displaying Multiple Sets of Control Limits" on page 2081) but not for each block.
- Different sets of graphical options are available for identifying blocks and phases.

To display phases, your input data set must include a character variable named _PHASE_ of length 48 or less, and you must specify the READPHASES= option in the chart statement. (If your data set does not include a variable named _PHASE_, you can temporarily rename another character variable to _PHASE_, as illustrated by the following statements.) The procedure classifies the data into phases (groups of consecutive observations with the same value of _PHASE_) and reads only those observations whose _PHASE_ value matches one of the values specified with the READPHASES= option.

You can identify and highlight the phases with various options, as illustrated by the following statements, which produce the chart shown in Figure 19.148. The PHASELEGEND option displays a legend with the _PHASE_ values, and the CPHASELEG= option specifies the color of the legend text. The PHASEREF option delineates the phases with vertical reference lines. The CFRAME= option fills the framed areas for the phases with different colors.

```
ods graphics off;
symbol v=dot h=3.0 pct;
title 'Control Chart for Diameter Stratified by Machine';
proc shewhart history=Parts(rename=(Machine=_phase_));
   xchart Diam*Sample /
      stddeviations
      readphases = ('A386' 'A455' 'C334' 'A386')
      cframe     = ( vibg   ywh    ligr   vibg )
      phaselegend
      cphaseleg  = black
      phaseref
      nolegend;
   label  Sample = 'Sample Number'
          DiamX  = 'Average Diameter';
run;
```

Figure 19.148 Control Chart Stratified by Phases



Note that the data set Parts does not contain a variable named _PHASE_, so the variable Machine is renamed as _PHASE_ for the duration of the procedure step.

The observations read from Parts are those whose value of Machine matches one of the values listed with the READPHASES= option in that order. Here, the value 'A386' is listed twice; consequently, both groups of observations for which Machine equals 'A386' are read.

In this example, the input data set contains a single observation for each subgroup. If your input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the value of _PHASE_ must be the same for all observations with the same value of the *subgroup-variable*. Thus, in general, subgroups must be nested within phases.

Recall that the horizontal axis scale is determined by the *subgroup-variable* (see "Subgroup Variables" on page 1970). If your *subgroup-variable* is numeric, this scale is continuous; consequently, you should select phases that are reasonably contiguous in order to avoid large empty gaps in your chart. For instance, if you were to specify

```
readphases = ('A386' 'A455' 'A386')
```

in the preceding XCHART statement, there would be a gap between the 15th and 25th points (these points would be connected unless you specified the PHASEBREAK option). You can avoid gaps by specifying a character *subgroup-variable*[11] for which a discrete horizontal axis scale will be displayed.

[11] You can use the PUT function in a DATA step to create a character *subgroup-variable* from a numeric *subgroup-variable*.

Note that the values listed in the READPHASES= option must be listed in the same order as they occur in the input data set. Thus, in order to display all the observations in the data set Parts, 'A386' must be listed as both the first and last value. An alternative method for selecting all the phases from your input data is to specify READPHASES=ALL, as described in the next section.

The control limits shown in Figure 19.148 are computed from the data and are, therefore, the same across all phases. More generally, you can display a distinct set of control limits for each phase. To do so, you must provide the control limits in a LIMITS= data set and specify the READINDEXES= option in addition to the READPHASES= option, as described in the next section.

Displaying Multiple Sets of Control Limits

NOTE: See *Displaying Multiple Sets of Control Limits* in the SAS/QC Sample Library.

This section describes the use of the READPHASES= and READINDEXES= options for creating Shewhart charts that display distinct sets of control limits for multiple phases of observations. The term *phase* refers to a group of consecutive observations in the input data set. For example, the phases might correspond to the time periods during which a new process was brought into production and then put through successive changes.

To display phases, your input data must include a character variable named _PHASE_, whose length cannot exceed 48. (If your data set does not include a variable named _PHASE_, you can temporarily rename another character variable to _PHASE_, as illustrated in the statements in the section "Displaying Stratification in Phases" on page 2079.) Each phase consists of a group of consecutive observations with the same value of _PHASE_.

To display distinct sets of predetermined control limits for the phases, you must provide the limits in a LIMITS= data set. This data set must include a character variable named _INDEX_, whose length cannot exceed 48. This variable identifies the sets of control limits (observations) in the LIMITS= data set that are to be associated with the phases. This data set must also include a number of other variables with reserved names that begin and end with an underscore. The particular structure of a LIMITS= data set depends on the chart statement that you are using; for details, see the sections titled "LIMITS= Data Set" in the sections for the various chart statements. In addition to specifying a LIMITS= data set, you must also specify the READINDEXES= and READPHASES= options in the chart statement.

NOTE: To display a *single* set of predetermined control limits with multiple phases, simply specify a LIMITS= data set in the procedure statement. If you are using SAS 6.09 or an earlier release, you must also specify the READLIMITS option. The control limits are read from the first observation in the LIMITS= data for which the variable _VAR_ is equal to the name of the *process* and the variable _SUBGRP_ is equal to the name of the *subgroup-variable*. For an example, see "Reading Preestablished Control Limits" on page 1892.

This section describes the combinations of the READINDEXES= and READPHASES= options that you can specify. The examples that follow use the HISTORY= data set Flange listed in Figure 19.149 and the LIMITS= data set Flangelim listed in Figure 19.150. The data in Flange consist of means and ranges of flange width measurements for subgroups of size five. The observations are grouped into three phases determined by the _PHASE_ values 'Production', 'Change 1', and 'Change 2'. Three sets of control limits are provided in Flangelim, corresponding to the _INDEX_ values 'Start', 'Production', and 'Change 1'.

Figure 19.149 Listing of the HISTORY= Data Set Flange

Obs	_phase_	Day	Sample	FlwidthX	FlwidthR	FlwidthN
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
4	Production	11FEB90	9	0.95509	0.08378	5
5	Production	12FEB90	10	1.00348	0.09993	5
6	Production	15FEB90	11	1.02566	0.06766	5
7	Production	16FEB90	12	0.97053	0.07608	5
8	Production	17FEB90	13	0.94713	0.10170	5
9	Production	18FEB90	14	1.00377	0.04875	5
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
13	Change 1	24FEB90	18	1.02235	0.10541	5
14	Change 1	25FEB90	19	0.99950	0.11476	5
15	Change 1	26FEB90	20	0.99271	0.05395	5
16	Change 1	01MAR90	21	0.98695	0.03833	5
17	Change 1	02MAR90	22	1.00969	0.06183	5
18	Change 1	03MAR90	23	0.98791	0.05836	5
19	Change 1	04MAR90	24	1.00170	0.05243	5
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
23	Change 2	10MAR90	28	1.01463	0.05558	5
24	Change 2	11MAR90	29	0.99812	0.03648	5
25	Change 2	12MAR90	30	1.00047	0.04309	5
26	Change 2	15MAR90	31	0.99714	0.03689	5
27	Change 2	16MAR90	32	0.98642	0.04809	5
28	Change 2	17MAR90	33	0.98891	0.07777	5
29	Change 2	18MAR90	34	1.00087	0.06409	5
30	Change 2	19MAR90	35	1.00863	0.02649	5

Figure 19.150 Listing of the LIMITS= Data Set Flangelim

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_	_lclx_	_mean_
1	Change 1	Flwidth	Sample	ESTIMATE	5	.0026998	3	0.96167	0.99924
2	Production	Flwidth	Sample	ESTIMATE	5	.0026998	3	0.93792	0.98827
3	Start	Flwidth	Sample	ESTIMATE	5	.0026998	3	0.87088	0.96803

Obs	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	1.03680	0	0.06513	0.13771	0.028000
2	1.03862	0	0.08729	0.18458	0.037530
3	1.06517	0	0.16842	0.35612	0.072409

For each of the READINDEXES= and READPHASES= options, you can specify a single value, a list of values, or the keyword ALL. You can also leave these options unspecified. Thus, there are 16 possible combinations of specifications for the two options, as explained by the following table and notes. The two most commonly encountered combinations are

- reading a single set of limits for one or more phases (see Case 1)
- reading a set of limits matched with a set of phases (see Case 4)

	READPHASES=			
READINDEXES=	**Single *Value***	**Multiple *Values***	***Keyword* ALL**	**Not Specified**
Single *Value*	See Case 1	See Case 1	See Case 2	See Case 3
Multiple *Values*	See Case 9	See Case 4	See Case 2	See Case 2
Keyword ALL	See Case 5	See Case 5	See Case 6	See Case 6
Not Specified	See Case 7	See Case 7	See Case 8	See Case 8

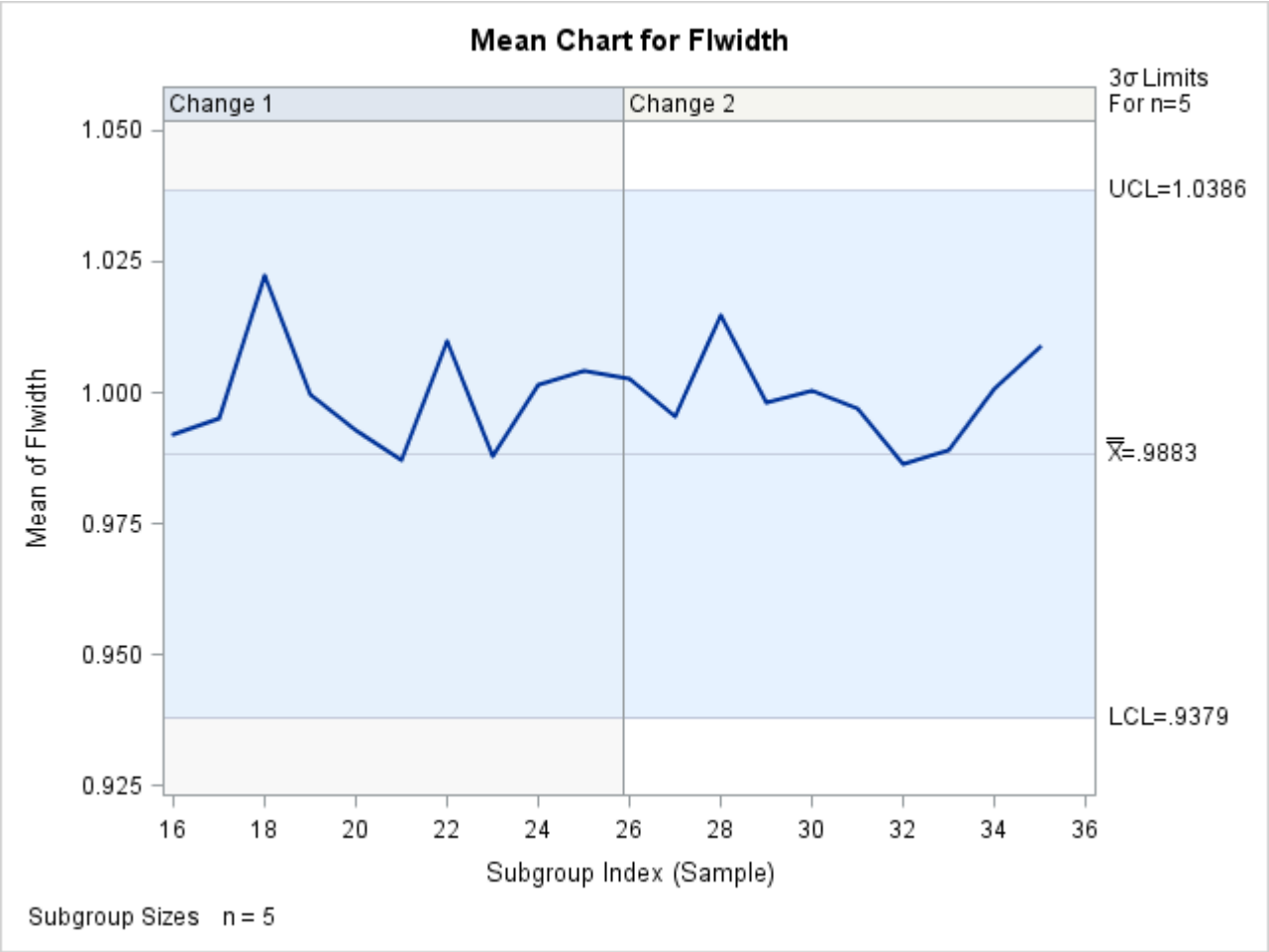
Case 1. READPHASES=*value*|*value-list* and READINDEXES=*value*

The only phases (groups of observations) read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits given by the first observation in the LIMITS= data set for which _INDEX_ is equal to the READINDEXES= *value*.

For example, the following statements create a chart for the phases 'Change 1' and 'Change 2', with control limits read from the second observation in Flangelim. The chart is displayed in Figure 19.151.

```
ods graphics on;
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphase = ('Change 1' 'Change 2')
      readindex = ('Production')
      phaseref
      phaselegend;
run;
```

Figure 19.151 A Single Set of Control Limits for Multiple Phases



Case 2. READPHASES=ALL and READINDEXES=*value*|*value-list* or READPHASES= is omitted and READINDEXES=*value-list*

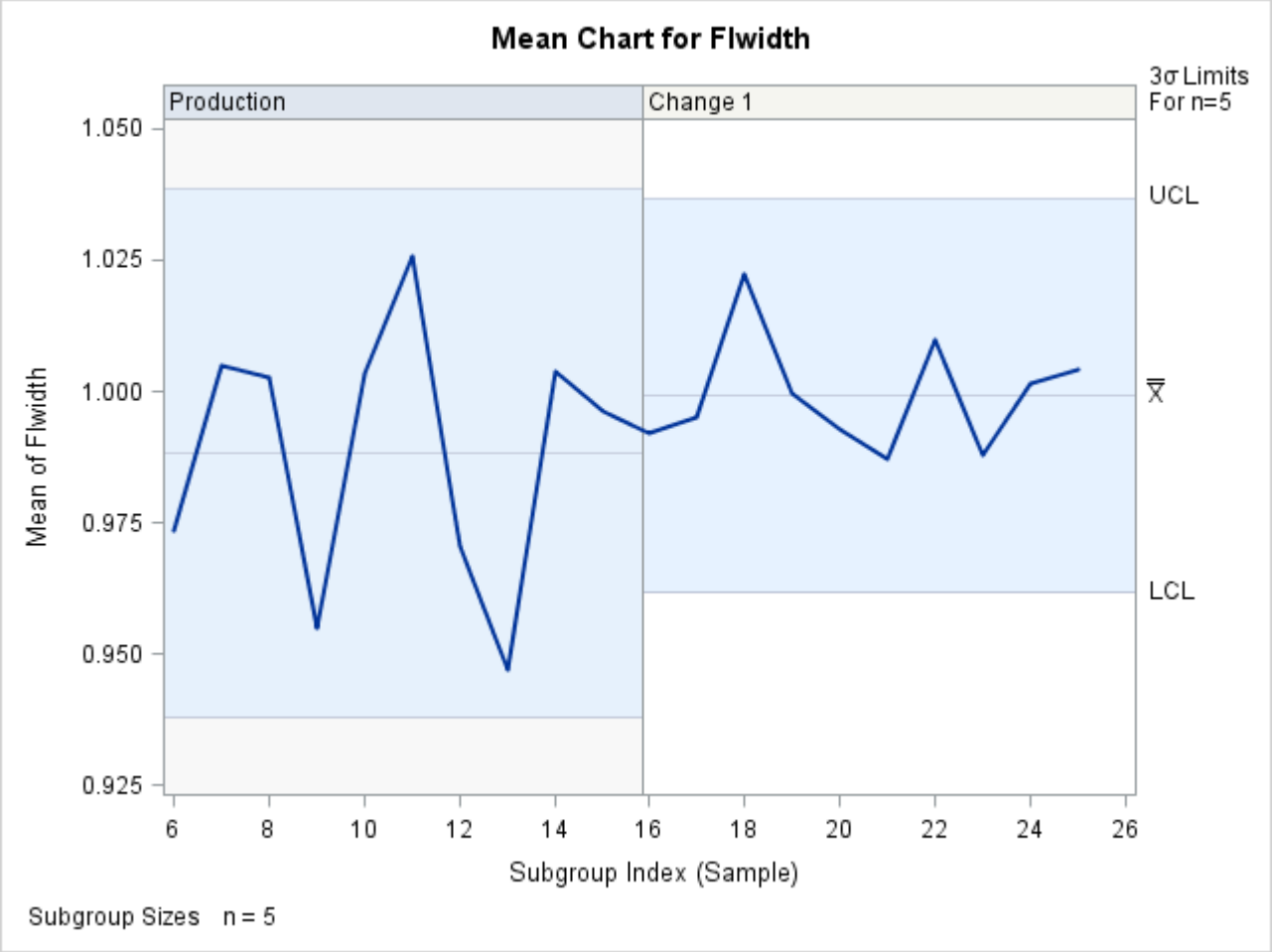
The only phases read are those for which _PHASE_ equals one of the *values* specified with the READINDEXES= option. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ is equal to the corresponding *value*.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with control limits read from the second and first observations in Flangelim, respectively. The chart is displayed in Figure 19.152.

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphase = all
      readindex = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

If you wish to specify a single set of control limits to use with all the phases, use the READINDEXES= option *without* the READPHASES= option (see Case 3).

Figure 19.152 READPHASES=ALL with a List of Values for READINDEXES=



Case 3. READPHASES= is omitted and READINDEXES=*value*

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data for which _INDEX_ equals the *value*.

Case 4. READPHASES=*value-list* and READINDEXES=*value-list*

The only phases read are those for which _PHASE_ equals one of the values specified with the READPHASES= option. The chart displays a different set of control limits for each phase, given by the first observation in the LIMITS= data set for which _INDEX_ equals the READINDEXES=*value*. Control limits are matched with phases in the order listed.

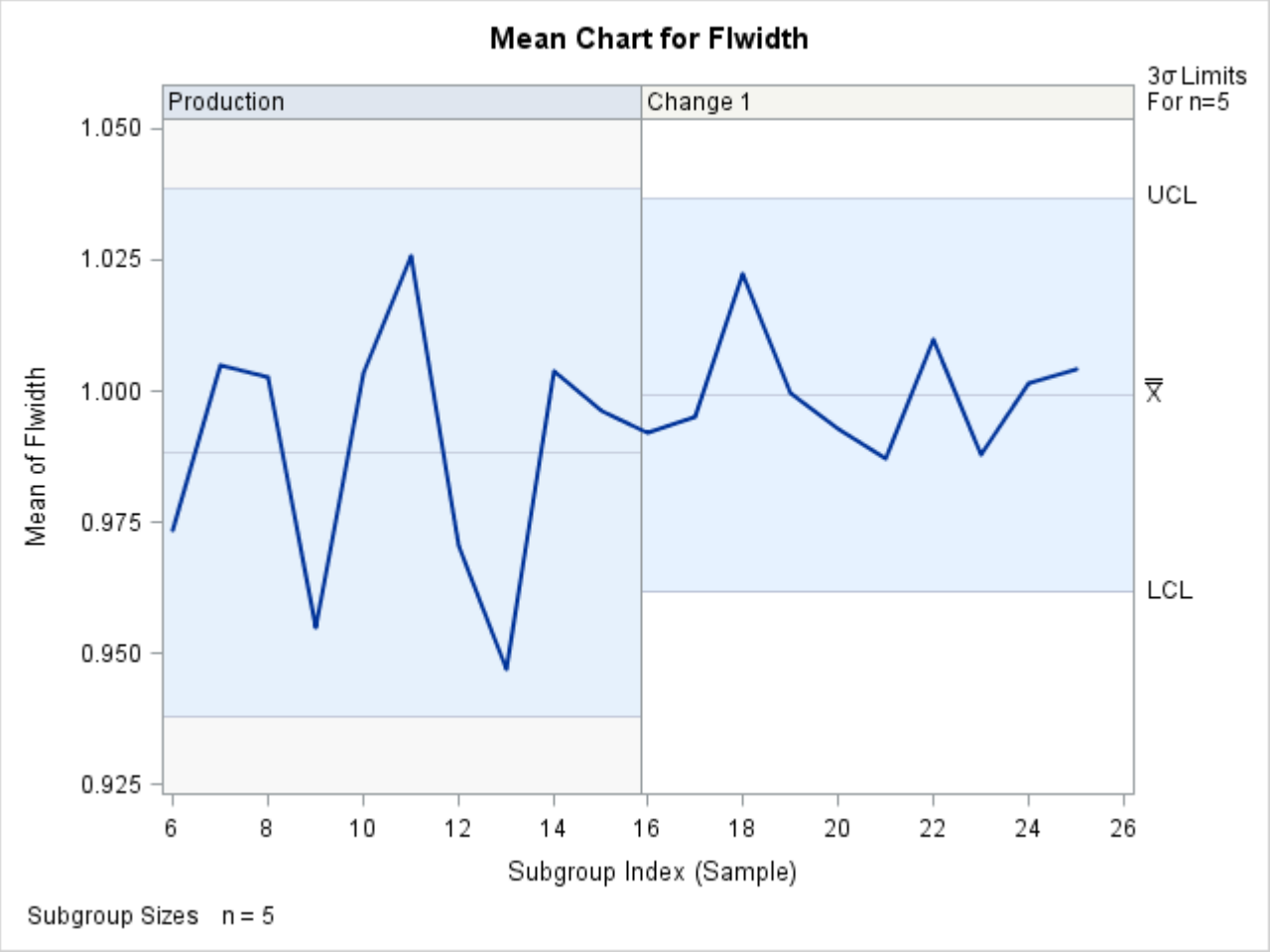
For example, the following statements create a chart for the phases 'Production' and 'Change 1' with control limits read from the first and second observations in Flangelim, respectively. The chart produced by these statements is identical to the chart in Figure 19.152.

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphases  = ('Production' 'Change 1')
      readindexes = ('Production' 'Change 1')
      phaseref
```

```
      phaselegend;
run;
```

The order of the READINDEX=*value-list* is critical. For instance, the previous statements with READINDEXES=('Change 1' 'Production') create the chart in Figure 19.153, in which the control limits are mismatched with the phases.

Figure 19.153 Multiple Phases with Mismatched Control Limits



Case 5. READPHASES=*value*|*value-list* and READINDEXES=ALL

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ equals the *value* corresponding to the phase.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with the control limits read from the second and first observations in Flangelim, respectively:

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphases  = ('Production' 'Change 1')
      readindexes = all
      phaseref
```

```
      phaselegend ;
run;
```

The chart is identical to the chart in Figure 19.152. In general, to read a set of phases with identically labeled control limits, you can specify the phases with either the READPHASES= or READINDEXES= option, and you can specify the keyword ALL with the other option.

Case 6. READPHASES=ALL and READINDEXES=ALL or READPHASES= is omitted and READINDEXES=ALL

All phases are read for which _PHASE_ is a value of _INDEX_ in the LIMITS= data set. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ equals the value of _PHASE_.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with control limits read from the second and first observations in Flangelim, respectively. These two phases are read because they are the only phases in Flange with matching _INDEX_ values in Flangelim. The chart is identical to that in Figure 19.152.

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphase = all
      readindex = all
      phaseref
      phaselegend ;
run;
```

Note that an identical chart would be produced if you were to omit the READPHASES= option.

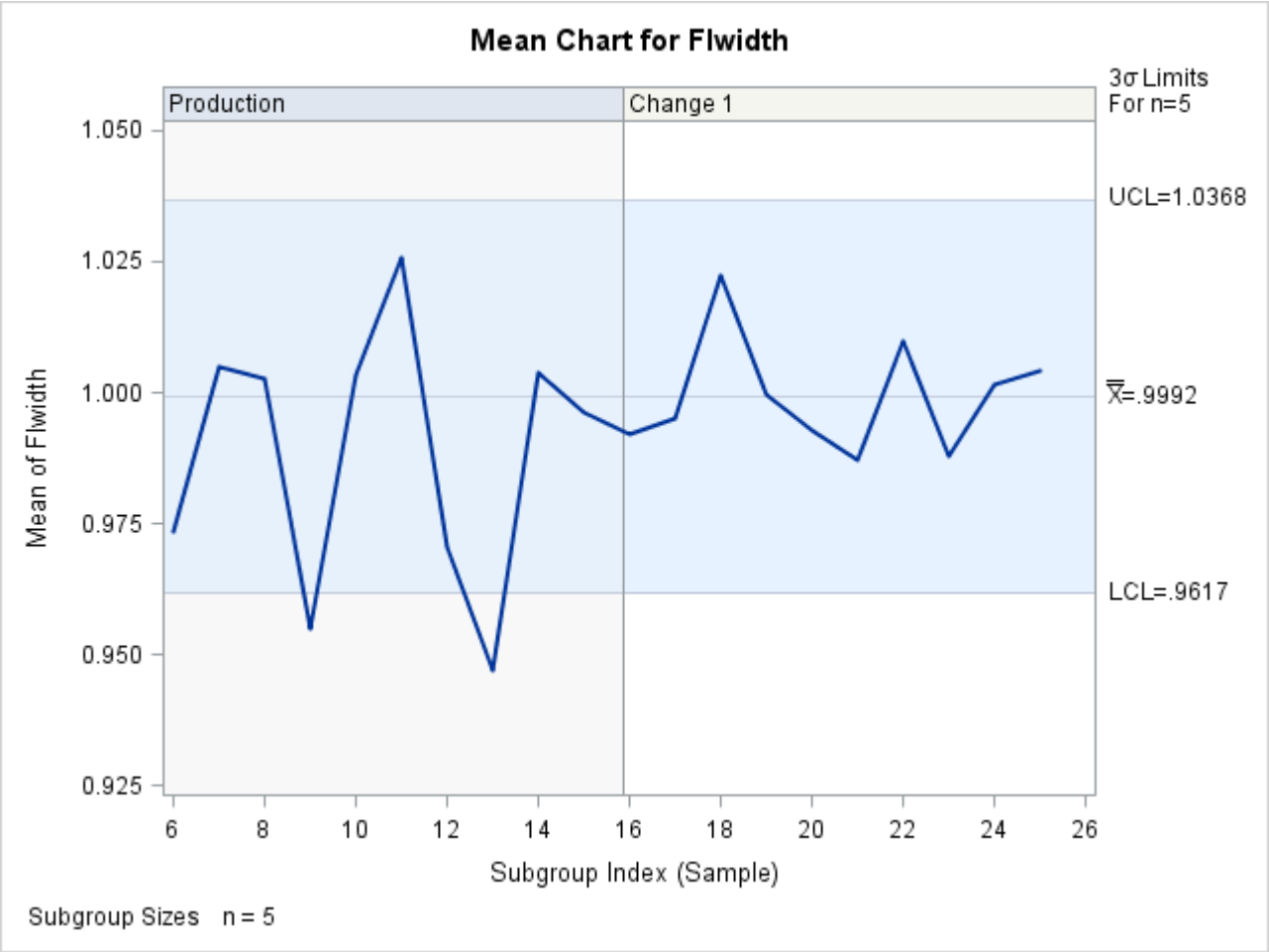
Case 7. READPHASES=*value*|*value-list* and READINDEXES= is omitted

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which _VAR_ equals the *process* and _SUBGRP_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with control limits read from the first observation in Flangelim, because this is the first observation for which _VAR_ equals 'Flwidth' and _SUBGRP_ equals 'Sample'.

The chart is displayed in Figure 19.154.

Figure 19.154 *Value-list* for READPHASES= with READINDEXES= Omitted



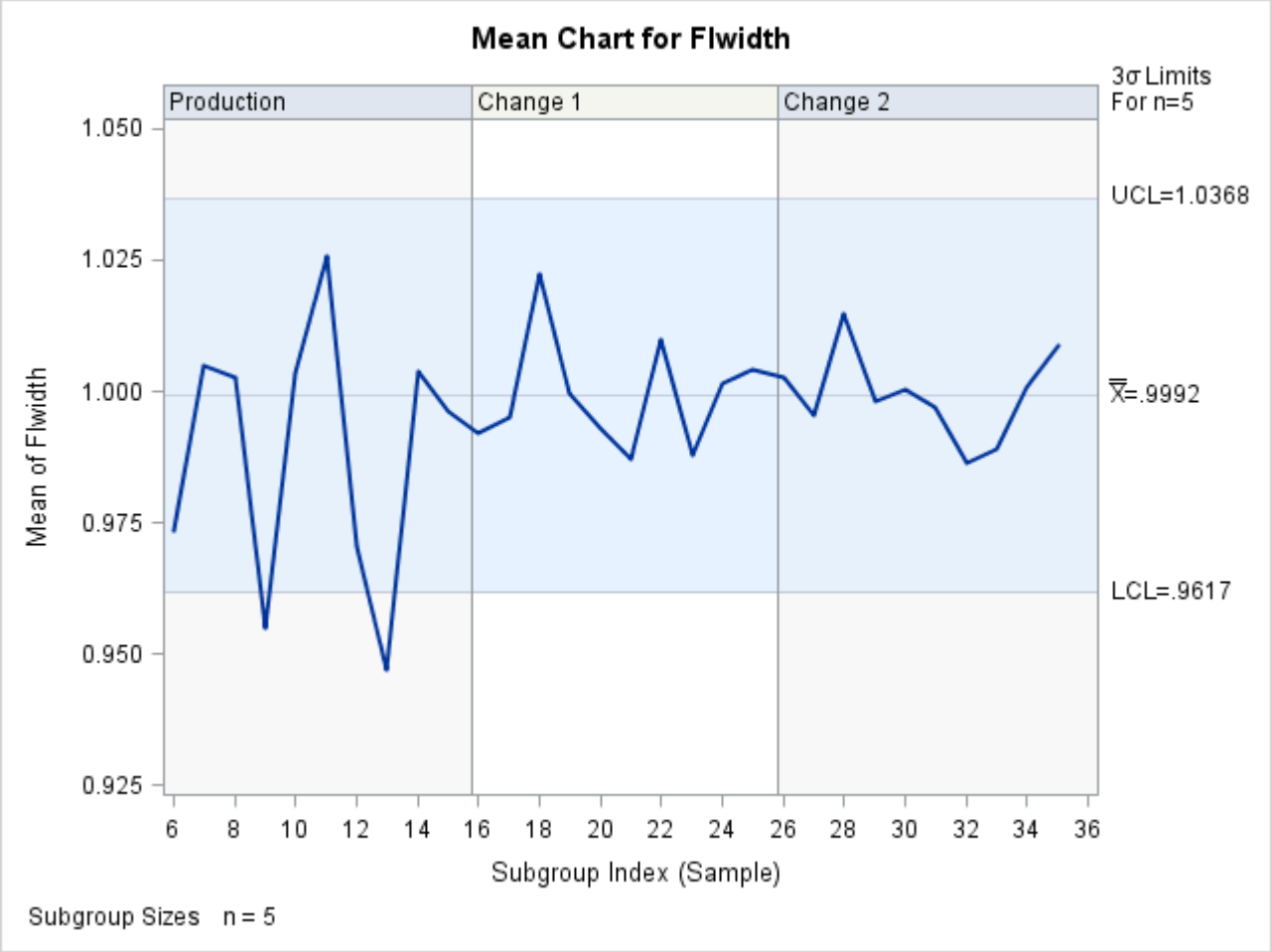
Case 8. READPHASES=ALL and READINDEXES= is omitted or READPHASES= is omitted and READINDEXES= is omitted

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which _VAR_ equals the *process* and _SUBGRP_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for all the phases in Flange with control limits read from the first observation in Flangelim, because this is the first observation for which _VAR_ equals 'Flwidth' and _SUBGRP_ equals 'Sample':

The chart is shown in Figure 19.155. Note that an identical chart would be produced if you were to omit the READPHASES= option (except that the phase reference lines and phase legends would be omitted).

Figure 19.155 READPHASES=ALL with READINDEXES= Omitted



Case 9. READPHASES=*value* and READINDEXES=*value-list*

The procedure generates an error message.

The following tables summarize the various combinations of the READPHASES= and READINDEXES= options that you can specify.

Table 19.94 READINDEXES=*index-value*

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	_INDEX_ = *index-value*
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *index-value*
Keyword ALL	_PHASE_ = *index-value*	_INDEX_ = *index-value*
Not Specified	All phases	_INDEX_ = *index-value*

Table 19.95 READINDEXES=*index-value list*

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	No chart displayed	No chart displayed
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *index-value list* with control limits matched to phases in the order listed
Keyword ALL	_PHASE_ = *index-value list*	_INDEX_ = *index-value list*
Not Specified	_PHASE_ = *index-value list*	_INDEX_ = *index-value list*

Table 19.96 READINDEXES=ALL

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	_INDEX_ = *phase-value*
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *phase-value list*
Keyword ALL	_PHASE_ = _INDEX_	_INDEX_ = _PHASE_
Not Specified	_PHASE_ = _INDEX_	_INDEX_ = _PHASE_

Table 19.97 READINDEXES= Not Specified

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	First LIMITS= observation for which _VAR_=*process* name and _SUBGRP_=*subgroup-variable* name
phase-value list	_PHASE_ = *phase-value list*	same as previous entry
Keyword ALL	All phases	same as previous entry
Not Specified	All phases	same as previous entry

Displaying Auxiliary Data with Stars

NOTE: See *Displaying Auxiliary Data with Stars* in the SAS/QC Sample Library.

In many control chart applications, it is useful to relate the variation of the process to other variables that are being observed simultaneously with the variable that is charted. You can use the features described here to represent auxiliary multivariate data with stars (polygons) that are superimposed on the control chart. See Figure 19.158 for an illustration.

This display, referred to here as a *star chart*, enables you to analyze a process with a control chart while visualizing other quantities such as environmental variables, experimental control variables, or other process variables. The control chart itself can be a standard Shewhart chart, a moving average chart (such as an EWMA chart), or a cumulative sum control chart.

The examples in this section use the HISTORY= input data set Paint (listed in Figure 19.156) and the LIMITS= data set Paintlim (listed in Figure 19.157). The data in Paint consist of the subgroup means, ranges, and sample size (pindexx, pindexr, and pindexn) for an index of paint quality that was monitored on an hourly basis, with six auxiliary variables that were measured simultaneously: thickness, gloss, defects, dust, humidity, and temperature.

Figure 19.156 Listing of the HISTORY= Data Set Paint

hour	pindexx	pindexr	pindexn	thick	gloss	defects	dust	humid	temp
1	5.8	3.0	5	0.2550	0.6800	0.2550	0.2125	0.1700	0.5950
2	6.2	2.0	5	0.2975	0.5950	0.0850	0.1700	0.2125	0.5525
3	3.7	2.5	5	0.3400	0.3400	0.4250	0.2975	0.2550	0.2125
4	3.2	6.5	5	0.3400	0.4675	0.3825	0.3485	0.2125	0.2125
5	4.7	0.5	5	0.5100	0.4250	0.5950	0.4080	0.5100	0.4675
6	5.2	3.0	5	0.5100	0.3400	0.6800	0.5525	0.5525	0.5525
7	2.6	2.0	5	0.4250	0.0425	0.8500	0.5355	0.5525	0.2550
8	2.1	1.0	5	0.3400	0.0170	0.8075	0.5950	0.5950	0.1700

Figure 19.157 Listing of the LIMITS= Data Set Paintlim

Obs	_var_	_subgrp_	_type_	_limitn_	_sigmas_	_lclx_	_mean_	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	pindex	hour	estimate	5	3	2.395	3.875	5.355	0	2.5625	5.4184	1.10171

The basic variable analyzed with the control chart (in this case, paint index) is referred to as the *process*. The auxiliary variables (in this case, thickness, gloss, defects, dust, humidity, and temperature) are referred to as *vertex variables*, because their values are represented by the vertices of the stars. A star chart can reveal relationships between the process and the vertex variables, and it can reveal relationships among the vertex variables.

You can create star charts for any number of vertex variables. However, the resolution of your graphics device and the number of subgroups per page will limit your ability to distinguish the vertices of the stars. A practical upper limit is twelve vertex variables.

You can specify star options in all chart statements of the SHEWHART procedure except the BOXCHART statement. You can use these options to

- specify the style of the star
- add reference circles to indicate limits of variation for the stars
- add a legend identifying the relationship between vertices and vertex variables
- label the vertices
- specify colors and line types for individual stars
- specify the size of the stars
- specify different methods of standardization for the vertex variables

The star options do not apply if the LINEPRINTER option is specified.

NOTE: A star chart is *not* the same as a multivariate control chart or a T^2 chart. A star chart is simply a univariate control chart enhanced with stars that represent auxiliary multivariate data. A multivariate control chart displays summary statistics (such as T^2) and control limits determined for a number of processes simultaneously. For an example of a multivariate control chart, see Figure 19.223. Figure 19.224 displays a multivariate control chart in which the principal components of the T^2 statistic are displayed with stars.

Creating a Basic Star Chart

NOTE: See *Displaying Auxiliary Data with Stars* in the SAS/QC Sample Library.

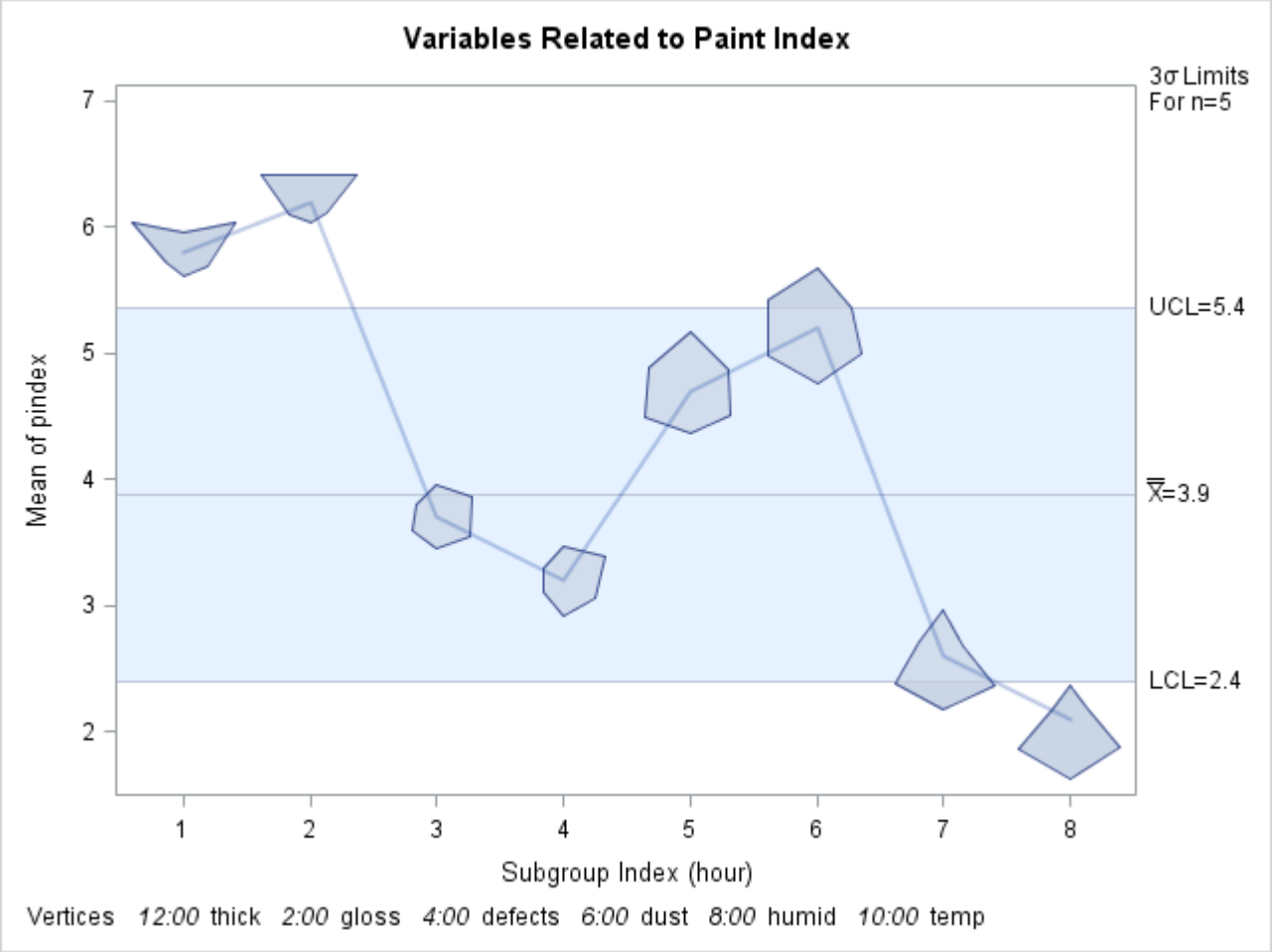
The following statements create the star chart shown in Figure 19.158:

```
ods graphics on;
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex*hour /
      nolegend
      odstitle     = title
      starvertices = (thick gloss defects dust humid temp);
run;
```

This chart is essentially an $\bar{X}$ chart for paint index. However, the chart also provides information about thickness, gloss, defects, dust, humidity, and temperature. These six variables are represented by the vertices of the stars, as indicated by the legend at the bottom of the chart. By default, the legend uses a clock representation for the vertices; for instance, dust corresponds to the vertex at the six o'clock position.

The stars are centered at the points for average paint index, and the distance from the center to a vertex represents the standardized value of the variable corresponding to the vertex. The star chart reveals that relatively high values of gloss (two o'clock) and temperature (ten o'clock) are associated with high out-of-control averages for paint index. Likewise, relatively high values of defects (four o'clock) and humidity (eight o'clock) are associated with low out-of-control averages for paint index. The star shapes reveal similarities in the data for runs 1 and 2, runs 3 and 4, runs 5 and 6, and runs 7 and 8.

Figure 19.158 A Basic Star Chart



Adding Reference Circles to Stars

NOTE: See *Displaying Auxiliary Data with Stars* in the SAS/QC Sample Library.

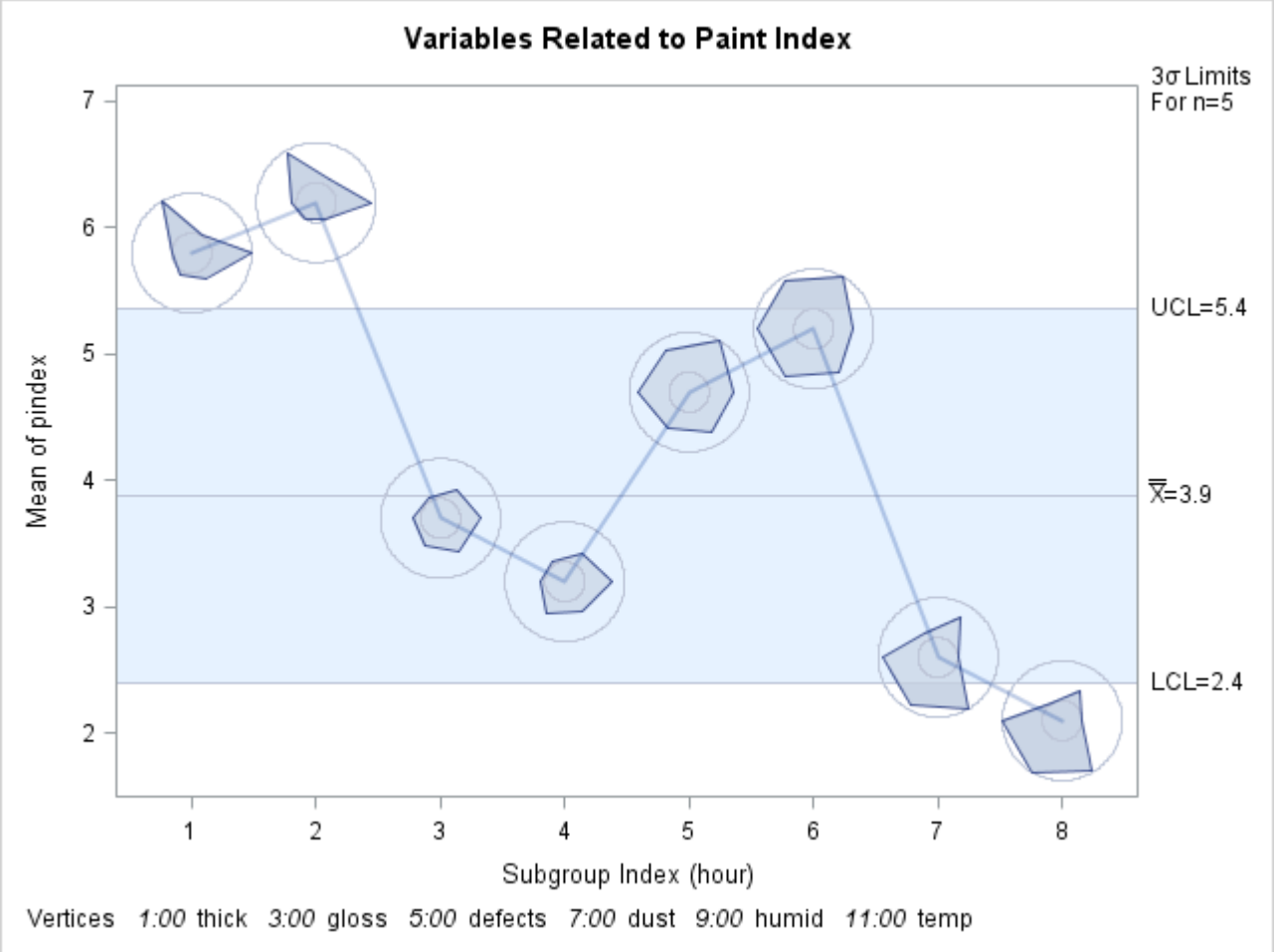
You can add reference circles to a star chart to represent limits of variation for the vertex variables. The following statements add two special reference circles, called the *inner circle* and the *outer circle*, to the star chart in Figure 19.158:

```
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex*hour /
      nolegend
      odstitle     = title
      starvertices = (thick gloss defects dust humid temp)
      starcircles  = 0.0 1.0
      lstarcircles = 1   2
      starstart    = '1:00'T ;
run;
```

The star chart shown in Figure 19.159 displays the two reference circles centered about each point. The STARCIRCLES= value 0.0 requests the *inner circle*, and the value 1.0 requests the *outer circle*. Whether or not they are displayed, these circles are always associated with each star.

The interpretation of the inner and outer circles depends on the method used to standardize the vertex variables. By default (as in this example), the data for each vertex variable are standardized by the range of the variable values taken across subgroups. That is, the inner circle represents the minimum value, and the outer circle represents the maximum value. You can specify other methods of standardization (see "Specifying the Method of Standardization" on page 2098.

Figure 19.159 Star Chart with Inner and Outer Circles Added



Note that the STARCIRCLES= option does not specify the physical radius of a reference circle. Instead, this option specifies the radius relative to the radii of the inner and outer circles. Thus, specifying STARCIRCLES=0.0 always displays the inner circle, and specifying STARCIRCLES=1.0 always displays the outer circle. Specifying STARCIRCLES=0.5 displays a reference circle halfway between the inner and outer circles. You can specify the physical radii (in percent screen units) of the inner and outer circles using the STARINRADIUS= and STAROUTRADIUS= options. In the preceding statements, the LSTARCIRCLES= option specifies line types (1=solid and 2=dashed) for the inner and outer circles. You can also use the WSTARCIRCLES= option to control the thickness of the circles.

The STARSTART= option gives the starting position for the first vertex variable listed. In the preceding example, this option specifies that the vertex corresponding to thick is to be positioned at one o'clock. The remaining vertices are uniformly spaced clockwise and correspond to the vertex variables in the order listed with the STARVERTICES= option.

For more information about the star options, see the appropriate entries in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Specifying the Style of Stars

NOTE: See *Star Charts-Specifying the Style of Stars* in the SAS/QC Sample Library.

The following statements create star charts for paint index using different styles for the stars specified with the STARTYPE= option:

```
ods graphics on;
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = wedge;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = radial;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = spoke;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = corona;
run;
```

The charts are shown in Figure 19.160, Figure 19.161, Figure 19.162, and Figure 19.163. The default style for the stars is STARTYPE=POLYGON, which is illustrated in Figure 19.158 and Figure 19.159. For more information, see the entry for the STARTYPE= option in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Figure 19.160 Star Chart Using STARTYPE=WEDGE

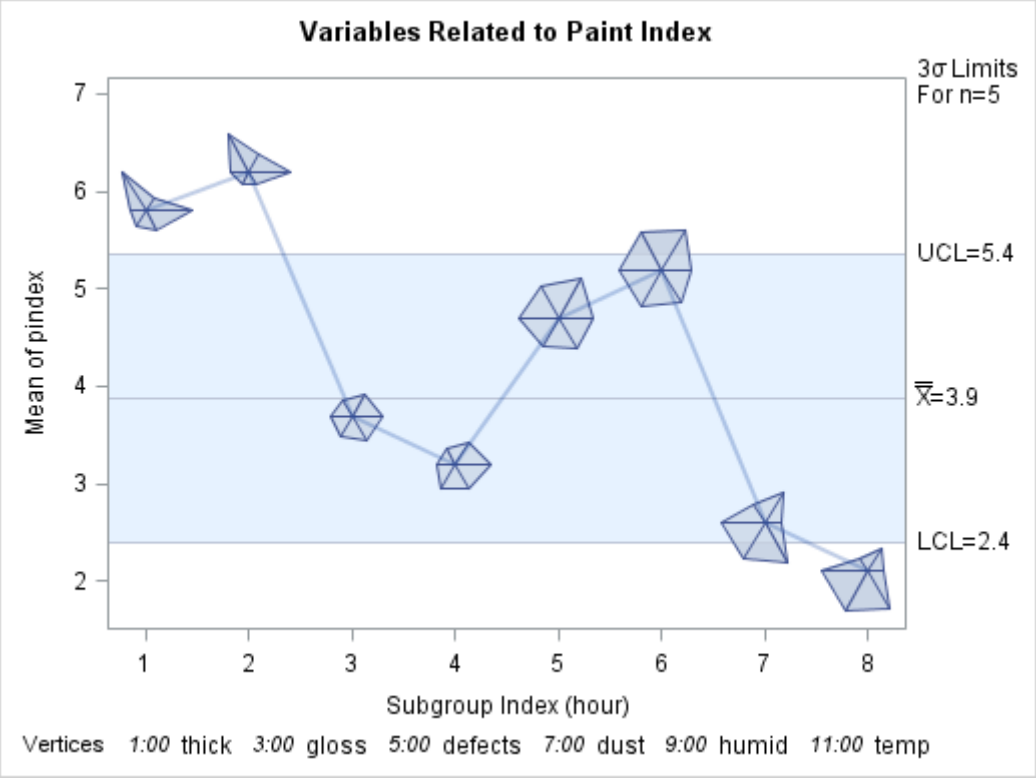


Figure 19.161 Star Chart Using STARTYPE=RADIAL

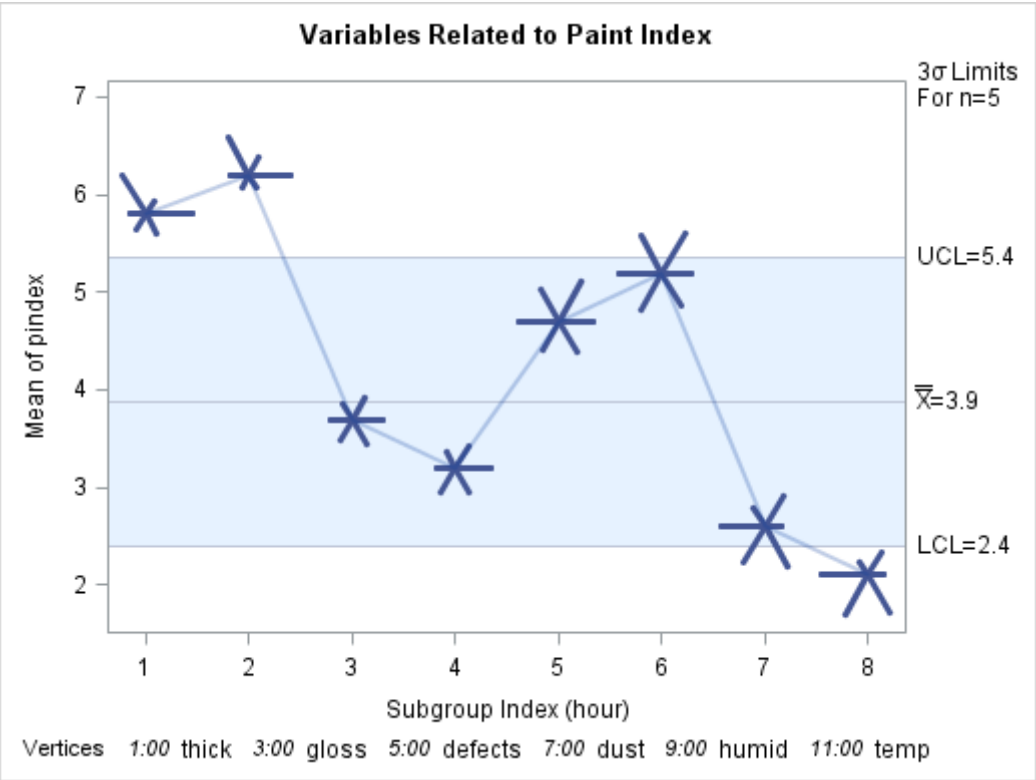


Figure 19.162 Star Chart Using STARTYPE=SPOKE

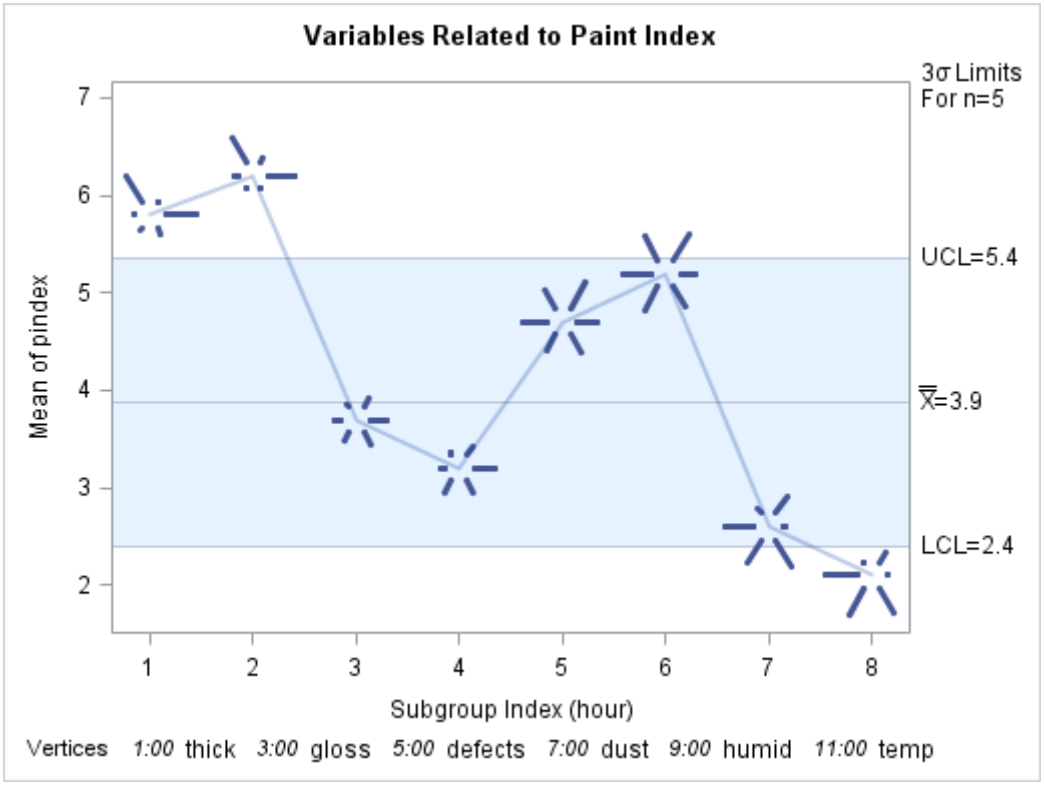
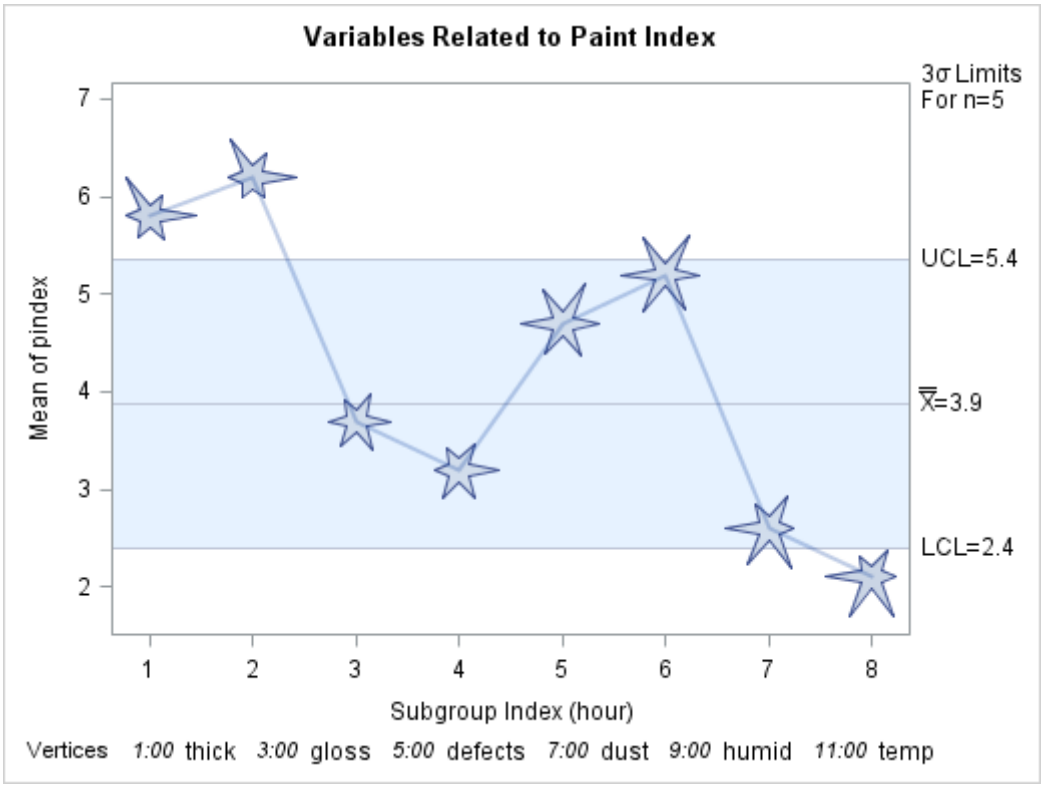


Figure 19.163 Star Chart Using STARTYPE=CORONA



Specifying the Method of Standardization

NOTE: See *Standardization Method on Star Charts* in the SAS/QC Sample Library.

In the previous examples in this section, the default method of standardization (based on ranges) is used for all six vertex variables. You can specify alternative methods with the STARSPECS= option. For example, specifying STARSPECS=3 standardizes each vertex variable so that the inner circle corresponds to three standard deviations below the mean and the outer circle corresponds to three standard deviations above the mean (that is, the circles represent 3σ limits). Specifying STARSPECS=*k* requests circles corresponding to $k\sigma$ limits, and specifying STARSPECS=0 requests the default method.

In some applications, it might be necessary to use distinct methods of standardization for the vertex variables. You can do this by creating an input SAS data set that provides the method for each vertex variable and specifying this data set with the STARSPECS= option.

The following statements create a data set named myspecs that specifies standardization methods for the vertex variables used in the previous examples:

```
data myspecs;
   length _var_     $8
          _label_   $16 ;
   input  _var_ _label_ _lspoke_ _sigmas_ _lsl_ _usl_ ;
   datalines;
thick     Thickness    1        .     0.25  0.50
gloss     Gloss        1        .     0.10  0.60
defects   Defects      1        .     0.10  0.60
dust      Dust         2      3.0      .     .
humid     Humidity     2      0.0      .     .
temp      Temperature  2      0.0      .     .
;
```

This data set contains a number of special variables whose names begin and end with an underscore.

Variable Name	Description
LABEL	Label for identifying the vertex (used in conjunction with the STARLABEL= option). This must be a character variable of length 16 or less.
LSL	Lower specification limit
LSPOKE	Line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
SIGMAS	Multiple of standard deviations above and below the average. A value of zero specifies standardization based on the range.
USL	Upper specification limit
VAR	Name of vertex variable. This must be a character variable whose length is no greater than 32.

Standardization is specified with the variables _SIGMAS_, _LSL_, and _USL_, as follows:

- Because nonmissing specification limits (_LSL_ and _USL_) are provided for the variables thick, gloss, and defects, the values of these variables are scaled so that the inner circle represents the lower specification limit and the outer circle represents the upper specification limit.

- Because _SIGMAS_ is equal to 3 for dust (and because both _LSL_ and _USL_ are missing), values of dust are scaled so that the inner circle represents three standard deviations below the mean, and the outer circle represents three standard deviations above the mean. The mean and standard deviation are calculated across all subgroups.
- Because _SIGMAS_ is equal to 0 for humid and temp (and because both _LSL_ and _USL_ are missing), values of humid and temp are scaled so that the inner circle represents the minimum and the outer circle represents the maximum. The minimum and maximum are calculated across all subgroups.

The following statements use the data set myspecs to create a star chart for paint index:

```
ods graphics on;
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      startype     = wedge
      starcircles  = 0.0 1.0
      lstarcircles = 2   2
      starstart    = -30
      labelfont    = simplex
      starlegend   = degrees
      starspecs    = myspecs
      starlabel    = high ;
run;
```

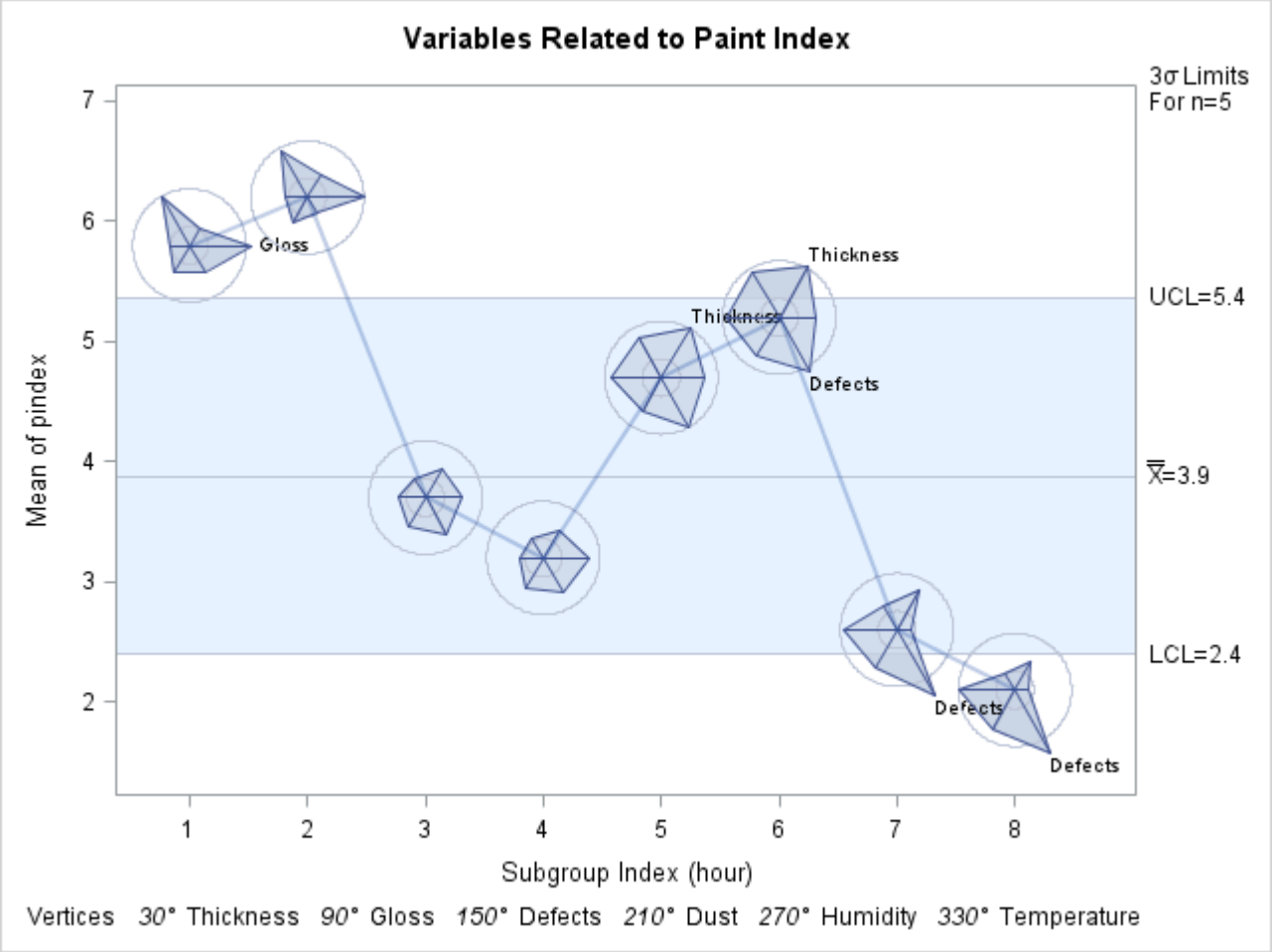
The chart is shown in Figure 19.164. Specifying STARLEGEND=DEGREES requests a legend that identifies the vertex variables by their angles (in degrees) rather than their clock positions. Here, zero degrees corresponds to twelve o'clock, and the degrees are measured clockwise. The first vertex variable is positioned at 30 degrees, as specified with the STARSTART= option. Note that you specify the STARSTART= value as a negative number to indicate that it is in degrees.

In Figure 19.161 the vertices that exceed the outer circle are labeled with the value of the variable _LABEL_ in the STARSPECS= data set. This type of labeling is requested by specifying STARLABEL=HIGH. A font (SIMPLEX) for the labels is specified with the LABELFONT= option.

The vertices for thick at hour=5, 6, and 7 are truncated, as indicated in the SAS log. The truncation value is the physical radius of an imaginary circle referred to as the *bounding circle* that lies outside the outer circle. In general, any vertex that exceeds the bounding circle is truncated to the *bounding radius*. This is done so that unusually large vertex variable values will not result in grossly distorted stars. You can specify a different bounding radius with the STARBDRADIUS= option.

The spokes corresponding to the environmental variables dust, humid, and temp are drawn with a dashed line style to distinguish them from the quality variables thick, gloss, and defects, whose spokes are drawn with a solid line. The styles are specified by the variable _LSPOKE_. Refer to *SAS/GRAPH: Reference* for a complete list of line styles. If you are producing charts in color, you can also use the variable _CSPOKE_ in the STARSPECS= data set to assign colors to the spokes.

Figure 19.164 Star Chart Using STARSPECS= Specifications



For more information about the options used in this example, see the appropriate entries in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Displaying Trends in Process Data

NOTE: See *X-Bar Chart for Data with Nonlinear Trend* in the SAS/QC Sample Library.

Time trends due to tool wear, environmental changes, and other gradual process changes are sometimes observed in $\bar{X}$ charts. The presence of a systematic trend makes it difficult to interpret the chart because the control limits are designed to indicate expected variation strictly due to common causes.

You can use the REG procedure (or other modeling procedure) in conjunction with the SHEWHART procedure to determine whether a process with a time trend is in control. With the REG procedure, you can model the trend and save the fitted subgroup means ($\hat{\bar{X}}_t$) and the residual subgroup means ($\bar{X}_t - \hat{\bar{X}}_t$) in an output data set. Then, using this data as input to the SHEWHART procedure, you can create a *trend chart*, which displays a trend plot of the fitted subgroup means together with an $\bar{X}$ chart for the residual subgroup means, thus removing the time-dependent component of the data from its random component.

Having accounted for the time trend, you can decide whether the process is in control by examining the $\bar{X}$ chart.

The following example illustrates the steps used to create a trend chart for a SAS data set named toolwear that contains diameter measurements for 20 subgroup samples each consisting of eight parts:

```
data toolwear;
   input hour @;
   do i=1 to 8;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
   1    10.0434   9.9427   9.9548   9.8056
        10.0780  10.0302  10.1173  10.0215
   2    10.1976   9.9654  10.0425  10.1183
        10.0963  10.1635  10.1382  10.1265
   3    10.0552  10.0695  10.2495  10.1753
        10.1268  10.1229  10.1351  10.2084
   4    10.1600  10.1378  10.2433  10.2634
        10.1808  10.1601  10.1035  10.0027
   5     9.9611  10.4322  10.1066  10.2653
        10.0310  10.1409  10.2709  10.0585
   6    10.2208  10.2298  10.2427  10.2315
        10.2048  10.2824  10.3347  10.1650
   7    10.2670  10.3793  10.2539  10.4037
        10.3281  10.1327  10.1986  10.1841
   8    10.2537  10.1981  10.2935  10.4308
        10.3195  10.3122  10.2033  10.3220
   9    10.2488  10.1866  10.3678  10.1755
        10.3225  10.2375  10.2466  10.3387
  10    10.3744  10.5221  10.2890  10.3123
        10.5134  10.3212  10.3139  10.1565
  11    10.3525  10.3237  10.4605  10.5139
        10.3650  10.1171  10.3863  10.2061
  12    10.3279  10.3338  10.1885  10.2810
        10.2400  10.3617  10.2938  10.2656
  13    10.1651  10.2404  10.1814  10.2330
        10.3094  10.3373  10.3266  10.3830
  14    10.3554  10.4577  10.5435  10.4805
        10.5358  10.4631  10.3689  10.1750
  15    10.2962  10.4221  10.3578  10.4694
        10.3465  10.4499  10.4645  10.3986
  16    10.6002  10.1924  10.3437  10.3228
        10.3438  10.3503  10.3761  10.3137
  17    10.4015  10.3592  10.3187  10.4108
        10.4834  10.4807  10.2178  10.3897
  18    10.4514  10.4492  10.3373  10.4497
        10.4197  10.3496  10.3949  10.1585
  19    10.3445  10.3310  10.4472  10.4684
        10.3975  10.2714  10.2952  10.6255
  20    10.2612  10.3824  10.4240  10.3120
        10.5744  10.4204  10.4073  10.3783
;
```

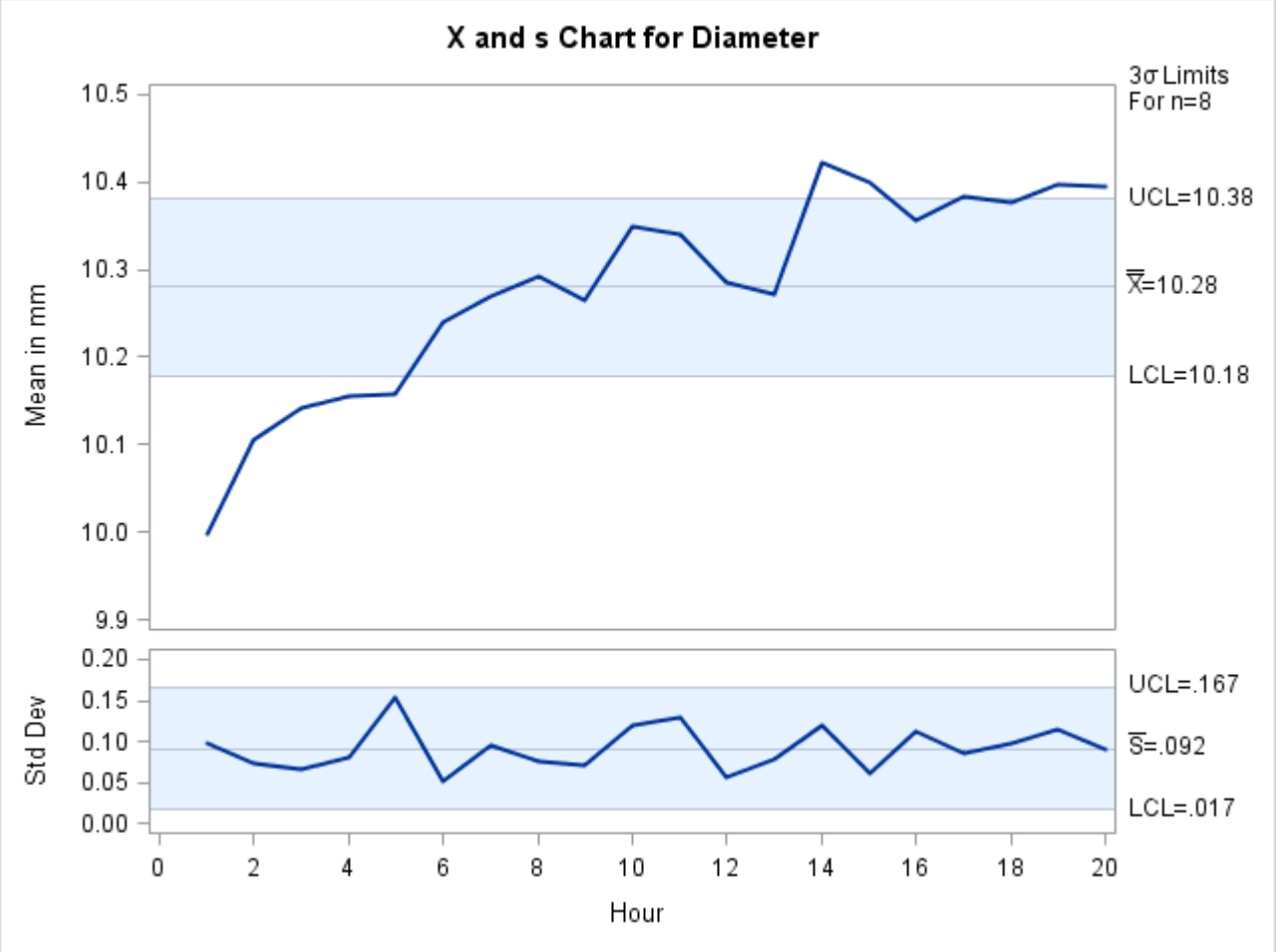
Step 1: Preliminary Mean and Standard Deviation Charts

The following statements create $\bar{X}$ and s charts for the diameter data:

```
ods graphics on;
title f=qcfont1 'X ' f=none 'and s Chart for Diameter';
proc shewhart data=toolwear;
   xschart Diameter*hour /
      odstitle   = title
      outhistory = submeans
      nolegend ;
   label Diameter = 'Mean in mm';
   label hour     = 'Hour';
run;
```

The charts are shown in Figure 19.165. The subgroup standard deviations are all within their control limits, indicating the process variability is stable. However, the $\bar{X}$ chart displays a nonlinear trend that makes it difficult to decide if the process is in control. Subsequent investigation reveals that the trend is due to tool wear.

Figure 19.165 $\bar{X}$ and s Charts for toolwear Data



Note that the symbol $\bar{X}$ is displayed in the title with the special font QCFONT4, which matches the SWISS font used for the remainder of the title. See Chapter D, "Special Fonts in SAS/QC Software," for a description of the fonts available for displaying $\bar{X}$ and related symbols.

Step 2: Modeling the Trend

The next step is to model the trend as a function of hour. The $\bar{X}$ chart in Figure 19.165 suggests that the mean level of the process (saved as DiameterX in the OUTLIMITS= data set submeans) grows as the log of hour. The following statements fit a simple linear regression model in which DiameterX is the response variable and loghour (the log transformation of hour) is the predictor variable. Part of the printed output produced by PROC REG is shown in Figure 19.166.

```
data submeans;
   set submeans;
   loghour=log(hour);
run;

proc reg data=submeans ;
   model Diameterx=loghour;
   output out=regdata predicted=fitted ;
run;
```

Figure 19.166 Trend Analysis for Diameter from PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: DiameterX Mean of Diameter

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	9.99056	0.02185	457.29	<.0001
loghour		1	0.13690	0.00967	14.16	<.0001

Figure 19.166 shows that the fitted equation can be expressed as

$$\hat{\bar{X}}_t = 9.99 + 0.14 \times \log(t)$$

where $\hat{\bar{X}}_t$ is the fitted subgroup average.[12] A partial listing of the OUT= data set REGDATA created by the REG procedure is shown in Figure 19.167.

Figure 19.167 Partial Listing of the Output Data Set regdata from the REG Procedure

hour	DiameterX	DiameterS	DiameterN	loghour	fitted
1	9.9992	0.09726	8	0.00000	9.9906
2	10.1060	0.07290	8	0.69315	10.0855
3	10.1428	0.06601	8	1.09861	10.1410
4	10.1565	0.08141	8	1.38629	10.1803
5	10.1583	0.15454	8	1.60944	10.2109

[12] Although this example does not check for the existence of a trend, you should do so by using the hypothesis tests provided by the REG procedure.

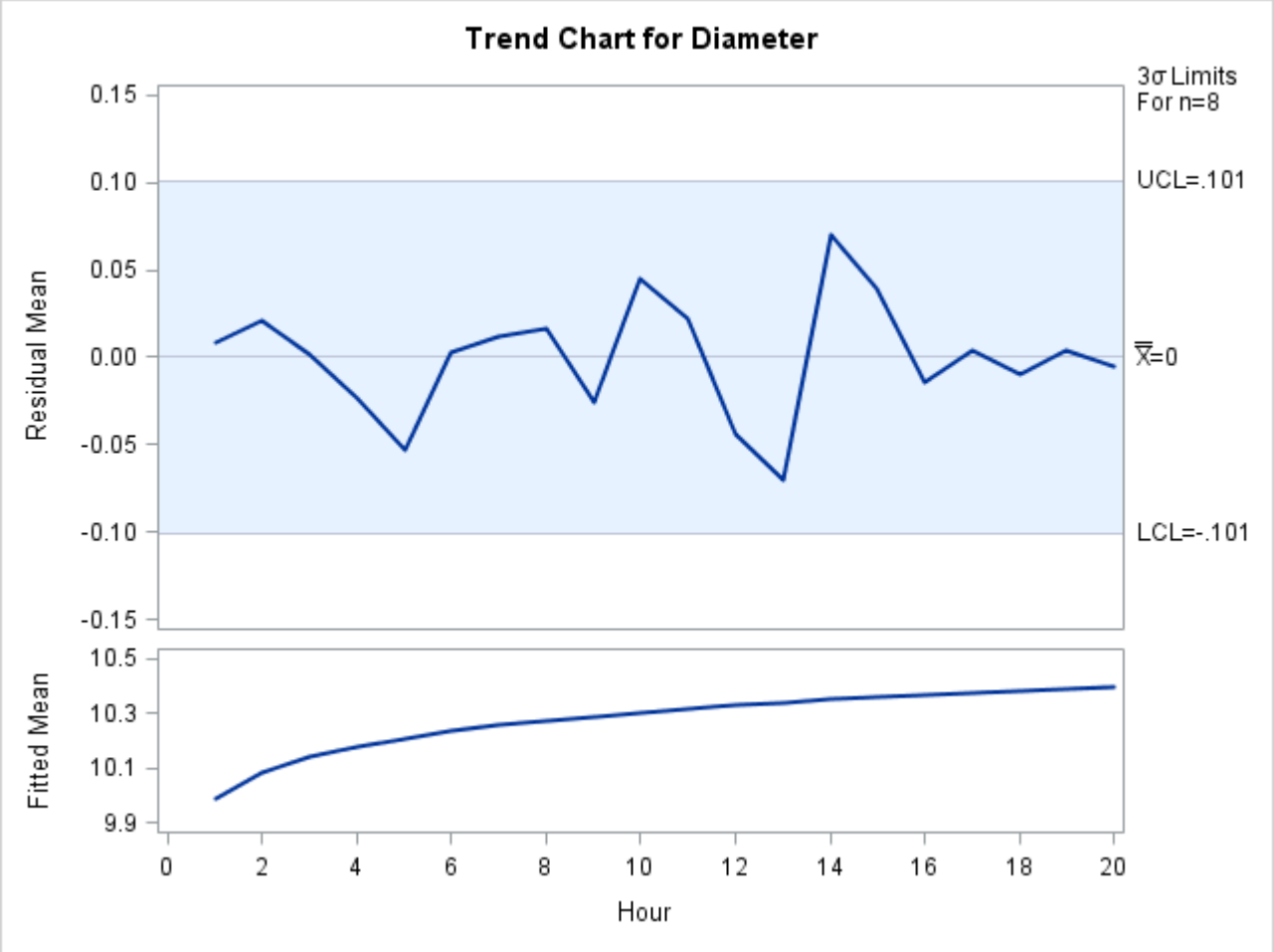
Step 3: Displaying the Trend Chart

The third step is to create a trend chart with the SHEWHART procedure, as follows:

```
title 'Trend Chart for Diameter';
proc shewhart history=regdata;
   xchart Diameter*hour /
      trendvar = fitted
      split    = '/'
      odstitle = title
      stddevs
      nolegend;
   label Diameterx = 'Residual Mean/Fitted Mean';
   label hour      = 'Hour';
run;
```

The chart is shown in Figure 19.168. The values of fitted are plotted in the lower half of the trend chart. The upper half of the trend chart is an $\bar{X}$ chart for the residual means (DiameterX – fitted). The $\bar{X}$ chart in Figure 19.168 shows that, after accounting for the trend, the mean level of the process is in control.

Figure 19.168 Trend Chart for Diameter Data



If the data are correlated in time, you can use the ARIMA or AUTOREG procedures in place of the REG procedure to remove autocorrelation structure and display a control chart for the residuals; for an example, see "Autocorrelation in Process Data" on page 2144. Another application of the TRENDVAR= option is the display of nominal values in control charts for short runs; see "Short Run Process Control" on page 2160.

Clipping Extreme Points

NOTE: See *Clipping Extreme Points* in the SAS/QC Sample Library.

In some control chart applications, the out-of-control points can be so extreme that the remaining points are compressed to a scale that is difficult to read. In such cases, you can clip the extreme points so that a more readable chart is displayed, as illustrated in the following example.

A company producing copper tubing uses $\bar{X}$ and R charts to monitor the diameter of the tubes. Based on previous production, known values of 70mm and 0.75mm are available for the mean and standard deviation of the diameter. The diameter measurements (in millimeters) for 15 batches of five tubes each are provided in the data set newtubes.

```
data newtubes;
   label Diameter='Diameter in mm';
   do batch = 1 to 15;
      do  i = 1 to 5;
         input Diameter @@;
         output;
      end;
   end;
   datalines;
69.13  69.83  70.76  69.13  70.81
85.06  82.82  84.79  84.89  86.53
67.67  70.37  68.80  70.65  68.20
71.71  70.46  71.43  69.53  69.28
71.04  71.04  70.29  70.51  71.29
69.01  68.87  69.87  70.05  69.85
50.72  50.49  49.78  50.49  49.69
69.28  71.80  69.80  70.99  70.50
70.76  69.19  70.51  70.59  70.40
70.16  70.07  71.52  70.72  70.31
68.67  70.54  69.50  69.79  70.76
68.78  68.55  69.72  69.62  71.53
70.61  70.75  70.90  71.01  71.53
74.62  56.95  72.29  82.41  57.64
70.54  69.82  70.71  71.05  69.24
;
```

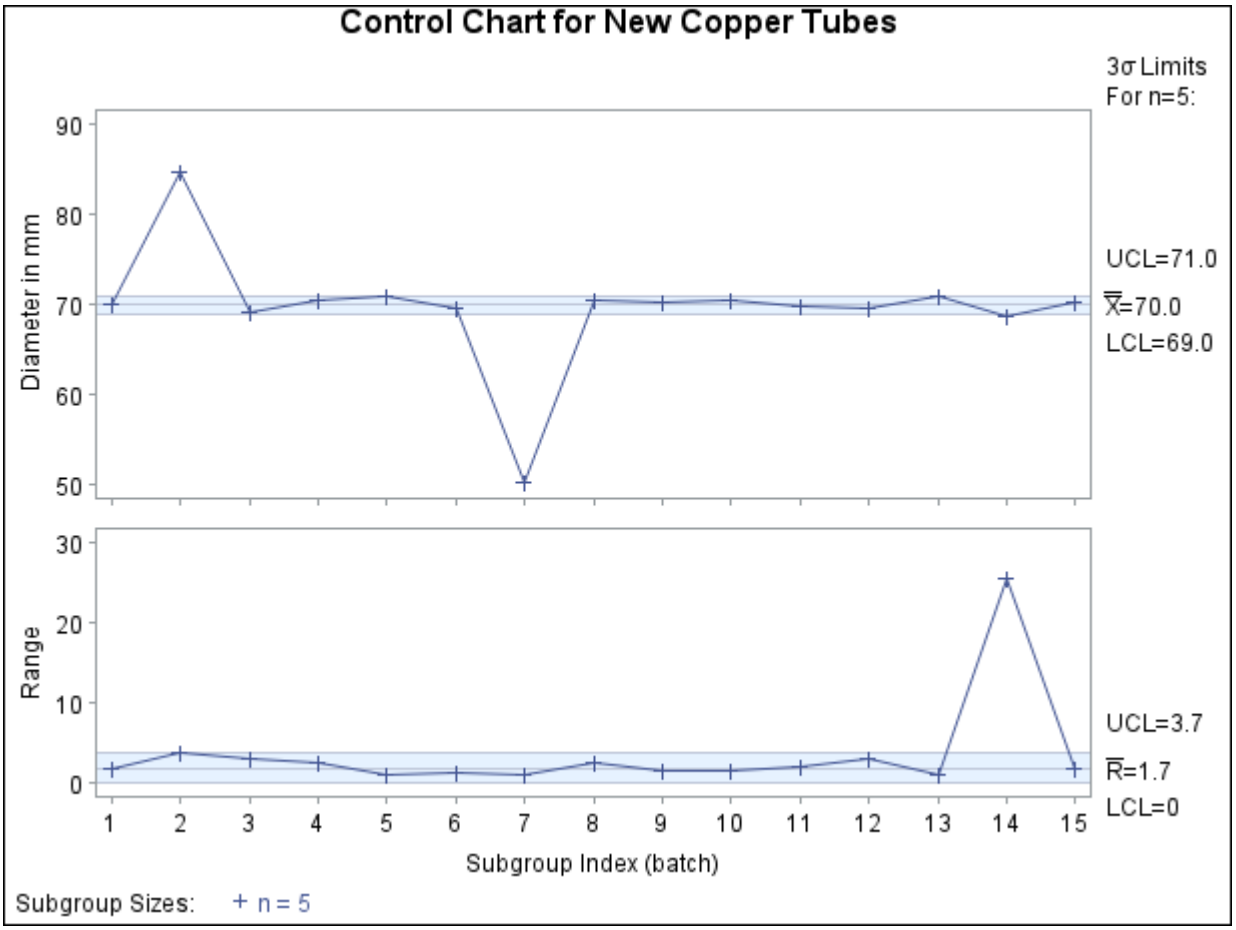
The following statements create the $\bar{X}$ and R charts shown in Figure 19.169 for the tube diameter:

```
ods graphics off;
symbol value=plus h=3.0 pct;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart Diameter*batch /
      mu0    = 70
```

```
        sigma0 = 0.75;
run;
```

Batches 2 and 7 result in extreme out-of-control points on the mean chart, and batch 14 results in an extreme out-of-control point on the range chart. The vertical axes are scaled to accommodate these extreme out-of-control points, and this in turn forces the control limits to be compressed.

Figure 19.169 $\bar{X}$ and R Charts Without Clipping



You can request clipping by specifying the option CLIPFACTOR=*factor*, where *factor* is a value greater than one (useful values are typically in the range 1.5 to 2). Clipping is applied in two steps, as follows:

1. If a plotted statistic is greater than $y_{\max}$, it is temporarily set to $y_{\max}$, where

   $$y_{\max} = \text{LCL} + (\text{UCL} - \text{LCL}) \times factor$$

   If a plotted statistic is less than $y_{\min}$, it is temporarily set to $y_{\min}$, where

   $$y_{\min} = \text{UCL} - (\text{UCL} - \text{LCL}) \times factor$$

2. Axis scaling is applied to the clipped statistics. Then the $y_{\max}$ values are reset to the maximum value on the axis and the $y_{\min}$ values are reset to the minimum value on the axis.

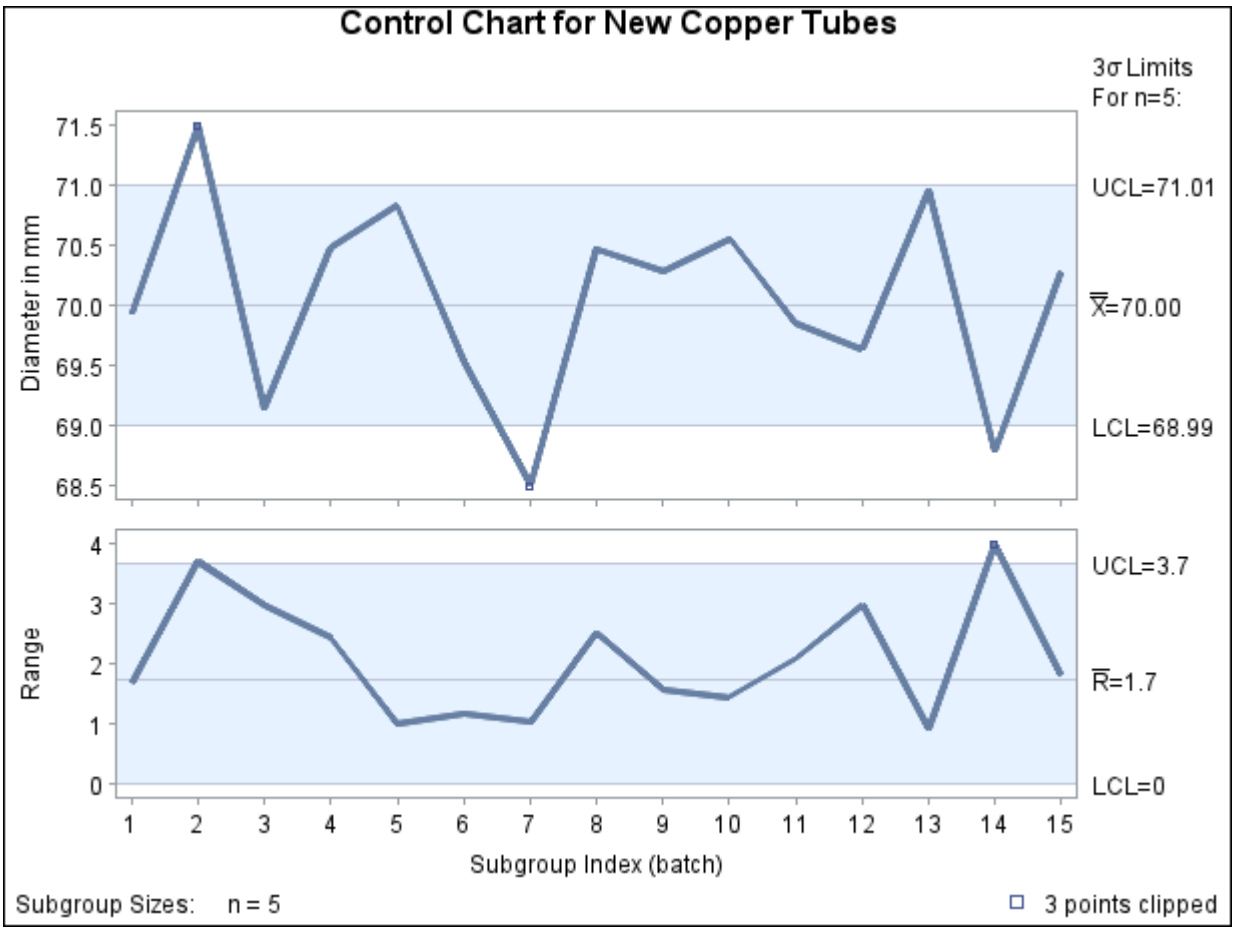
Notes:

- Clipping is applied only to the plotted statistics and not to the statistics tabulated or saved in an output data set.
- Because the *factor* must be greater than one, clipping does not affect whether a plotted statistic is inside or outside the control limits.
- Tests for special causes are applied to the plotted statistics before they are clipped, and clipping does not affect how the tests are flagged on the chart. In some situations, however, clipping can make the patterns associated with the tests less evident on the chart.
- When primary and secondary charts are displayed, the same clipping *factor* is applied to both charts.
- A special symbol is used for clipped points (the default symbol is a square), and a legend is added to the chart indicating the number of points that were clipped.

The following statements create $\bar{X}$ and R charts, shown in Figure 19.170, that use a clipping factor of 1.5:

```
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart Diameter*batch /
      mu0        = 70
      sigma0     = 0.75
      clipfactor = 1.5;
run;
```

Figure 19.170 $\bar{X}$ and R Charts with Clip Factor of 1.5

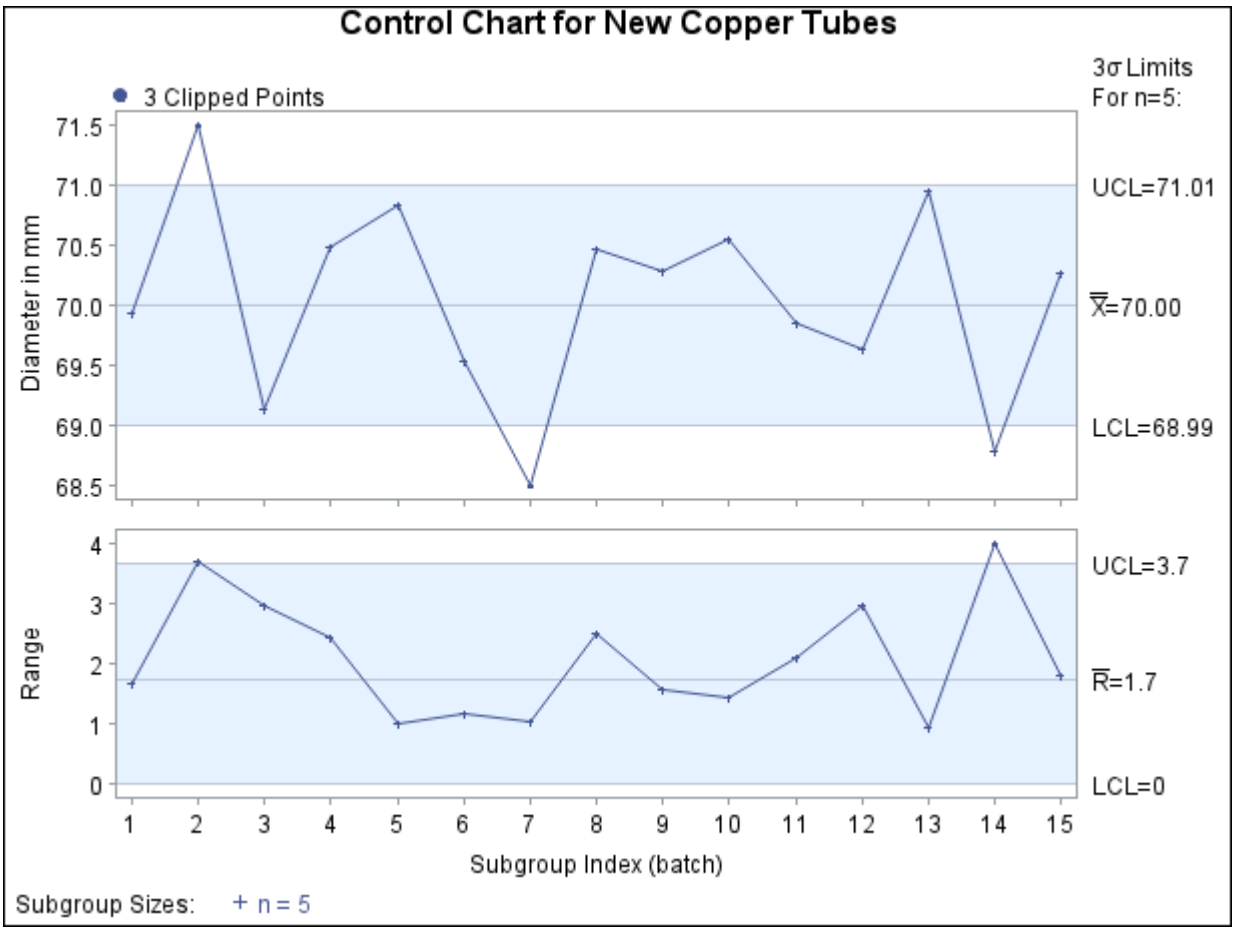


In Figure 19.170, the extreme out-of-control points are clipped making the points plotted within the control limits more readable. The clipped points are marked with a square, and a clipping legend is added at the lower right of the display.

Other clipping options are available, as illustrated by the following statements:

```
symbol value=plus;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart Diameter*batch /
      mu0         = 70
      sigma0      = 0.75
      clipfactor  = 1.5
      clipsymbol  = dot
      cliplegpos  = top
      cliplegend  = '# Clipped Points'
      clipsubchar = '#';
run;
```

Figure 19.171 $\bar{X}$ and R Charts Using Clipping Options



Specifying CLIPSYMBOL=DOT marks the clipped points with a dot instead of the default square. Specifying CLIPLEGPOS=TOP positions the clipping legend at the top of the chart. The options CLIPLEGEND='# Clipped Points' and CLIPSUBCHAR='#' request the clipping legend *3 Clipped Points*. For more information about the clipping options, see the appropriate entries in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Labeling Axes

NOTE: See *Labeling Axes on Shewhart Charts* in the SAS/QC Sample Library.

The SHEWHART procedure provides default labels for the horizontal and vertical axes of control charts. You can specify axis labels by assigning labels to variables, as discussed in the following sections.

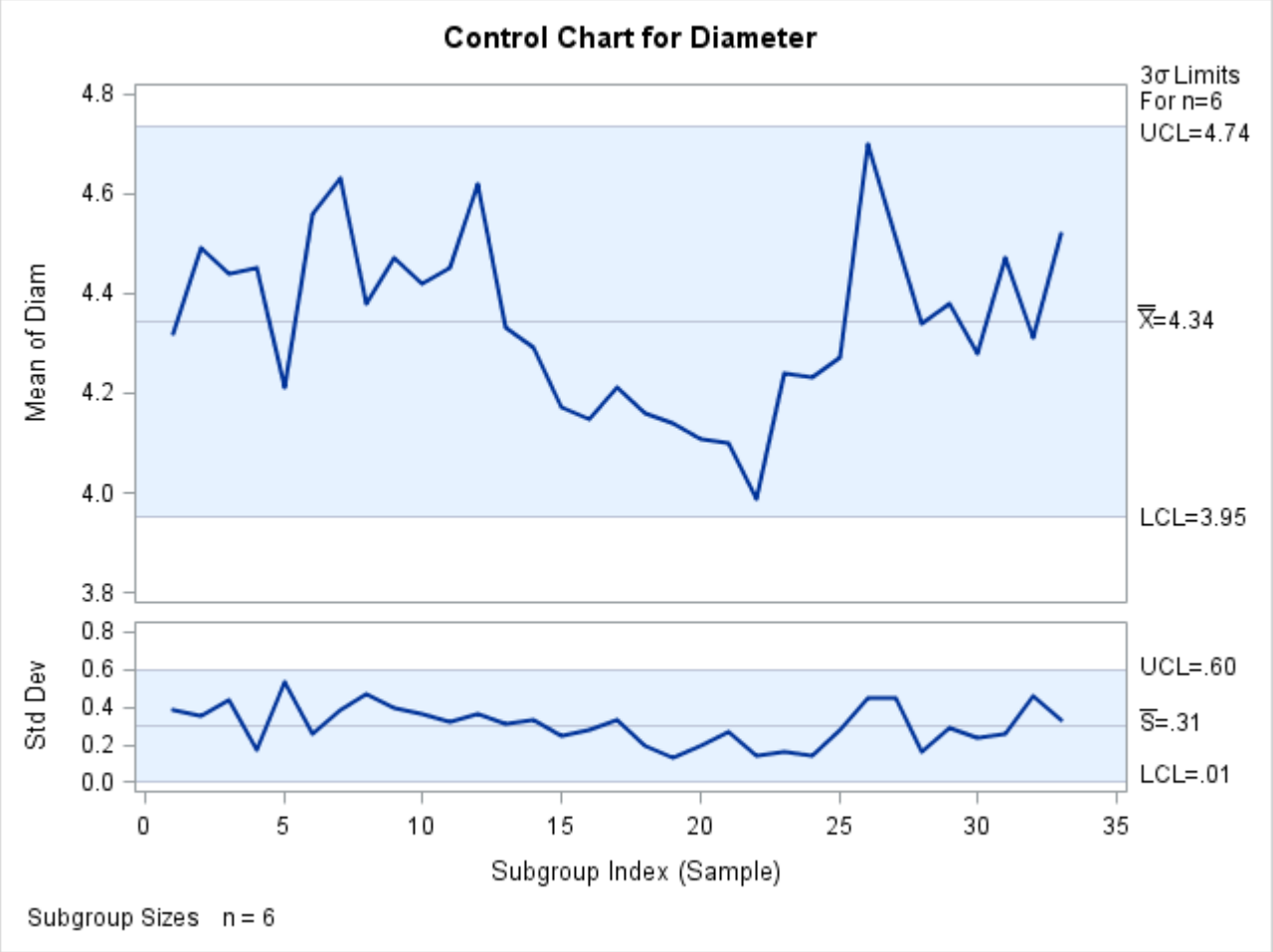
Default Labels

If a label is not associated with the *subgroup-variable*, the default horizontal axis label is "Subgroup Index (*subgroup-variable*)." The default vertical axis label for a primary chart identifies the chart type and the process variable. The default vertical axis label for a secondary chart identifies the chart type only.

For example, the following statements create $\bar{X}$ and s charts with default labels using the data set Parts given in "Displaying Stratified Process Data" on page 2071. The resulting charts are displayed in Figure 19.172.

```
ods graphics on;
title 'Control Chart for Diameter';
proc shewhart history=Parts;
   xschart Diam*Sample / odstitle = title;
run;
```

Figure 19.172 Control Charts with Default Labels



Labeling the Horizontal Axis

You can specify a label of up to 40 characters for the horizontal axis by assigning the label to the *subgroup variable* with a LABEL statement (refer to *SAS Statements: Reference* for a description of LABEL statements). If you use a LABEL statement after the PROC SHEWHART statement and before the RUN statement, the label is associated with the variable only for the duration of the PROC step.

For an example, see "Labeling the Vertical Axis" on page 2111, where Figure 19.173 redisplays the $\bar{X}$ and s charts in Figure 19.172 with specified horizontal and vertical axis labels.

Labeling the Vertical Axis

You can specify a label for the vertical axis of a primary chart by using a LABEL statement to assign the label to a particular variable in the input data set. The type of input data set, the chart statement, and the *process* specified in the chart statement determine which variable to use in the LABEL statement.

- If the input data set is a DATA= data set, assign the label to the process variable (*process*) specified in the chart statement.
- If the input data set is a HISTORY= data set, assign the label to the variable specified in the chart statement whose name begins with the prefix *process* and ends with the appropriate suffix given by the following list:

Chart Statement	Suffix
BOXCHART with CONTROLSTAT=MEAN	X
BOXCHART with CONTROLSTAT=MEDIAN	M
CCHART	U
IRCHART	none
MCHART	M
MRCHART	M
NPCHART	P
PCHART	P
RCHART	R
SCHART	S
UCHART	U
XCHART	X
XRCHART	X
XSCHART	X

  If the prefix *process* consists of 32 characters, shorten the prefix to its first 16 characters and last 15 characters before adding the suffix.

- If the input data set is a TABLE= data set, assign the label to the predefined variable given by the following table:

Chart Statement	Variable
BOXCHART with CONTROLSTAT=MEAN	_SUBX_
BOXCHART with CONTROLSTAT=MEDIAN	_SUBMED_
CCHART	_SUBC_
IRCHART	_SUBI_
MCHART	_SUBMED_
MRCHART	_SUBMED_
NPCHART	_SUBNP_
PCHART	_SUBP_
RCHART	_SUBR_
SCHART	_SUBS_
UCHART	_SUBU_

Chart Statement	Variable
XCHART	_SUBX_
XRCHART	_SUBX_
XSCHART	_SUBX_

If the chart statement produces primary and secondary charts, as in the case of the XSCHART statement, you can break the label into two parts by including a split character in the label. The part before the split character labels the vertical axis of the primary chart, and the part after the split character labels the vertical axis of the secondary chart. To specify the split character, use the SPLIT= option in the chart statement.

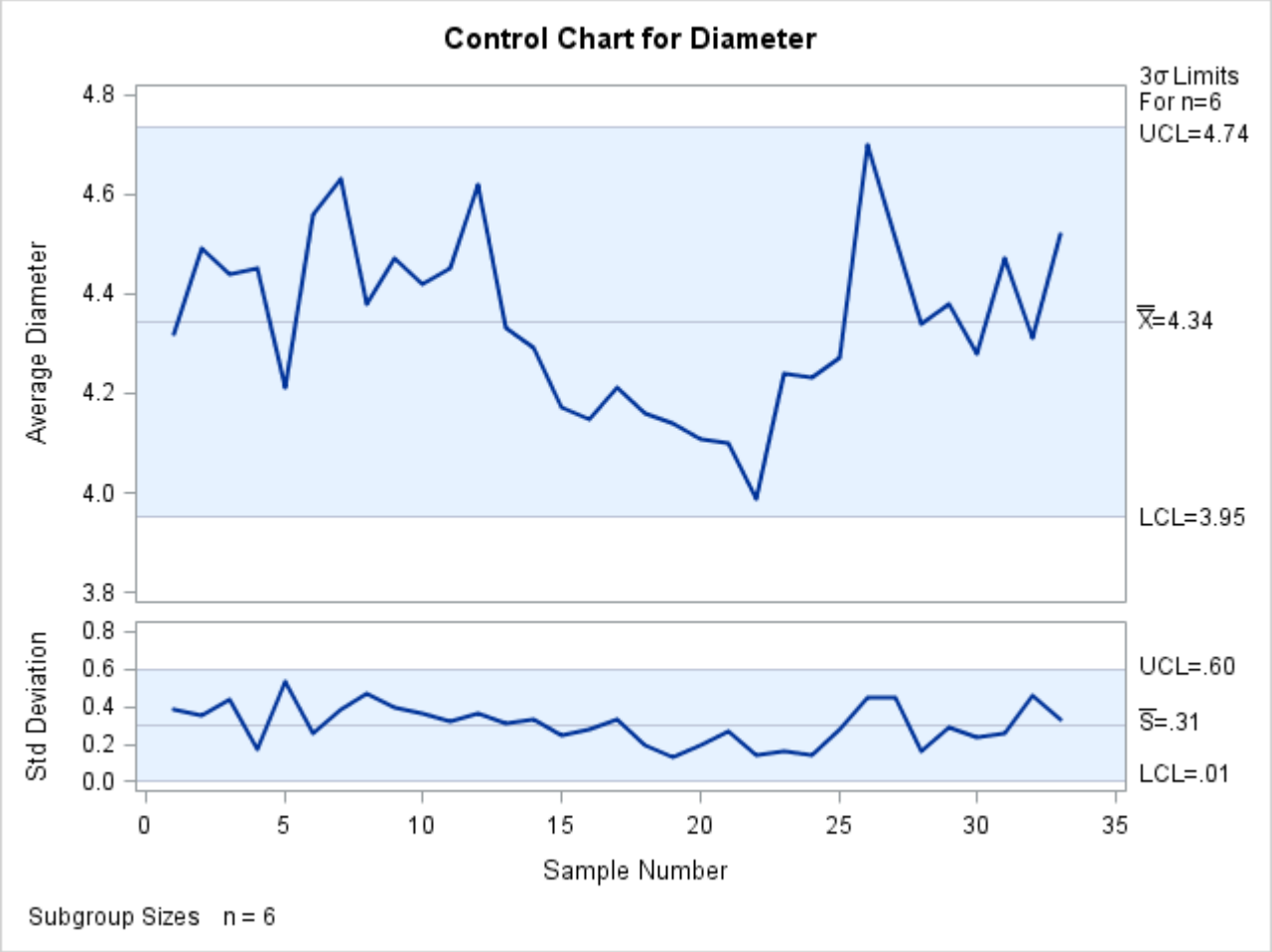
For example, the following statements redisplay the $\bar{X}$ and s charts in Figure 19.172 with specified labels for the horizontal and vertical axes:

```
title 'Control Chart for Diameter';
proc shewhart history=Parts;
   xschart Diam*Sample / split    = '/'
                         odstitle = title;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter/Std Deviation';
run;
```

The charts are displayed in Figure 19.173. Because the input data set Parts is a HISTORY= data set, the vertical axes are labeled by assigning a label to the subgroup mean variable DiamX (that is, the *process* Diam with the suffix X).[13] Assigning a label to Diam would result in an error message because Diam is interpreted as a prefix rather than a SAS variable.

[13] If the *process* were Diameter rather than Diam, the label would be assigned to the variable DiameterX.

Figure 19.173 Control Charts with Axis Labels Specified



If the input data set were a DATA= data set rather than a HISTORY= data set, you would associate the label with the variable Diam. If the input data set were a TABLE= data set, you would associate the label with the variable _SUBX_.

For another illustration, see Example 19.17.

Selecting Subgroups for Computation and Display

This section describes methods for specifying which subgroups of observations in an input data set (DATA=, HISTORY=, or TABLE=) are to be used to compute control limits and which subgroups are to be displayed as points on the chart.

Using WHERE Statements

NOTE: See *Selecting Subgroups Using WHERE Statements* in the SAS/QC Sample Library.

The following statements create a data set named Bottles that records the number of cracked bottles encountered each day during two months (January and February) of a soft drink bottling operation:

```
data Bottles;
   informat Day date7.;
   format Day date7.;
   nBottles = 3000;
   input Day nCracks @@;
   datalines;
04JAN94  61  05JAN94  56  06JAN94  71  07JAN94  56
10JAN94  51  11JAN94  64  12JAN94  71  13JAN94  91
14JAN94  98  17JAN94  68  18JAN94  63  19JAN94  60
20JAN94  58  21JAN94  55  24JAN94  78  25JAN94  47
26JAN94  54  27JAN94  69  28JAN94  73  31JAN94  66
01FEB94  57  02FEB94  55  03FEB94  63  04FEB94  50
07FEB94  69  08FEB94  54  09FEB94  64  10FEB94  66
11FEB94  70  14FEB94  49  15FEB94  57  16FEB94  56
17FEB94  59  18FEB94  66  21FEB94  60  22FEB94  58
23FEB94  67  24FEB94  60  25FEB94  62  28FEB94  48
;
```

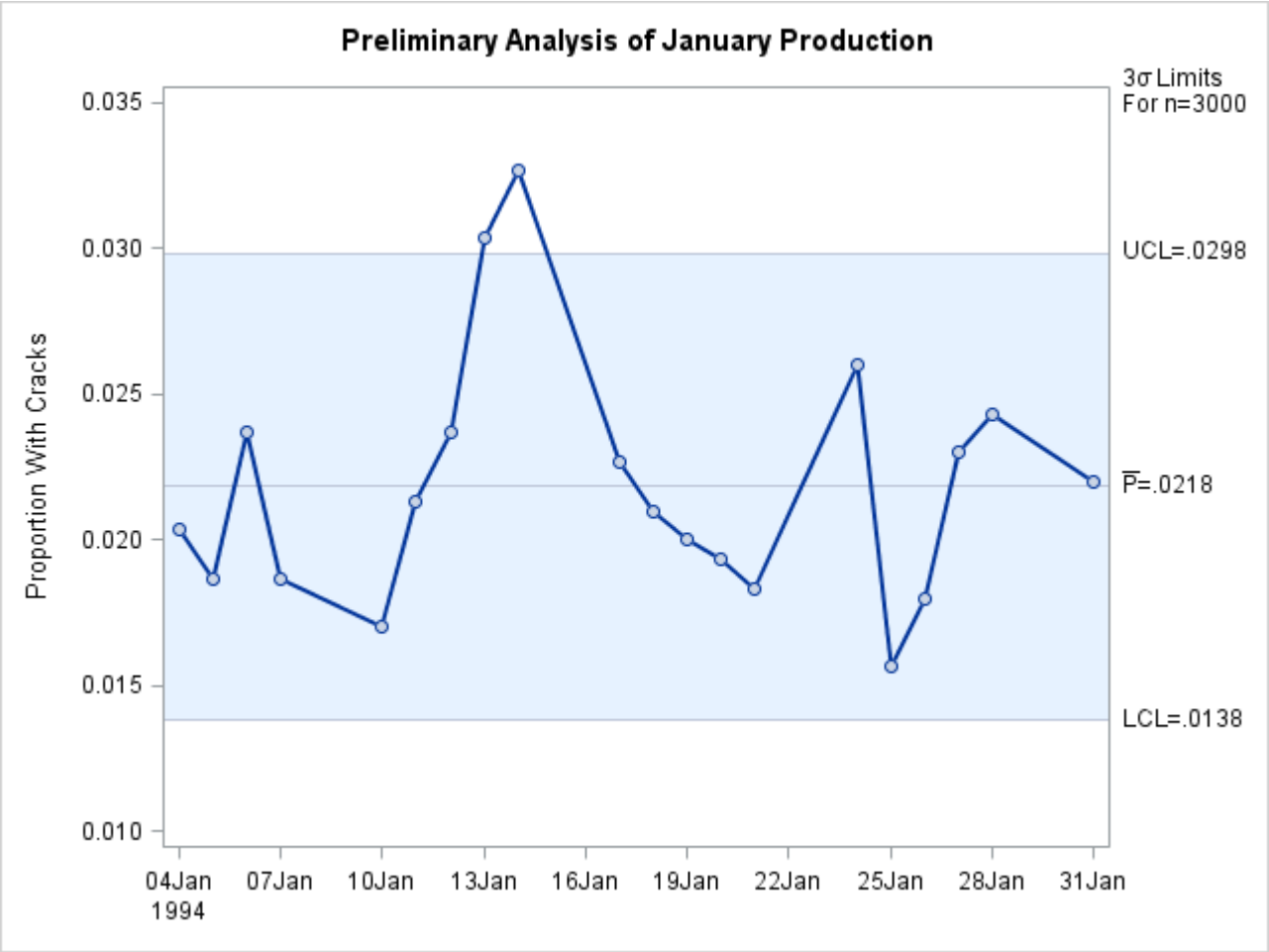
The variable nBottles contains the number of bottles sampled each day, and the variable nCracks contains the number of cracked bottles in each sample.

The following statements create a *p* chart for the number of cracked bottles based on the January production:

```
ods graphics on;
title 'Preliminary Analysis of January Production';
proc shewhart data=Bottles;
   where Day <= '31JAN94'D;
   pchart nCracks * Day / subgroupn = nBottles
                          nohlabel
                          nolegend
                          markers
                          odstitle  = title
                          outlimits = mylim;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 19.174. The WHERE statement restricts the observations read from Bottles so that the control limits are estimated from the January data, and only the January data are displayed on the chart. For details concerning the WHERE statement, refer to *SAS Statements: Reference*.

Figure 19.174 Preliminary p Chart for January Data

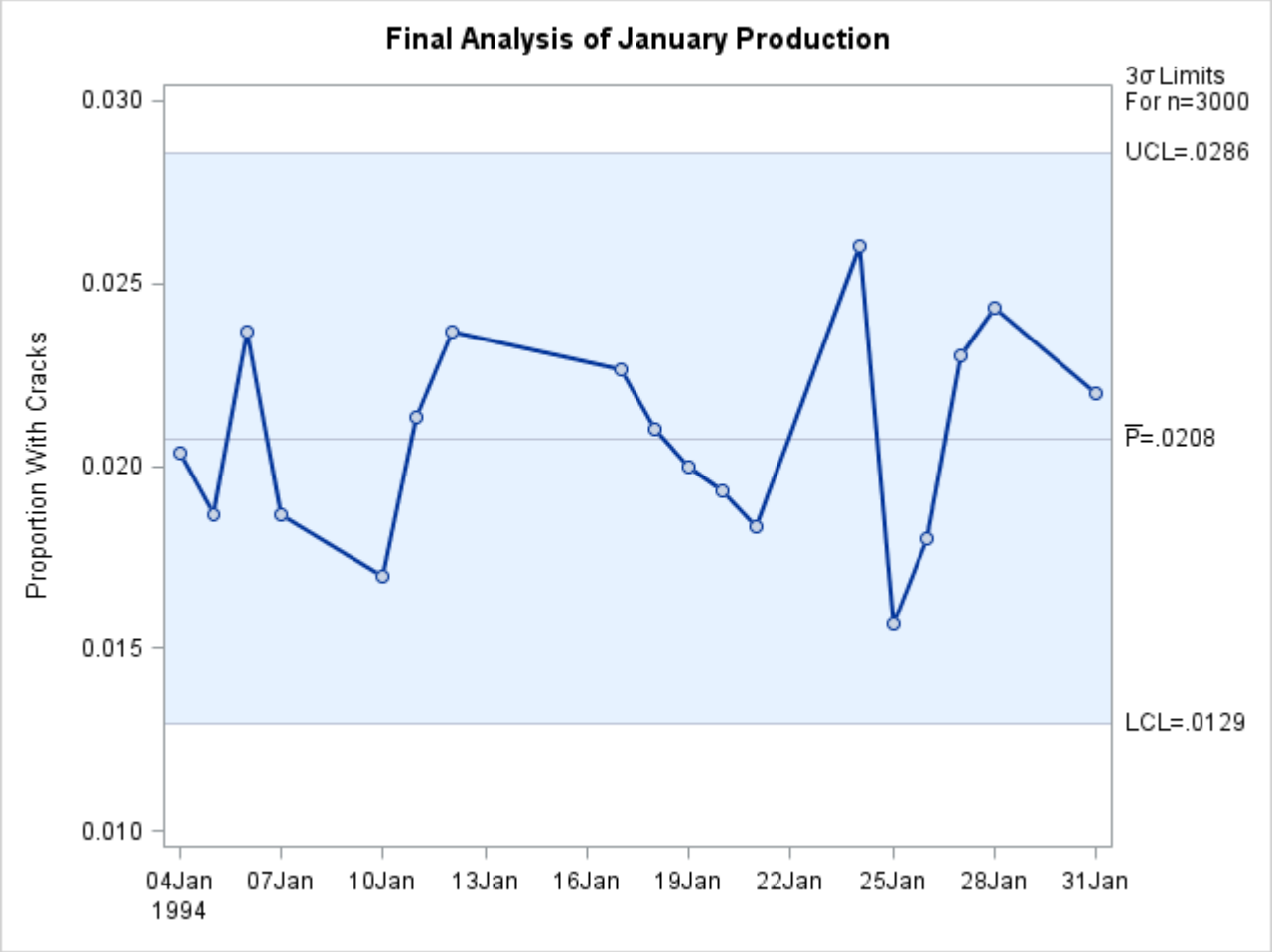


In Figure 19.174, a special cause of variation is signaled by the proportions for January 13 and January 14, which exceed the upper control limit. Because the cause, an improper machine setting, was corrected, it is appropriate to recompute the control limits by excluding the data for these two days. Again, this can be done with a WHERE statement, as follows:

```
title 'Final Analysis of January Production';
proc shewhart data=Bottles;
   where ( Day <= '31JAN94'D ) &
         ( Day ne '13JAN94'D ) &
         ( Day ne '14JAN94'D )   ;
   pchart nCracks * Day / subgroupn = nBottles
                          nohlabel
                          nolegend
                          markers
                          odstitle  = title
                          outlimits = Janlim;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 19.175.

Figure 19.175 Final p Chart for January Data



The data set Janlim, which saves the control limits, is listed in Figure 19.176.

Figure 19.176 Listing of the LIMITS= Data Set Janlim

Final Analysis of January Production

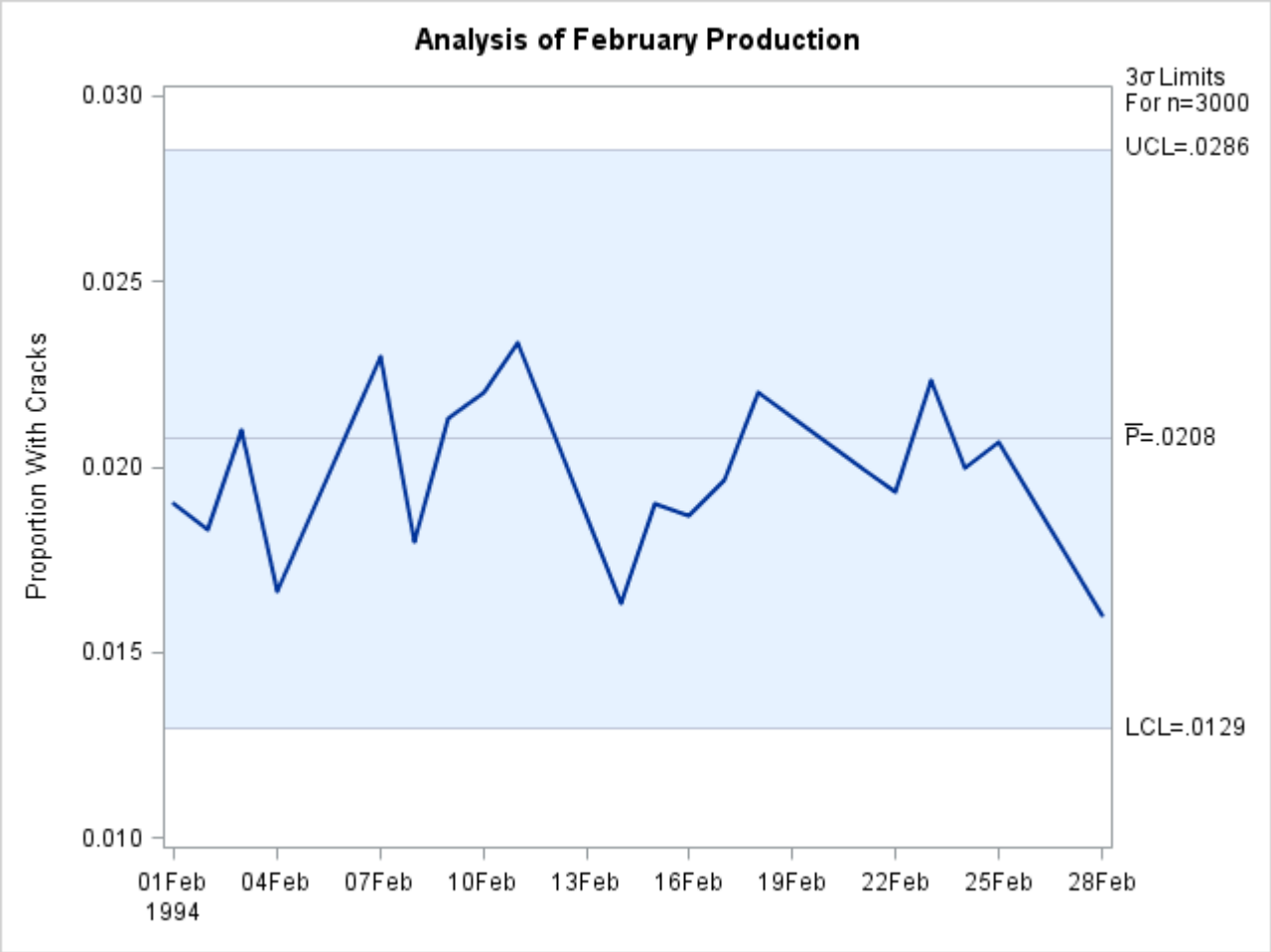
VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
nCracks	Day	ESTIMATE	3000	.002298782	3	0.012950	0.020759	0.028569

Now, the control limits based on the January data are to be applied to the February data. Again, this can be done with a WHERE statement, as follows:

```
title 'Analysis of February Production';
proc shewhart data=Bottles limits=Janlim;
   where Day > '31JAN94'D;
   pchart nCracks * Day / subgroupn = nBottles
                          odstitle  = title
                          nolegend
                          nohlabel;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 19.177.

Figure 19.177 *p* Chart for February Data



Using Switch Variables

NOTE: See *Selecting Subgroups Using Switch Variables* in the SAS/QC Sample Library.

As an alternative to reading a LIMITS= data set and using a WHERE statement, you can provide two special switch variables named _COMP_ and _DISP_ in the input data set. The rules for using these variables are as follows:

- Switch variables must be character variables of length one. Valid values for these variables are 'Y' (or 'y') and 'N' (or 'n'). A blank value is treated as 'Y'.
- Subgroups for which _COMP_ is equal to 'Y' are included in computations of parameter estimates and control limits, and observations for which _COMP_ is equal to 'N' are excluded.
- Subgroups for which _DISP_ is equal to 'Y' are displayed on the chart, and subgroups for which _DISP_ is equal to 'N' are not displayed.

- If the chart statement creates a chart for variables, you can provide two additional switch variables named _COMP2_ and _DISP2_, which are defined similarly to _COMP_ and _DISP_. In this case, the variable _COMP_ specifies which subgroups are used to estimate the process mean μ, and the variable _COMP2_ specifies which subgroups are used to estimate the process standard deviation σ. The variable _DISP_ specifies which subgroups are displayed on the primary chart ($\bar{X}$ chart, median chart, or individual measurements chart), and the variable _DISP2_ specifies which subgroups are displayed on the secondary chart (R chart or s chart).
- The variables _COMP_ and _COMP2_ are not applicable when control limits or control limit parameters are read from a LIMITS= data set.
- The variables _DISP_ and _DISP2_ take precedence over the display controlled by the LIMITN= and ALLN options.
- If the input data set is a DATA= data set with multiple observations per subgroup, switch variable values must be constant within a subgroup.
- Switch variables are saved in OUTHISTORY= and OUTTABLE= data sets. Subgroups for which _DISP_ is equal to 'N' are not saved in an OUTTABLE= data set, and such subgroups are not displayed in tables created with TABLE and related options.

The following statements illustrate how the switch variables _COMP_ and _DISP_ can be used with the bottle production data:

```
data Bottles;
   length _comp_ _disp_ $ 1;
   set Bottles;
   if      Day  = '13JAN94'D then _comp_ = 'n';
   else if Day  = '14JAN94'D then _comp_ = 'n';
   else if Day <= '31JAN94'D then _comp_ = 'y';
   else                           _comp_ = 'n';
   if      Day <= '31JAN94'D then _disp_ = 'n';
   else                           _disp_ = 'y';
run;

title 'Analysis of February Production';
proc shewhart data=Bottles;
   pchart nCracks * Day / subgroupn = nBottles
                          odstitle  = title
                          markers
                          nolegend
                          nohlabel;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is identical to the chart in Figure 19.177.

In general, switch variables are more versatile than WHERE statements in applications where subgroups are simultaneously selected for computation and display. Switch variables also provide a permanent record of which subgroups were selected. The WHERE statement does not alter the input data set; it simply restricts the observations that are read; consequently, the WHERE statement can be more efficient than switch variables for processing large data sets.

Tests for Special Causes: SHEWHART Procedure

This section provides details concerning standard and nonstandard tests for special causes that you can apply with the SHEWHART procedure.

Standard Tests for Special Causes

The SHEWHART procedure provides eight standard *tests for special causes*, also referred to as *rules for lack of control*, *supplementary rules*, *runs tests*, *runs rules*, *pattern tests*, and *Western Electric rules*. These tests improve the sensitivity of the Shewhart chart to small changes in the process. [14] You can also improve the sensitivity of the chart by increasing the rate of sampling, increasing the subgroup sample size, and using control limits that represent less than three standard errors of variation from the central line. However, increasing the sampling rate and sample size is often impractical, and tightening the control limits increases the chances of falsely signaling an out-of-control condition. By detecting particular nonrandom patterns in the points plotted on the chart, the tests can provide greater sensitivity and useful diagnostic information while incurring a reasonable probability of a false signal.

The patterns detected by the eight standard tests are defined in Table 19.100 and Table 19.101, and they are illustrated in Figure 19.178 and Figure 19.179. All eight tests were developed for use with fixed 3σ limits. The tests are indexed according to the numbering sequence used by Nelson (1984, 1985). You can request any combination of the eight tests by specifying the test *indexes* with the TESTS= option in the BOXCHART, CCHART, IRCHART, MCHART, MRCHART, NPCHART, PCHART, UCHART, XCHART, XRCHART, and XSCHART statements.

The following restrictions apply to the tests:

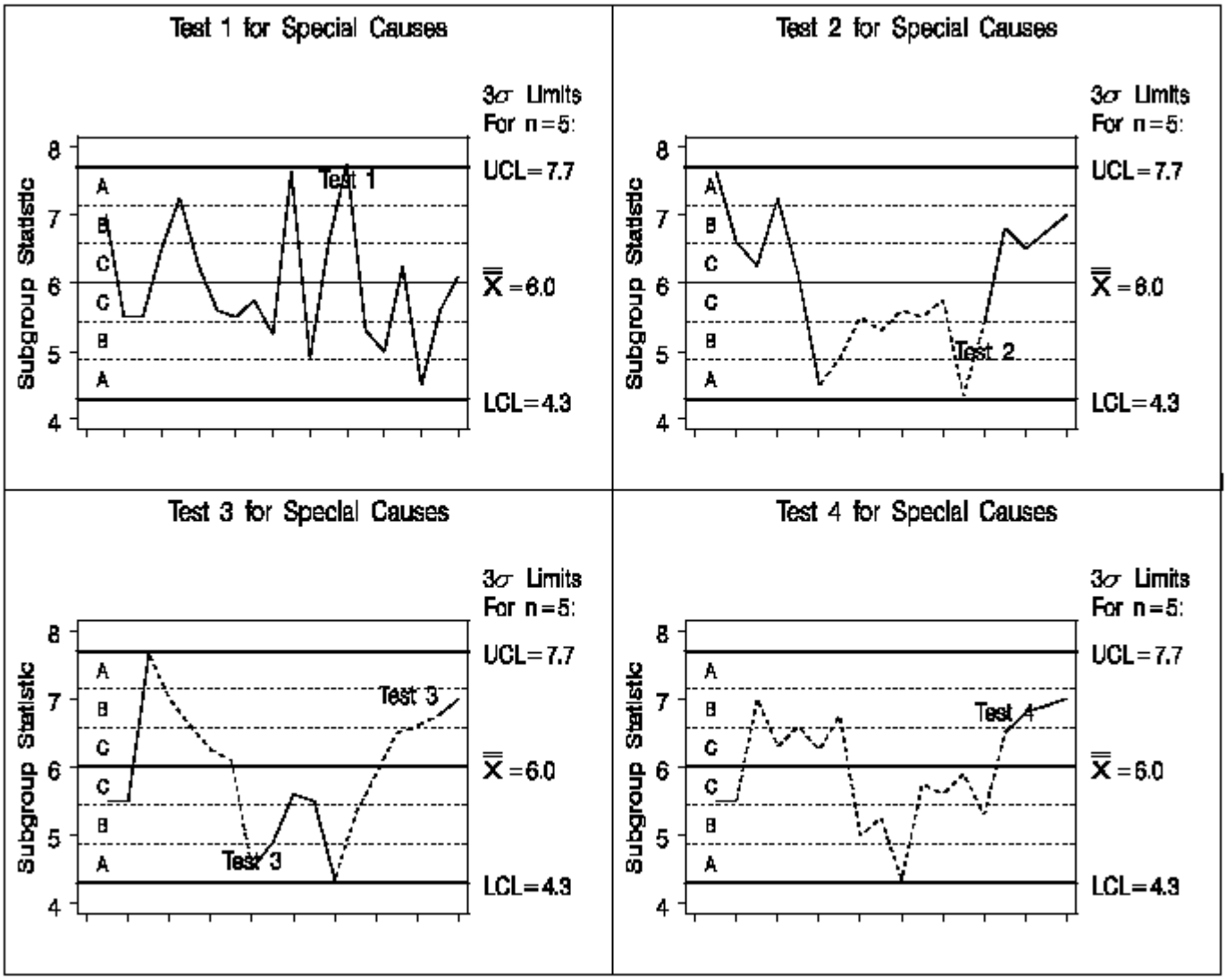
- Only Tests 1, 2, 3, and 4 are recommended for c charts, np charts, p charts, and u charts created with the CCHART, NPCHART, PCHART, and UCHART statements, respectively. In these four cases, Test 2 should not be used unless the process distribution is symmetric or nearly symmetric.
- By default, the TESTS= option is not applied with control limits that are not 3σ limits or that vary with subgroup sample size. You can use the NO3SIGMACHECK option to request tests for special causes when the SIGMAS= option specifies control limits other than 3σ limits. This is not recommended for standard control chart applications, because the standard tests for special causes are based on 3σ limits. You can apply tests for special causes when control limits vary with subgroup sample size by using the LIMITN= or TESTNMETHOD= options (see "Requesting Standard Tests" on page 2122 and "Applying Tests with Varying Subgroup Sample Sizes" on page 2125).

[14] Cumulative sum control charts and moving average control charts also detect small shifts more quickly than an ordinary Shewhart chart. See the sections "PROC CUSUM Statement" on page 549 and "PROC MACONTROL Statement" on page 790 for more information.

Table 19.100 Definitions of Tests 1 to 4

Test Index	Pattern Description
1	One point beyond Zone A (outside the control limits)
2	Nine points in a row in Zone C or beyond on one side of the central line (see Note 1 below)
3	Six points in a row steadily increasing or steadily decreasing (see Note 2 below)
4	Fourteen points in a row alternating up and down

Figure 19.178 Examples of Tests 1 to 4



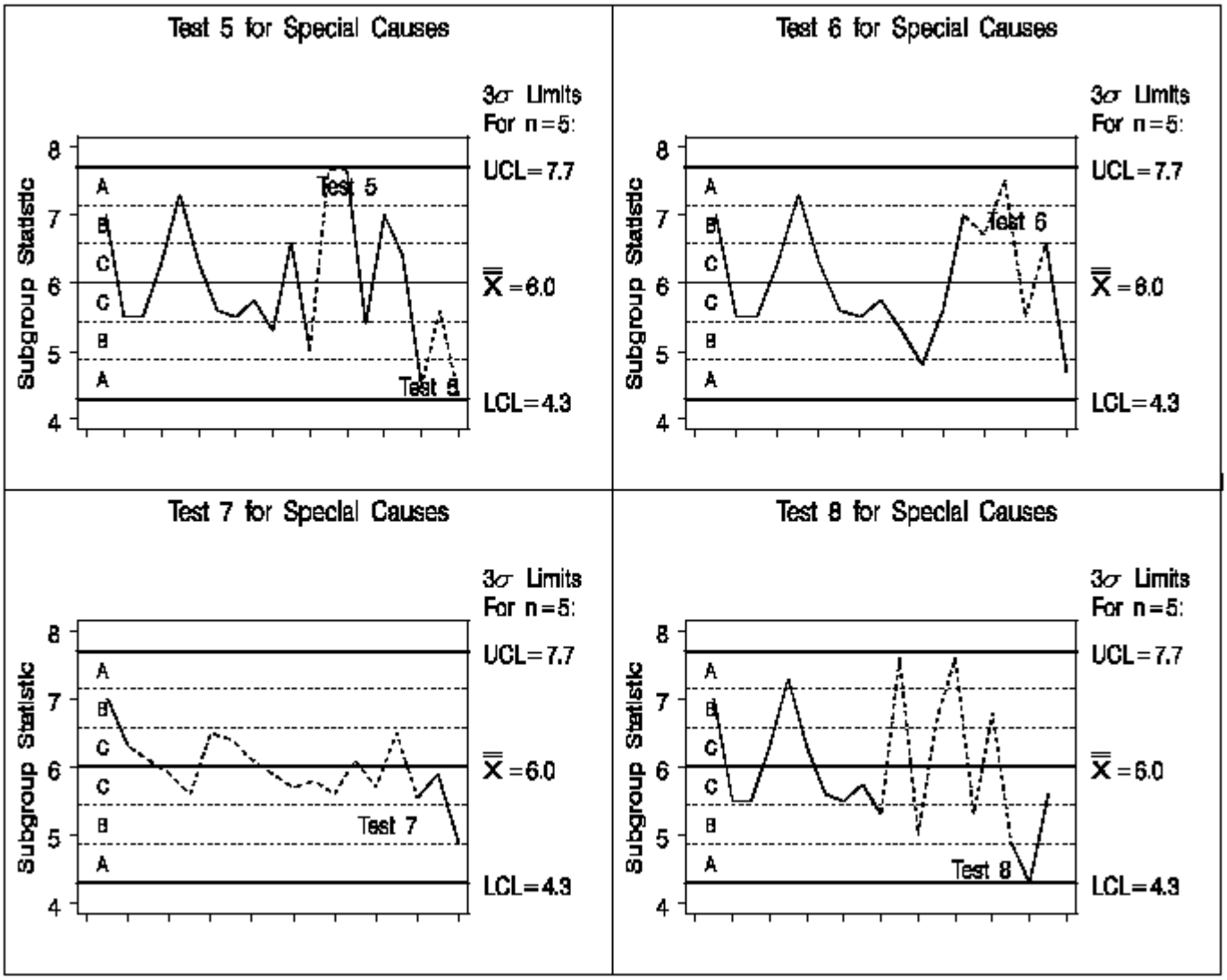
Notes:

1. The number of points in Test 2 can be specified as 7, 8, 9, 11, 14, or 20 with the TEST2RUN= option.
2. The number of points in Test 3 can be specified as 6, 7, or 8 with the TEST3RUN= option.

Table 19.101 Definitions of Tests 5 to 8

Test Index	Pattern Description
5	Two out of three points in a row in Zone A or beyond
6	Four out of five points in a row in Zone B or beyond
7	Fifteen points in a row in Zone C on either or both sides of the central line
8	Eight points in a row on either or both sides of the central line with no points in Zone C

Figure 19.179 Examples of Tests 5 to 8



Requesting Standard Tests

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

The following example illustrates how to request the standard tests for special causes. The tests are applied to an $\bar{X}$ chart for assembly offset measurements whose subgroup means, ranges, and sample sizes are provided by the variables OffsetX, OffsetR, and OffsetN, respectively, in a data set named Assembly. [15]

```
data Assembly;
   length System $ 1 comment $ 16;
   label Sample = 'Sample Number';
   input System Sample OffsetX OffsetR OffsetN comment $16. ;
   datalines;
T   1  19.80  3.8  5
T   2  17.16  8.3  5
T   3  20.11  6.7  5
T   4  20.89  5.5  5
T   5  20.83  2.3  5
T   6  18.87  2.6  5
T   7  20.84  2.3  5
T   8  23.33  5.7  5  New Tool
T   9  19.21  3.5  5
T  10  20.48  3.2  5
T  11  22.05  4.7  5
T  12  20.02  6.7  5
T  13  17.58  2.0  5
T  14  19.11  5.7  5
T  15  20.03  4.1  5
R  16  20.56  3.7  5  Changed System
R  17  20.86  3.3  5
R  18  21.10  5.6  5  Reset Tool
R  19  19.05  2.7  5
R  20  21.76  2.8  5
R  21  21.76  6.4  5
R  22  20.54  4.8  5
R  23  20.04  8.2  5
R  24  19.94  8.8  5
R  25  20.70  5.1  5
Q  26  21.40 12.1  7  Bad Reading
Q  27  21.32  3.2  7
Q  28  20.03  5.2  7  New Gauge
Q  29  22.02  5.9  7
Q  30  21.32  4.3  7
;
```

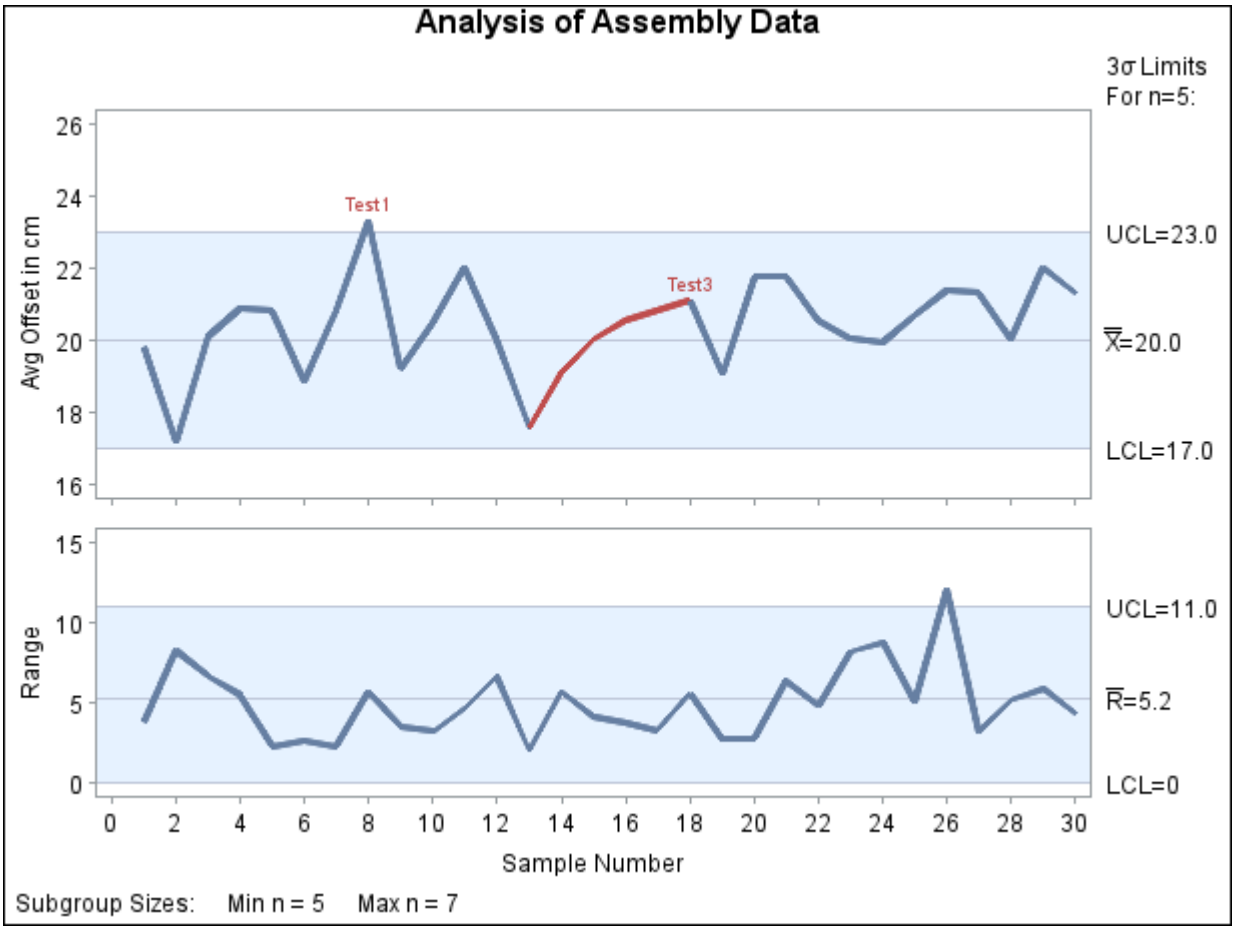
The following statements use the TESTS= option to request Tests 1 to 4. Note that the *process* Offset is specified in the XRCHART statement to indicate that the three summary variables OffsetX, OffsetR, and OffsetN are to be read from the HISTORY= data set Assembly.

[15] The data set Assembly is also used by subsequent examples in this section.

```
ods graphics off;
title 'Analysis of Assembly Data';
proc shewhart history=Assembly;
   xrchart Offset * Sample / mu0      = 20
                             sigma0   = 2.24
                             limitn   = 5
                             alln
                             tests    = 1 to 4
                             vaxis    = 16 to 26 by 2
                             split    = '/';
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The chart is displayed in Figure 19.180. Test 1 is positive at the 8th subgroup, and Test 3 is positive at the 18th subgroup.

Figure 19.180 Standard Tests Using the TESTS= Option



The control limits in Figure 19.180 are based on standard values for the process mean and standard deviation specified with the MU0= and SIGMA0= options, respectively. Although the subgroup sizes vary, fixed control limits are displayed corresponding to a nominal sample size of five, which is specified with the LIMITN= option. Because ALLN is specified, points are displayed for all subgroups, regardless of sample size.

NOTE: If the LIMITN= option were not specified, the control limits would vary with subgroup sample size, and by default the tests would not be applied. An alternative method for applying the tests with varying subgroup sample sizes is discussed in "Applying Tests with Varying Subgroup Sample Sizes" on page 2125.

Interpreting Standard Tests for Special Causes

Nelson (1984, 1985) makes the following comments concerning the interpretation of the tests:

- When a process is in statistical control, the chance of a false signal for each test is less than five in one thousand.
- Test 1 is positive if there is a shift in the process mean, if there is an increase in the process standard deviation, or if there is a "single aberration in the process such as a mistake in calculation, an error in measurement, bad raw material, a breakdown of equipment, and so on" (Nelson 1985).
- Test 2 signals a shift in the process mean. The use of nine points (rather than seven as in (Grant and Leavenworth 1988) for the pattern that defines Test 2 makes the chance of a false signal comparable to that of Test 1. (To control the number of points for the pattern in test 2, use the TEST2RUN= option in the chart statement.)
- Test 3 signals a drift in the process mean. Nelson (1985) states that causes can include "tool wear, depletion of chemical baths, deteriorating maintenance, improvement in skill, and so on."
- Test 4 signals "a systematic effect such as produced by two machines, spindles, operators or vendors used alternately" (Nelson 1985).
- Tests 1, 2, 3, and 4 should be applied routinely; the combined chance of a false signal from one or more of these tests is less than one in a hundred. Nelson (1985) describes these tests as "a good set that will react to many commonly occurring special causes."
- In the case of charts for variables, the first four tests should be augmented by Tests 5 and 6 when earlier warning is desired. The chance of a false signal increases to two in a hundred.
- Tests 7 and 8 indicate stratification (observations in a subgroup have multiple sources with different means). Test 7 is positive when the observations in the subgroup always have multiple sources. Test 8 is positive when the subgroups are taken from one source at a time.

Nelson (1985) also comments that "the probabilities quoted for getting false signals should not be considered to be very accurate" because the probabilities are based on assumptions of normality and independence that might not be satisfied. Consequently, he recommends that the tests "should be viewed as simply practical rules for action rather than tests having specific probabilities associated with them." Nelson cautions that "it is possible, though unlikely, for a process to be out of control yet not show any signals from these eight tests."

Modifying Standard Tests for Special Causes

Some textbooks and references present slightly different versions of Tests 2 and 3. You can use the following options to request these modifications:

- TEST2RUN=*run-length* specifies the length of the pattern for Test 2. The form of the test for each *run-length* is given in the following table. The default *run-length* is 9.

Run-length	Number of Points on One Side of Central Line
7	7 in a row
8	8 in a row
9	9 in a row
11	at least 10 out of 11 in a row
14	at least 12 out of 14 in a row
20	at least 16 out of 20 in a row

- TEST3RUN=*run-length* specifies the length of the pattern for Test 3. The *run-length* values allowed are 6, 7, and 8. The default *run-length* is 6.

The Western Electric Company (now AT&T) *Statistical Quality Control Handbook* (1956) and Montgomery (1996) discuss a test that is signaled by eight points in a row in Zone C or beyond (on one side of the central line). You can request this test by specifying TESTS=2 and TEST2RUN=8. The *Handbook* also discusses tests corresponding to Tests 1, 5, 6, 7, and 8.

Kume (1985) recommends a number of tests for special causes that can be regarded as modifications of Tests 2 and 3:

- seven points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=7.
- at least 10 out of 11 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=11.
- at least 12 out of 14 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=14.
- at least 16 out of 20 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=20.
- seven points in a row steadily increasing or decreasing. Specify TESTS=3 and TEST3RUN=7.

Applying Tests with Varying Subgroup Sample Sizes

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

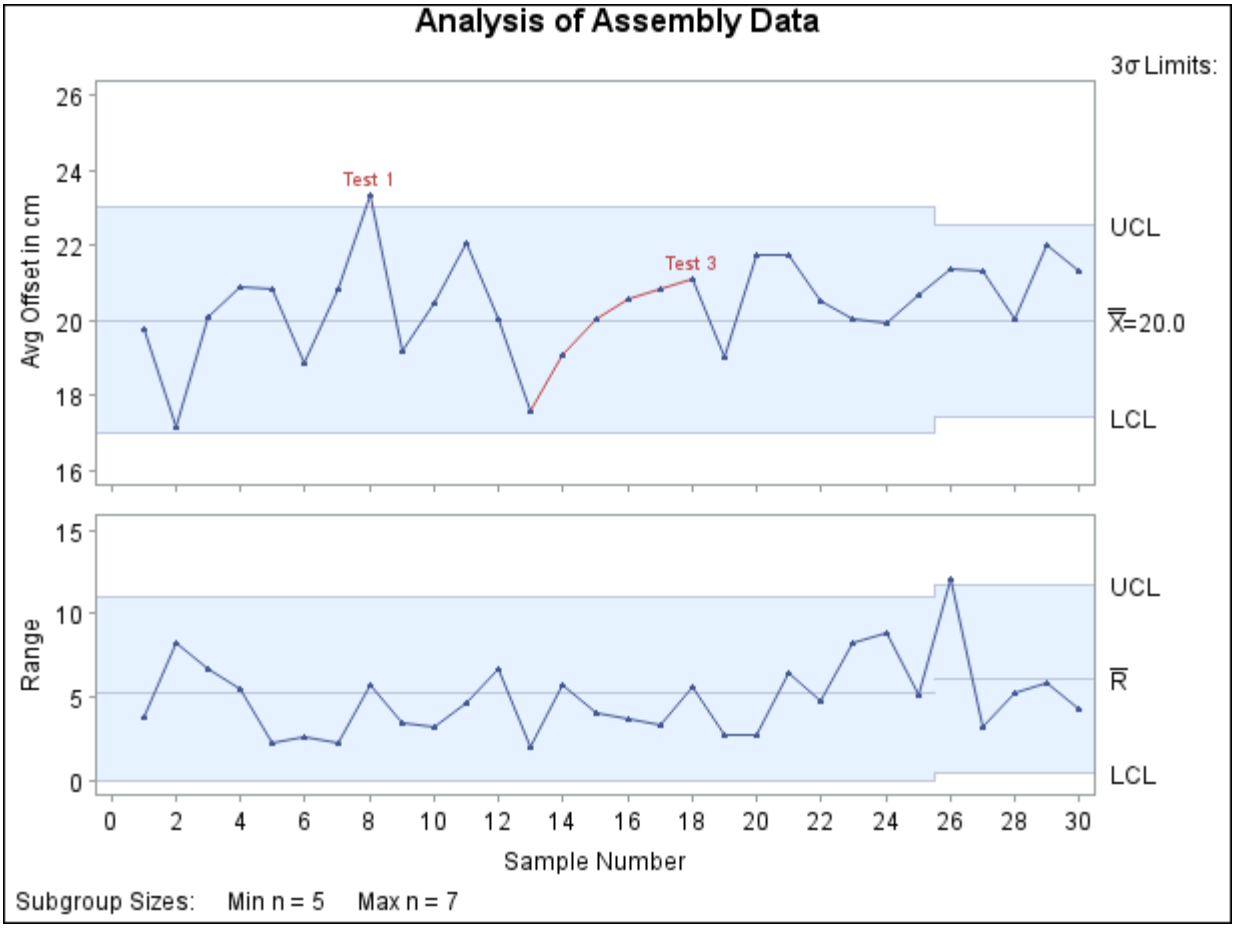
Nelson (1989, 1994) describes the use of standardization to apply the tests for special causes to data involving varying subgroup samples. This approach applies the tests to the standardized subgroup statistics, setting the control limits at ± 3 and the zone boundaries at ± 1 and ± 2. For instance, for an $\bar{X}$ chart with subgroup means $\bar{X}_i$ and varying subgroup sample sizes n_i, the tests are applied to the standardized values $z_i = (\bar{X}_i - \overline{\overline{X}})/(s/\sqrt{n_i})$, where $\overline{\overline{X}}$ estimates the process mean, and s estimates the process standard deviation. You can request this method with the TESTNMETHOD= option,[16] as illustrated by the following statements:

[16] If the TESTNMETHOD= option were omitted in this example, the tests would not be applied, and a warning message would be displayed in the SAS log.

```
ods graphics off;
title 'Analysis of Assembly Data';
symbol v=dot;
proc shewhart history=Assembly;
   xrchart Offset * Sample / mu0         = 20
                             sigma0      = 2.24
                             tests       = 1 to 4
                             testnmethod = standardize
                             testlabel   = space
                             vaxis       = 16 to 26 by 2
                             wtests      = 1
                             split       = '/';
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

Here the tests are applied to $z_i = (\bar{X}_i - 20)/(2.24/\sqrt{n_i})$. The chart, shown in Figure 19.181, displays the results of the tests on a plot of the *unstandardized* means.

Figure 19.181 The TESTNMETHOD= Option for Varying Subgroup Sizes

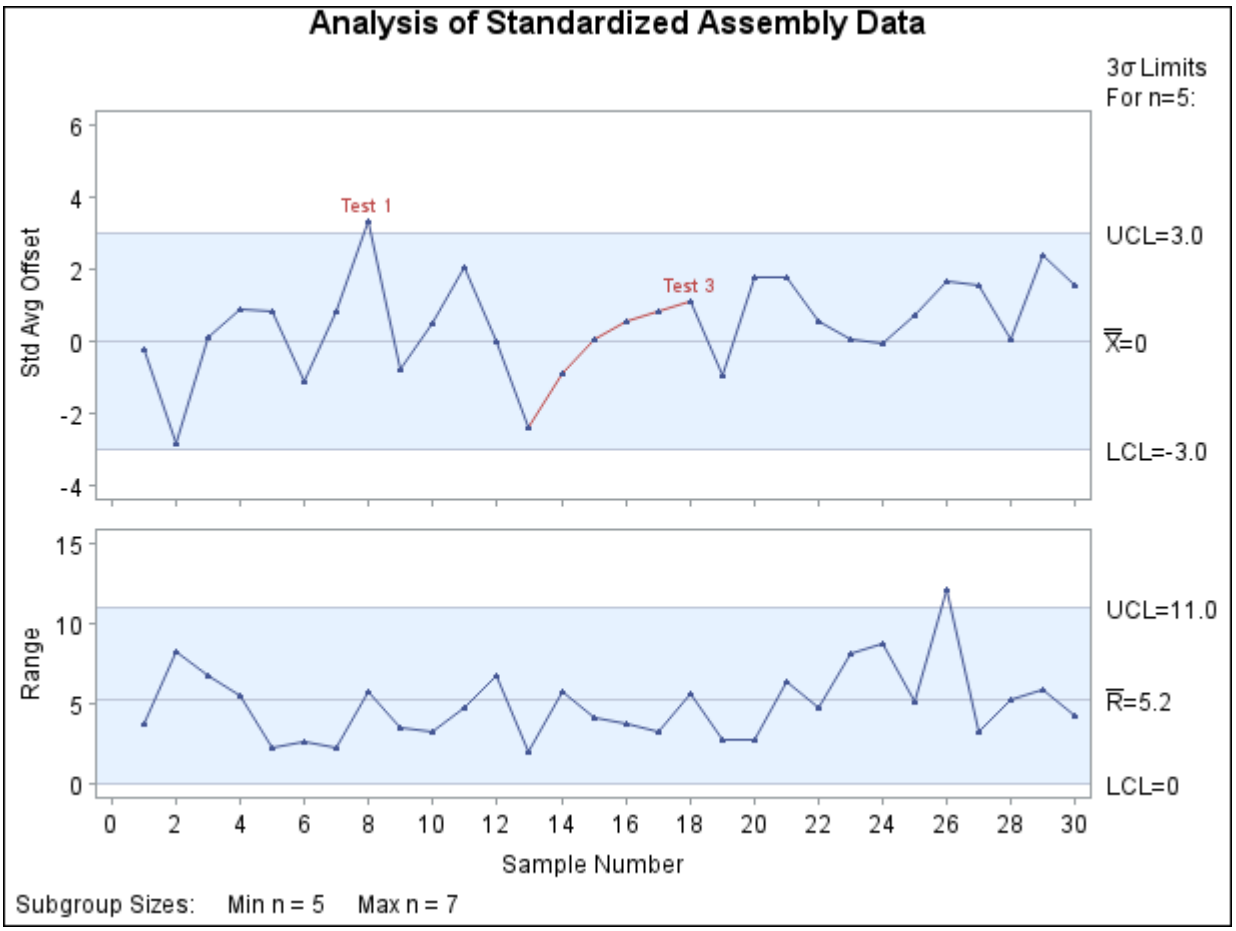


The following statements create an equivalent chart that plots the *standardized* means:

```
data Assembly;
   set Assembly;
   zx = ( OffsetX - 20 ) / ( 2.24 / sqrt( OffsetN ) );
run;

ods graphics off;
title 'Analysis of Standardized Assembly Data';
symbol v=dot;
proc shewhart
   history=Assembly (rename = (OffsetR=zr OffsetN=zn));
   xrchart z * Sample / mu0         = 0
                        sigma0      = 2.2361 /* sqrt 5 */
                        limitn      = 5
                        alln
                        tests       = 1 to 4
                        testlabel   = space
                        vaxis       = -4 to 6 by 2
                        wtests      = 1
                        split = '/';
   label zx = 'Std Avg Offset/Range';
run;
```

Here, the SIGMA0= value is the square root of the LIMITN= value. The chart is shown in Figure 19.182.

Figure 19.182 Tests with Standardized Means



NOTE: In situations where the standard deviation is estimated from the data and the subgroup sample sizes vary, you can use the SMETHOD= option to request various estimation methods, including the method of Burr (1969).

Labeling Signaled Points with a Variable

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

If a test is signaled at a particular point, the point is labeled by default with the *index* of the test, as illustrated in Figure 19.180.[17] You can use the TESTLABEL= option to specify a variable in the input data set whose *values* provide the labels, as illustrated by the following statements:

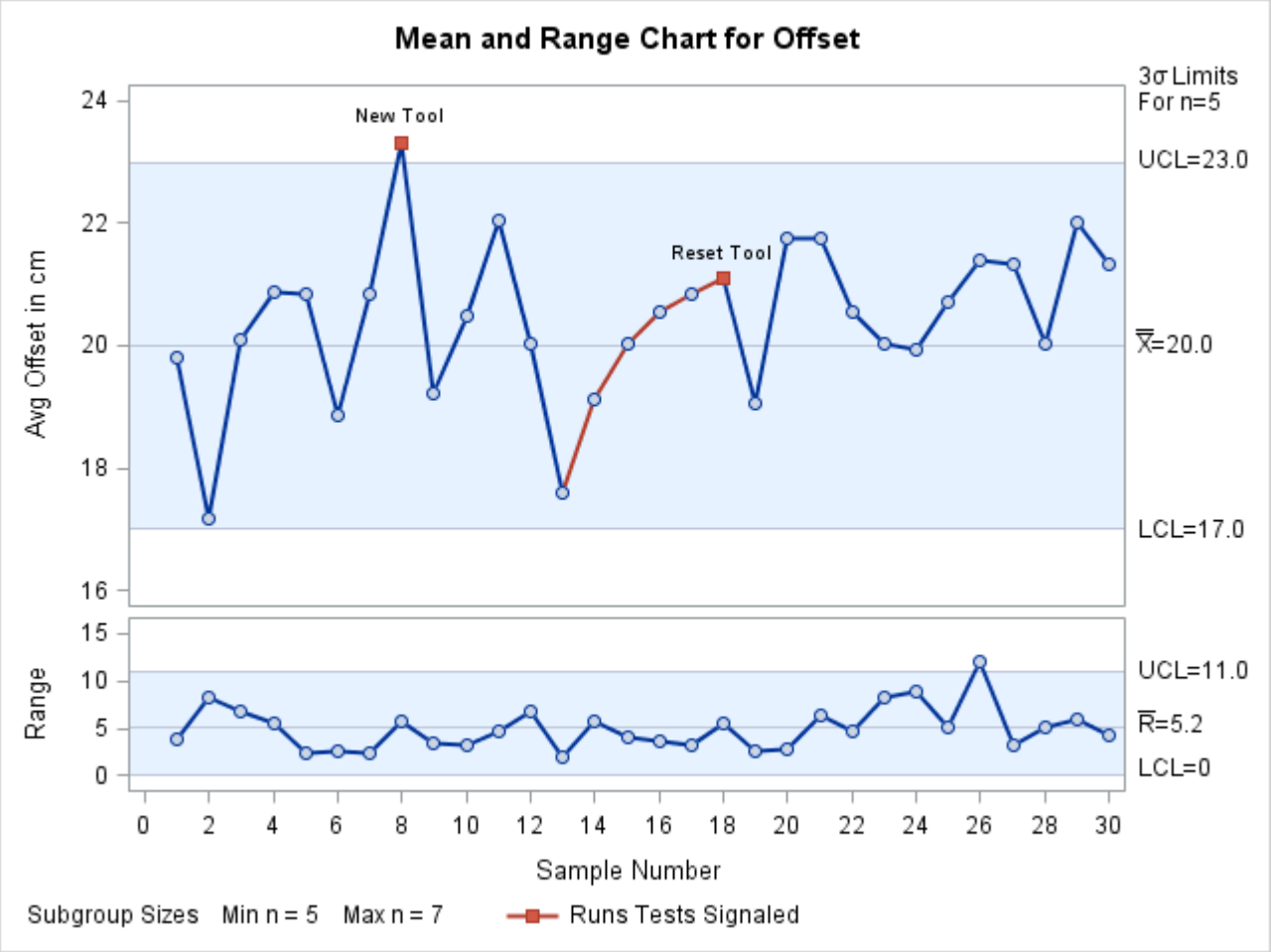
```
ods graphics on;
title 'Analysis of Assembly Data';
proc shewhart history=Assembly;
   xrchart Offset * Sample / mu0       = 20
                             sigma0    = 2.24
                             limitn    = 5
                             alln
```

[17] If two or more tests are positive at a particular point, the default label identifies the *index* of the test that was specified first with the TESTS= option.

```
                                tests     = 1 to 4
                                testlabel = ( comment )
                                vaxis     = 16 to 24 by 2
                                split      = '/'
                                markers;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The labels are shown in Figure 19.183. It is often helpful to specify a variable with the TESTLABEL= option that provides operator comments or other information that can aid in the identification of special causes.

Figure 19.183 Labeling Points with a TESTLABEL= Variable



Applying Tests with Multiple Phases

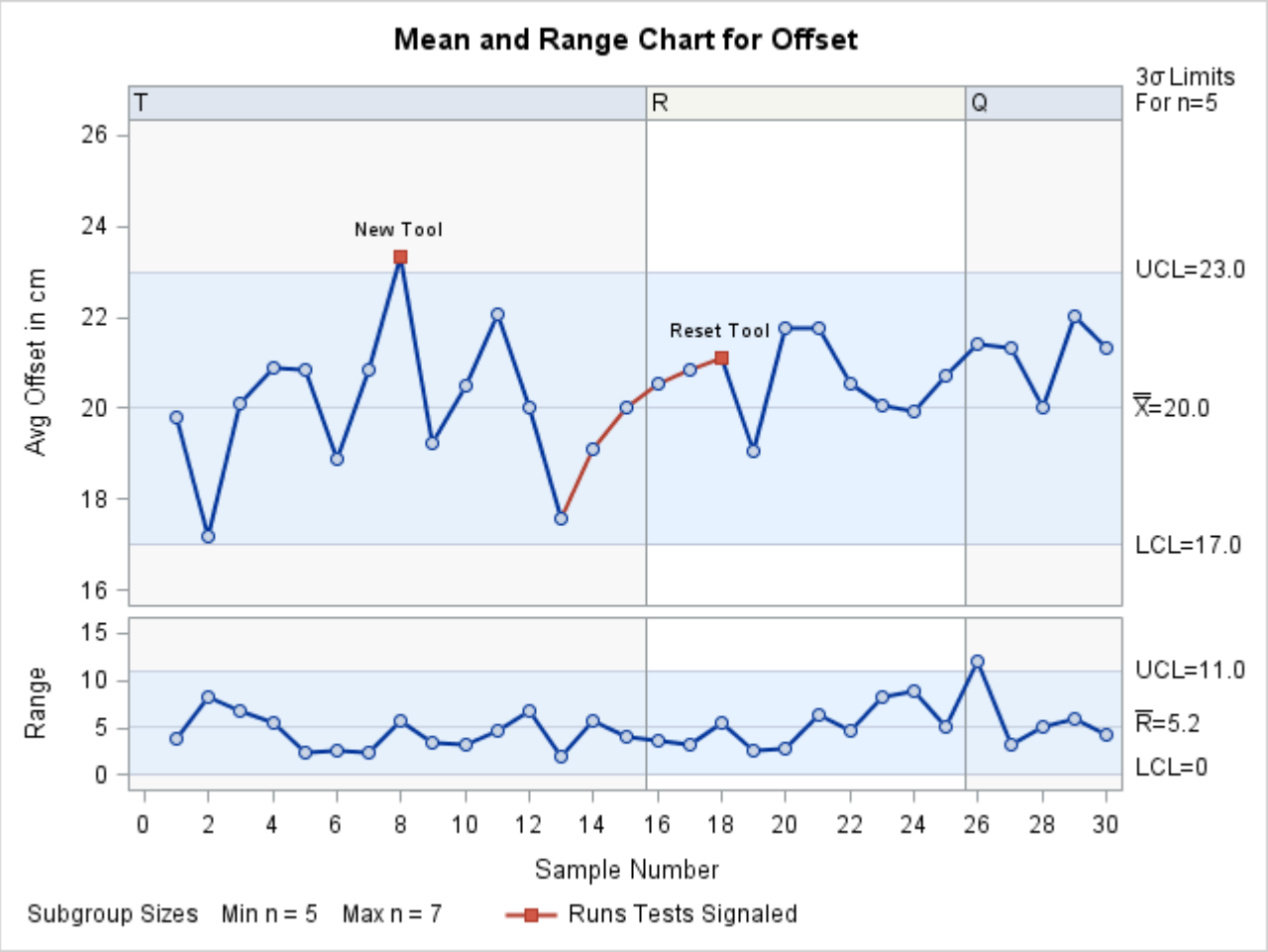
NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

The data set Assembly includes a variable named System, which indicates the manufacturing system used to produce each assembly. As shown by the following statements, this variable can be temporarily renamed and read as the variable _PHASE_ to create a control chart that displays the *phases* (groups of consecutive subgroups) for which System is equal to 'T', 'R', and 'Q':

```
ods graphics on;
title 'Manufacturing Systems Used in Assembly';
proc shewhart
   history=Assembly (rename=(System=_phase_));
   xrchart Offset * Sample /
      mu0          = 20
      sigma0       = 2.24
      limitn       = 5
      alln
      tests        = 1 to 4
      testlabel    = ( comment )
      readphases   = ('T' 'R' 'Q')
      phaselegend
      phaseref
      vaxis        = 16 to 26 by 2
      split        = '/'
      markers;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The chart is shown in Figure 19.184.

Figure 19.184 Single Set of Limits with Multiple Phases



Note that a single set of fixed 3σ limits is displayed for all three phases because LIMITN=5 and ALLN are specified. Consequently, the tests requested with the TESTS= option are applied independently of the phases. In general, however, it is possible to display distinct sets of control limits for different phases, and in such situations, the tests are not applied independently of phases, as discussed in the next example.

Applying Tests with Multiple Sets of Control Limits

NOTE: See *Applying Tests with Multiple Control Limits* in the SAS/QC Sample Library.

This example is a continuation of the previous example, except that distinct control limits are displayed for each of the phases determined by the variable System. The control limit parameters (mean, standard deviation, and nominal sample size) for each phase (manufacturing system) are provided in the following data set:

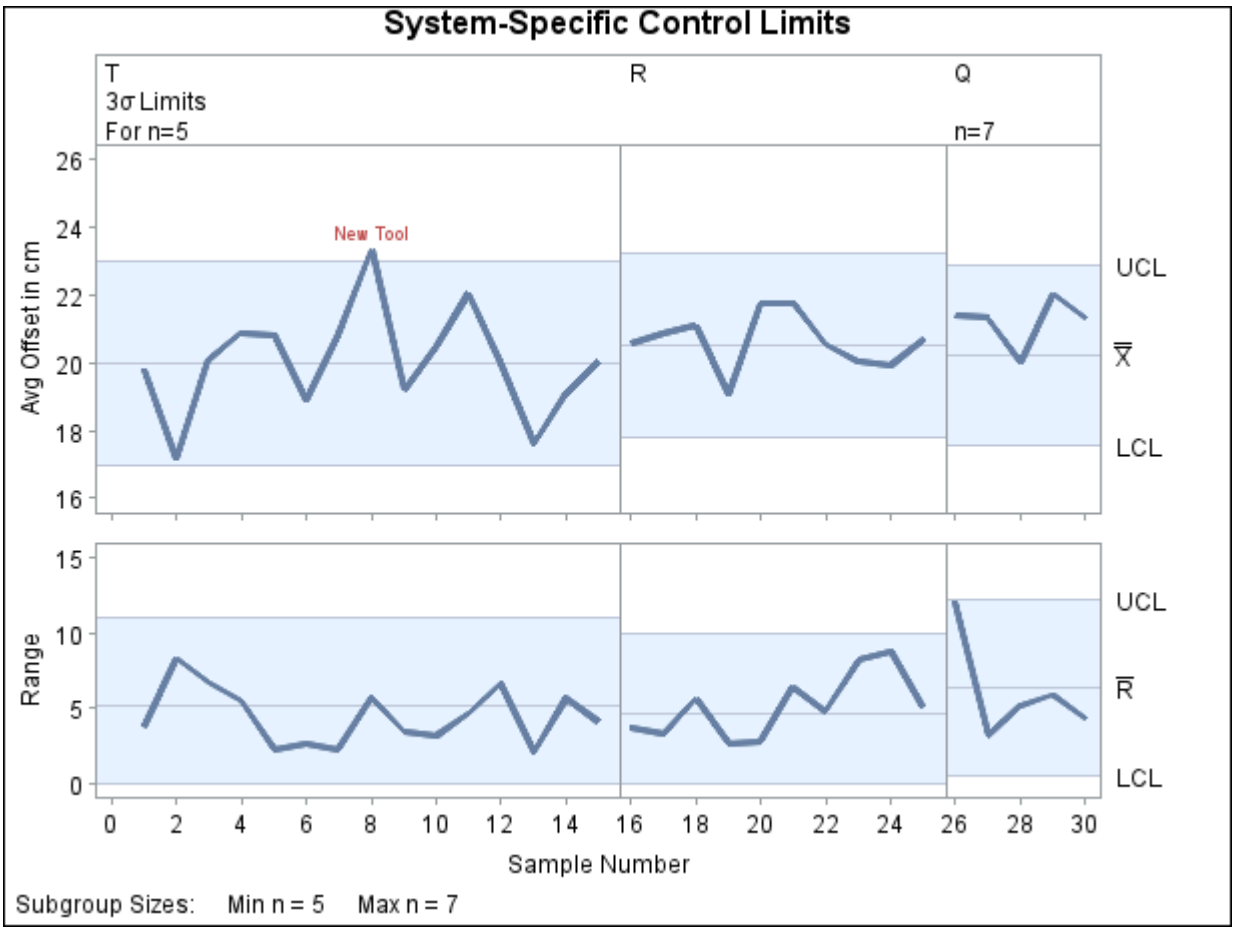
```
data Syslim;
   length _var_ $8 _subgrp_ $8 _type_ $8 _index_ $1;
   input _var_ _subgrp_ _index_ _type_ _mean_ _stddev_
         _limitn_ _sigmas_;
   datalines;
Offset Sample R standard 20.5 2.02 5 3
Offset Sample Q standard 20.2 2.35 7 3
Offset Sample T standard 20.0 2.24 5 3
;
```

The following statements read the control limit parameters from Syslim and use the READPHASES= and READINDEXES= options to display a distinct set of control limits for each phase:

```
ods graphics off;
title 'System-Specific Control Limits';
proc shewhart
   limits=Syslim
   history=Assembly (rename=(System=_phase_));
   xrchart Offset * Sample /
      tests       = 1 to 4
      testlabel   = ( comment )
      readindexes = ('T' 'R' 'Q')
      readphases  = ('T' 'R' 'Q')
      phaselegend
      phaseref
      phasebreak
      vaxis       = 16 to 26 by 2
      split       = '/' ;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The chart is shown in Figure 19.185. The tests requested with the TESTS= option are applied strictly within the phases, because the control limits are not constant across the phases (as in Figure 19.184). In particular, note that the pattern labeled *Reset Tool* in Figure 19.184 is not detected in Figure 19.185.

Figure 19.185 Multiple Sets of Control Limits



In most applications involving multiple control limits, a known change or improvement has occurred at the beginning of each phase; consequently, it is appropriate to restart the tests at the beginning of each phase rather than search for patterns that span the boundaries of consecutive phases. In these situations, the PHASEBREAK option is useful for suppressing the connection of points from one phase to the next. Note that it is not necessary to specify the TESTNMETHOD= option here because the subgroup sample sizes are constant within each phase.

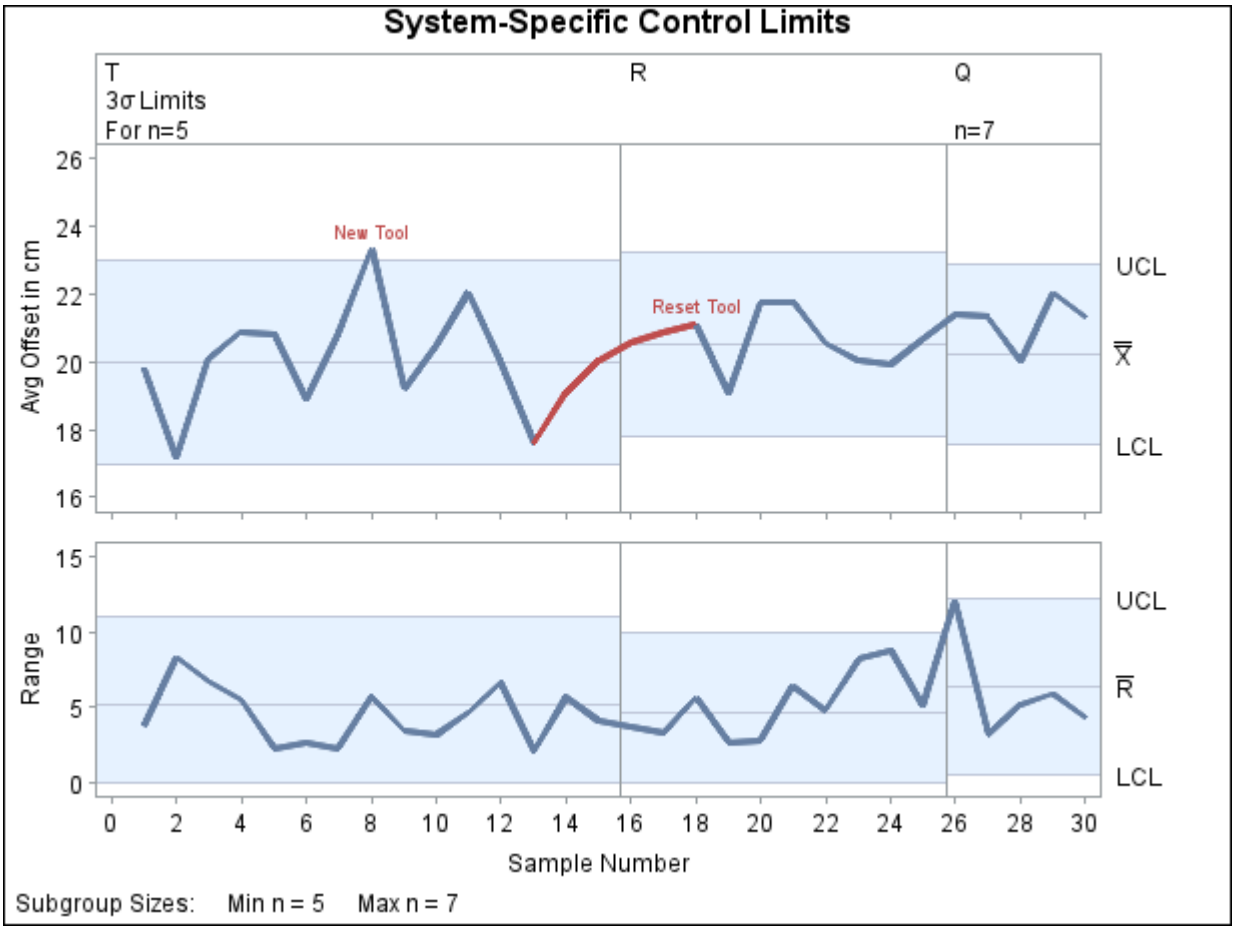
There might be applications in which it is appropriate to apply the tests across phase boundaries. You can use the TESTACROSS option to request this behavior.

```
ods graphics off;
title 'System-Specific Control Limits';
proc shewhart
   limits=Syslim
   history=Assembly (rename=(System=_phase_));
   xrchart Offset * Sample /
      tests       = 1 to 4
      testlabel   = ( comment )
      testnmethod = standardize
      testacross
      readindexes = ('T' 'R' 'Q')
```

```
      readphases  = ('T' 'R' 'Q')
      phaselegend
      phaseref
      vaxis       = 16 to 26 by 2
      split       = '/';
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The chart created with the TESTACROSS option is displayed in Figure 19.186.

Figure 19.186 Multiple Sets of Control Limits with the TESTACROSS Option



Here, it is necessary to specify TESTNMETHOD=STANDARDIZE in conjunction with the TESTACROSS option, because the subgroup sample sizes are not constant across phases.

Although Test 3 is now signaled at sample 18, this result should be interpreted with care because the test is applied to standardized average offsets, and the averages for samples 13, 14, and 15 are standardized differently than the averages for samples 16, 17, and 18. If, for instance, the value of _MEAN_ for phase 'R' in Syslim were 21.0 rather than 20.5, the standardized mean for sample 16 would be less than the standardized mean for sample 15, and Test 3 would not be signaled at sample 18.

In summary, when working with multiple control limits, you should

- use the TESTACROSS option only if the process is operating in a continuous manner across phases
- use TESTNMETHOD=STANDARDIZE only if it is clearly understood by users that tests signaled on the chart are based on *standardized* statistics rather than the plotted statistics

Enhancing the Display of Signaled Tests

There are various options for labeling points at which a test is signaled.

- The default label for Test *i* is *Testi*. See Figure 19.180 for an example.
- Specify TESTLABEL=SPACE to request labels of the form *Test i*. See Figure 19.181 for an example.
- Specify TESTLABEL*i*='*label*' to provide a specific *label* for the *i*th test. See Figure 19.191 for an example.
- Specify TESTLABEL=(*variable*) to request labels provided by a *variable* in the input data set. See Figure 19.183 for an example.

If two or more tests are signaled at a particular point, the label displayed corresponds to the test that was specified first in the TESTS= list.

If you are producing traditional graphics, you can specify the color of the label and the connecting line segments for the pattern with the CTESTS= option. You can specify the line type for the line segments with the LTESTS= option. If you are creating line printer charts, you can specify the plot character for the line segments with the TESTCHAR= option.

You can specify the ZONES option to display the zone lines on the chart, and you can specify ZONELABELS to label the zone lines. If you are creating traditional graphics, you can specify the color of the lines with the CZONES= option, and if you are producing line printer charts, you can specify the plot character for the lines with the ZONECHAR= option.

Nonstandard Tests for Special Causes

This section describes options and programming techniques for requesting various nonstandard tests for special causes.

Applying Tests to Range and Standard Deviation Charts

NOTE: See *Applying Tests for Special Causes-R charts* in the SAS/QC Sample Library.

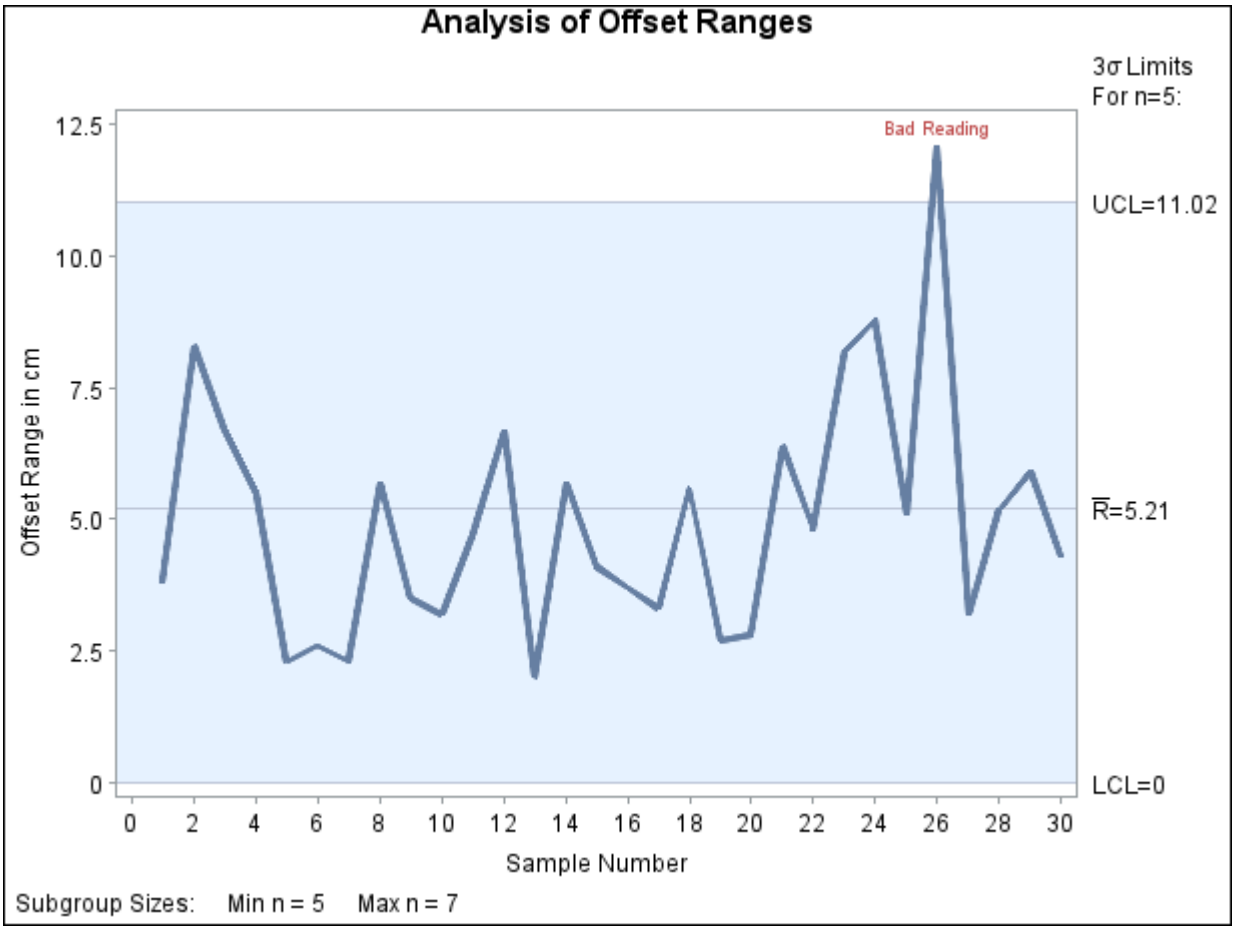
If you are using the MRCHART, RCHART, SCHART, XRCHART, or XSCHART statement, you can use the TESTS2= option to request tests for special causes with an *R* chart or *s* chart. The syntax and test definitions for the TESTS2= option are identical to those for the TESTS= option, and you can use the ZONES2 and ZONE2LABELS options to display the zones on the secondary chart.

The following statements request Test 1 for a range chart of the data in Assembly (see "Requesting Standard Tests" on page 2122):

```
ods graphics off;
title 'Analysis of Offset Ranges';
proc shewhart history=Assembly;
   rchart Offset * Sample / sigma0    = 2.24
                            limitn    = 5
                            alln
                            tests2    = 1
                            testlabel = (comment) ;
   label OffsetR = 'Offset Range in cm';
run;
```

The *R* chart is shown in Figure 19.187.

Figure 19.187 Range Chart with Test 1



CAUTION: Except for requesting Test 1, use of the TESTS2= option is not recommended for general process control work. At the time of this writing, there is insufficient published research supporting the application of the other tests to *R* charts and *s* charts. There are no established guidelines for interpreting the other tests, nor are there assessments of their false signal probabilities or average run length characteristics. The TESTS2= option is intended primarily as a research tool.

Applying Tests Based on Generalized Patterns

In addition to *indices* for standard tests, you can specify up to eight T-patterns, M-patterns, or S-patterns with the TESTS= option:

- Specifying a T-pattern requests a search for k out of m points in a row in the interval (a, b). Tests based on T-patterns are generalizations of Tests 1, 2, 5, and 6. The average run length properties of tests based on T-patterns have been analyzed by Champ and Woodall (1987). Also refer to Chapter 8 of Wetherill and Brown (1991).
- Specifying an M-pattern requests a search for k points in a row increasing or decreasing. Tests based on M-patterns are generalizations of Test 3.
- Specifying an S-pattern requests a search for a statistically significant linear trend. Tests based on S-patterns are not generalizations of any standard test for special causes. Instead, a parametric or nonparametric test for a linear trend is applied over a window of k data points.

The general syntax for a T-pattern is of the form

T(K=*k* **M=***m* **LOWER=***a* **UPPER=***b* **SCHEME=***scheme* **CODE=***character* **LABEL=***'label'* **LEGEND=***legend* **)**

The options for a T-pattern are summarized in the following table:

Option	Description
K=*k*	Number of points ($k \leq m$)
M=*m*	Number of consecutive points
LOWER=*value*	Lower limit of interval (a, b)
UPPER=*value*	Upper limit of interval (a, b)
SCHEME=ONESIDED	One-sided scheme using (a, b)
SCHEME=TWOSIDED	Two-sided scheme using $(a, b) \cup (-b, -a)$
CODE=*character*	Identifier for test (A–H)
LABEL=*'label'*	Label for points that are signaled
LEGEND=*'legend'*	Legend used with the TABLELEGEND option

The following rules apply to the T-pattern options:

1. You must specify SCHEME=*scheme*. Specifying SCHEME=ONESIDED requests a one-sided test that searches for k out of m points in a row in the interval (a, b). Specifying SCHEME=TWOSIDED with positive values for a and b (where $a < b$) requests a two-sided test that searches for k out of m points in a row in the interval (a, b) or k out of m points in a row in the interval $(-b, -a)$.
2. The values a and b must be specified in standardized units, and they must both have the same sign. For instance, specifying LOWER=2 and UPPER=3 with SCHEME=TWOSIDED corresponds to Zone A in Figure 19.178.
3. Specifying a missing value for the LOWER= option and a negative value for b requests a search in the interval $(-\infty, b)$. Specifying a positive value for a and a missing value for the UPPER= option requests a search in the interval (a, ∞).

4. You must specify a CODE=*character*, which can be any of the letters A through H. The character identifies the pattern in tables requested with the TABLETESTS and TABLEALL options and in the value of the variable _TESTS_ in the OUTTABLE= data set. The character is analogous to the indices 1 through 8 that are used to identify the standard tests. If you request multiple T-patterns, you must specify a unique character for each pattern.

5. You can specify a *label* with the LABEL= option. The label must be enclosed in quotation marks and can be up to 16 characters long. The label is used to label points on the chart at which the test defined by the T-pattern is signaled. The LABEL= option is similar to the TESTLABEL*n*= options used with the standard tests.

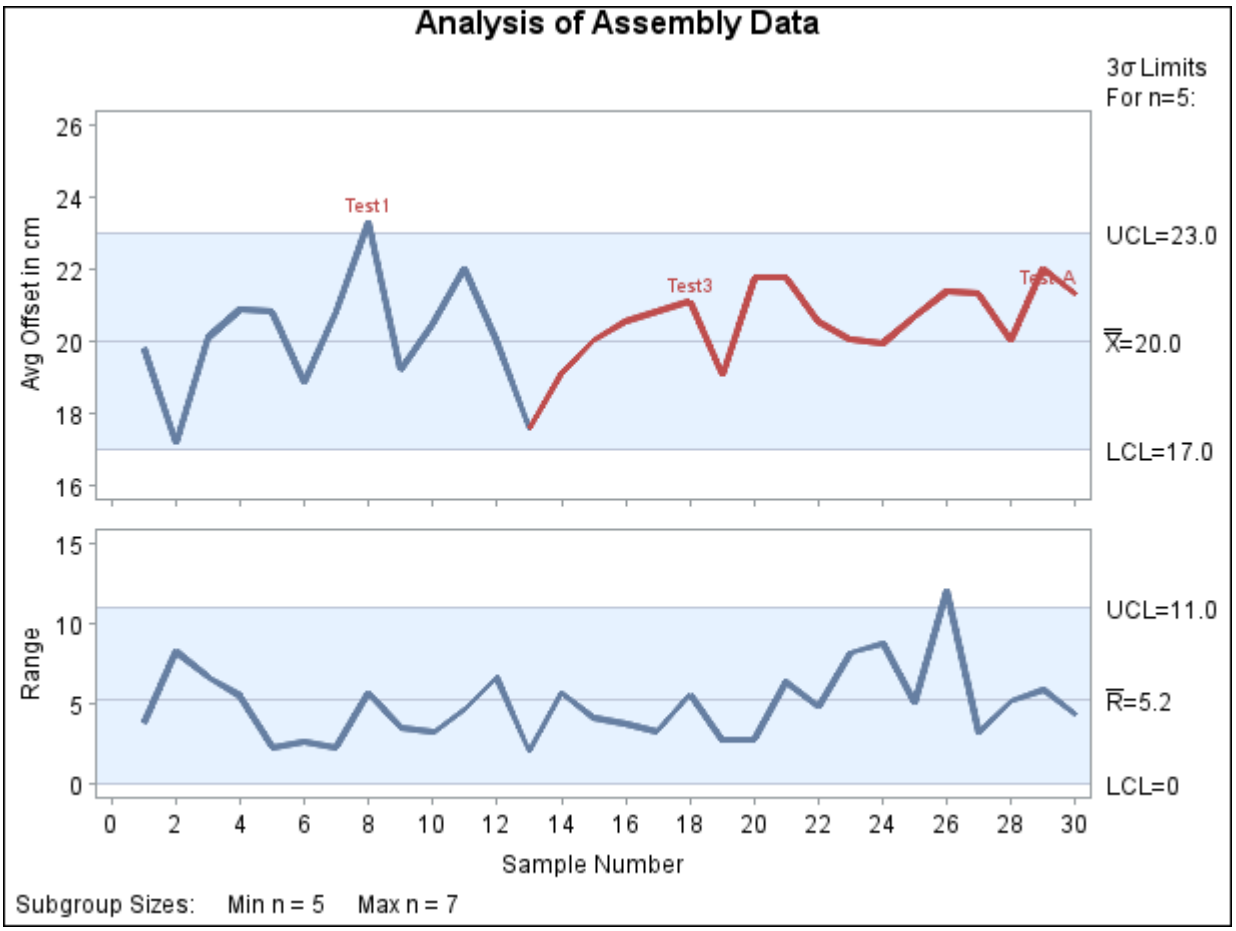
6. You must specify a *legend* with the LEGEND= option if you also specify the TABLELEGEND or TABLEALL option. The legend must be enclosed in quotation marks and can be up to 40 characters long. The legend is used to describe the test defined by the T-pattern in the table legend requested with the TABLELEGEND and TABLEALL options.

NOTE: See *Applying Tests Based on General Patterns* in the SAS/QC Sample Library.

An example of a nonstandard test using a T-pattern is the run test based on 14 out of 17 points in a row on the same side of the central line that is suggested by Wheeler and Chambers (1986). The following statements apply this test with Tests 1, 3, and 4. The resulting chart is shown in Figure 19.188.

```
ods graphics off;
title 'Analysis of Assembly Data';
proc shewhart history=Assembly;
   xrchart Offset * Sample /
      mu0       = 20
      sigma0    = 2.24
      limitn    = 5
      alln
      tests     = 1
                  t( k=14 m=17
                     lower=0 upper=. scheme=twosided
                     code=A label='Test A' )
                  3 4
      vaxis     = 16 to 26 by 2
      split     = '/' ;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

Figure 19.188 Generalized T-Pattern Applied to Assembly Data



The specified T-pattern is signaled at 30th subgroup. Consequently, this point is labeled *Test A*.

The general syntax for an M-pattern is of the form

M(K=*k* **DIR=***direction* **CODE=***character* **LABEL=***'label'* **LEGEND=***'legend'* **)**

The options for an M-pattern are summarized in the following table:

Option	Description
K=*k*	Number of points
DIR=INC	Increasing pattern
DIR=DEC	Decreasing pattern
CODE=*character*	Identifier for test (A–H)
LABEL=*'label'*	Label for points that are signaled
LEGEND=*'legend'*	Legend used with the TABLELEGEND option

You must specify the direction of the pattern with the DIR= option. You use the CODE=, LABEL=, and LEGEND= options as described on page 2137.

The general syntax for an S-pattern is of the form

S(K=*k* **CLEV=***α* **FORM=***character* **CODE=***character* **LABEL=***'label'* **LEGEND=***'legend'* **)**

The options for an S-pattern are summarized in the following table:

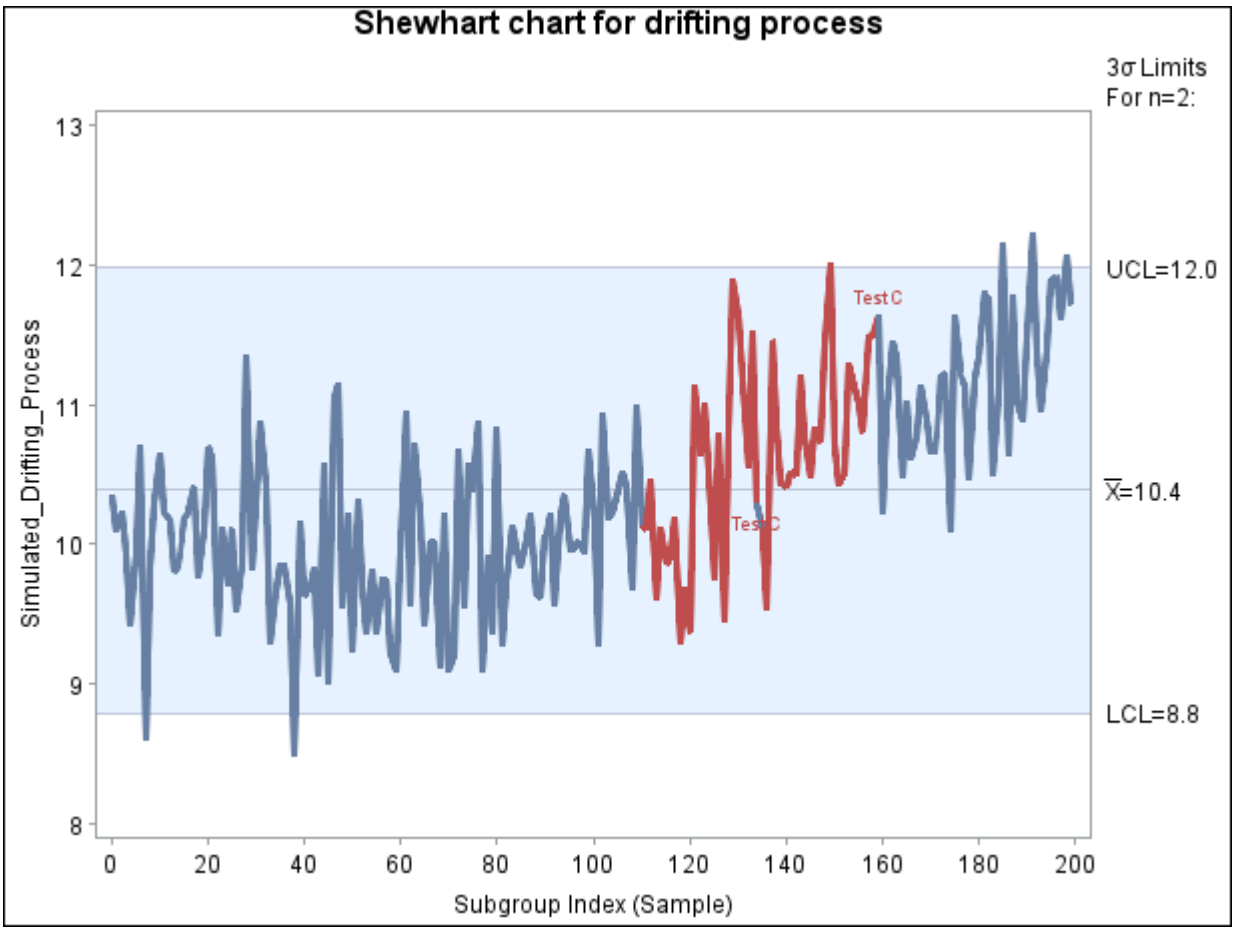
Option	Description
K=*k*	Number of points in sliding window ($k > 2$)
CLEV=*α*	Type I (false positive) error rate ($0 < \alpha \leq 0.5$)
FORM=*character*	Type of trend test (P=parametric, N=nonparametric)
CODE=*character*	Identifier for test (A–H)
LABEL=*'label'*	Label for points that are signaled
LEGEND=*'legend'*	Legend used with the TABLELEGEND option

The S-pattern provides the flexibility to employ either a parametric or nonparametric linear trend test that uses a sliding window of length k. The parametric trend test is based on simple linear regression, with a t test to determine whether the computed slope is statistically significant (Draper and Smith 1981). Similarly, the nonparametric trend test uses a standardized Kendall rank correlation coefficient to identify a statistically significant trend (Kendall 1955). You can vary the power of either test type through judicious choices of the type I error rate and the sliding window length k. You use the CODE=, LABEL=, and LEGEND= options as described on page 2137.

An example of a nonstandard test that uses an S-pattern follows. Here, a simulated process is in statistical control for the first 100 samples, but then the process starts to drift in a linear fashion. The following statements apply the S-pattern test to the simulated data. The resulting chart is shown in Figure 19.189.

```
ods graphics off;
title 'Shewhart chart for drifting process';
proc shewhart data=work.temp;
   irchart Simulated_Drifting_Process*Sample /
          tests=s(k=25 clev=0.02 form=N code=C
                  legend='Nonparametric Slope Test Signaled')
          odstitle=title
          nochart2
          totpanels=1
          outtable=work.temp_out;
run;
```

Figure 19.189 S-Pattern Applied to Simulated Data



The specified S-pattern is signaled at the 134th and 159th samples. Consequently, these points are labeled *Test C*, and the preceding k points are highlighted to indicate the data that exhibit a linear trend.

CAUTION: You should not substitute tests based on arbitrarily defined T-patterns, M-patterns, or S-patterns for standard tests in general process control applications. The pattern options are intended primarily as a research tool.

NOTE: See *ARL With Supplementary Run Rules* in the SAS/QC Sample Library.

Champ and Woodall (1990) provide a FORTRAN program for assessing the run length distribution of tests based on T-patterns. A version of their algorithm is implemented by a SAS/IML program in the SAS/QC Sample Library.

If you specify a T-pattern, M-pattern, or S-pattern with the TESTS= option and save the results in an OUTTABLE= data set, the length of the variable _TESTS_ is 16 rather than 8 (the default). The ninth character of _TESTS_ is assigned the value 'A' if the test with CODE=A is signaled, the tenth character of _TESTS_ is assigned the value 'B' if the test with CODE=B is signaled, and so on. If you also specify one or more standard tests, the ith character of _TESTS_ is assigned the value i if Test i is signaled.

Customizing Tests with DATA Step Programs

NOTE: See *Customizing Tests with DATA Step Programs* in the SAS/QC Sample Library.

Occasionally, you might find it necessary to apply customized tests that cannot be specified with the TESTS= option. You can program your own tests as follows:

1. Run the SHEWHART procedure without the TESTS= option and save the results in an OUTTABLE= data set. Use the NOCHART option to suppress the display of the chart.
2. Use a DATA step program to apply your tests to the subgroup statistics in the OUTTABLE= data set. If tests are signaled at certain subgroups, save these results as values of a flag variable named _TESTS_, which should be a character variable of length 8. Recall that each observation of an OUTTABLE= data set corresponds to a subgroup. Assign the character *i* to the *i*th character of _TESTS_ if the *i*th customized test is signaled at that subgroup (otherwise, assign a blank character).
3. Run the procedure reading the modified data set as a TABLE= data set.

The following example illustrates these steps by creating an $\bar{X}$ chart for the data in Assembly (see "Requesting Standard Tests" on page 2122) that signals a special cause of variation if an average is greater than 2.5 standard errors above the central line. The first step is to compute 2.5σ limits and save both the subgroup statistics and the limits in an OUTTABLE= data set named First.

```
proc shewhart history=Assembly;
   xchart Offset * Sample /
      sigmas   = 2.5
      outtable = First
      nochart ;
run;

title ;
proc print data=First(obs=10) noobs;
run;
```

A partial listing of the data set First is shown in Figure 19.190.

Figure 19.190 Partial Listing of the Data Set First

VAR	Sample	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_	_EXLIM_
Offset	1	2.5	5	5	18.1515	19.80	20.4733	22.7951	2.07665	
Offset	2	2.5	5	5	18.1515	17.16	20.4733	22.7951	2.07665	LOWER
Offset	3	2.5	5	5	18.1515	20.11	20.4733	22.7951	2.07665	
Offset	4	2.5	5	5	18.1515	20.89	20.4733	22.7951	2.07665	
Offset	5	2.5	5	5	18.1515	20.83	20.4733	22.7951	2.07665	
Offset	6	2.5	5	5	18.1515	18.87	20.4733	22.7951	2.07665	
Offset	7	2.5	5	5	18.1515	20.84	20.4733	22.7951	2.07665	
Offset	8	2.5	5	5	18.1515	23.33	20.4733	22.7951	2.07665	UPPER
Offset	9	2.5	5	5	18.1515	19.21	20.4733	22.7951	2.07665	
Offset	10	2.5	5	5	18.1515	20.48	20.4733	22.7951	2.07665	

The second step is to carry out the test and create the flag variable _TESTS_.

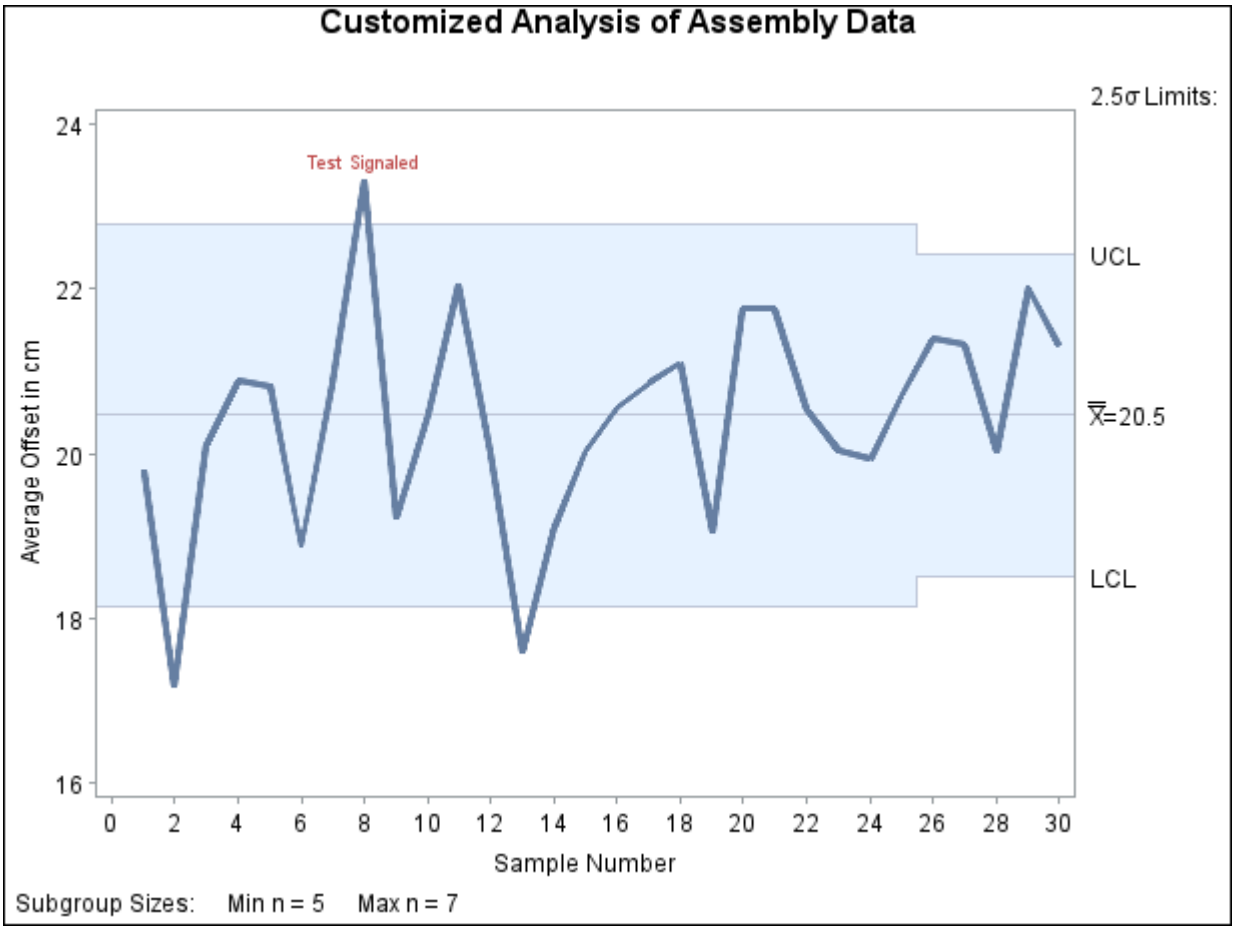
```
data First;
   set First;
   length _tests_ $ 8;
   if _subx_ > _uclx_ then substr( _tests_, 1 ) = '1';
run;
```

Finally, the data set First is read by the SHEWHART procedure as a TABLE= data set.

```
ods graphics off;
title 'Customized Analysis of Assembly Data';
proc shewhart table=First;
   xchart Offset * Sample / tests      = 1
                            testlabel1 = 'Test Signaled';
   label _subx_ = 'Average Offset in cm';
run;
```

The chart is shown in Figure 19.191. Note that the variable _TESTS_ is read "as is" to flag points on the chart, and the standard tests are *not* applied to the data. The option TESTS=1 specifies that a point is to be labeled if the first character of _TESTS_ for the corresponding subgroup is `1`. The label is specified by the TESTLABEL1= option (the default would be *Test1*).

Figure 19.191 Customized Test



In general, you can simultaneously apply up to eight customized tests with the variable _TESTS_, which is of length 8. If two or more tests are signaled at a particular point, the label that is displayed corresponds to the test that appears first in the TESTS= list. In the preceding example, the test involves only the current subgroup. For customized tests involving patterns that span multiple subgroups, you will find it helpful to use the LAG functions described in *SAS Functions and CALL Routines: Reference*.

Notes:

1. If you provide the variable _TESTS_ in a TABLE= data set, you must also use the TESTS= option to specify which characters of _TESTS_ are to be checked.
2. The CTESTS= and LTESTS= options specify colors and line styles for *standard* patterns and might not be applicable with customized tests.

Specialized Control Charts: SHEWHART Procedure

Overview: Specialized Control Charts

Although the Shewhart chart serves well as the fundamental tool for statistical process control (SPC) applications, its assumptions are challenged by many modern manufacturing environments. For example, when standard control limits are used in applications where the process is sampled frequently, autocorrelation in the measurements can result in too many out-of-control signals. This section also considers process control applications involving multiple components of variation, short production runs, nonnormal process data, and multivariate process data.

These questions are subjects of current research and debate. It is not the goal of this section to provide definitive solutions but rather to illustrate some basic approaches that have been proposed and indicate how they can be implemented with short SAS programs. The examples in this section use the SHEWHART procedure in conjunction with various SAS procedures for statistical modeling, as summarized by the following table:

Process Control Application	Modeling Procedure
Diagnosing and modeling autocorrelation in process data	ARIMA
Developing control limits for processes involving multiple components of variation	MIXED
Establishing control with short production runs and checking for constant variance	GLM
Developing control limits for nonnormal individual measurements	CAPABILITY
Creating control charts for multivariate process data	PRINCOMP

Autocorrelation in Process Data

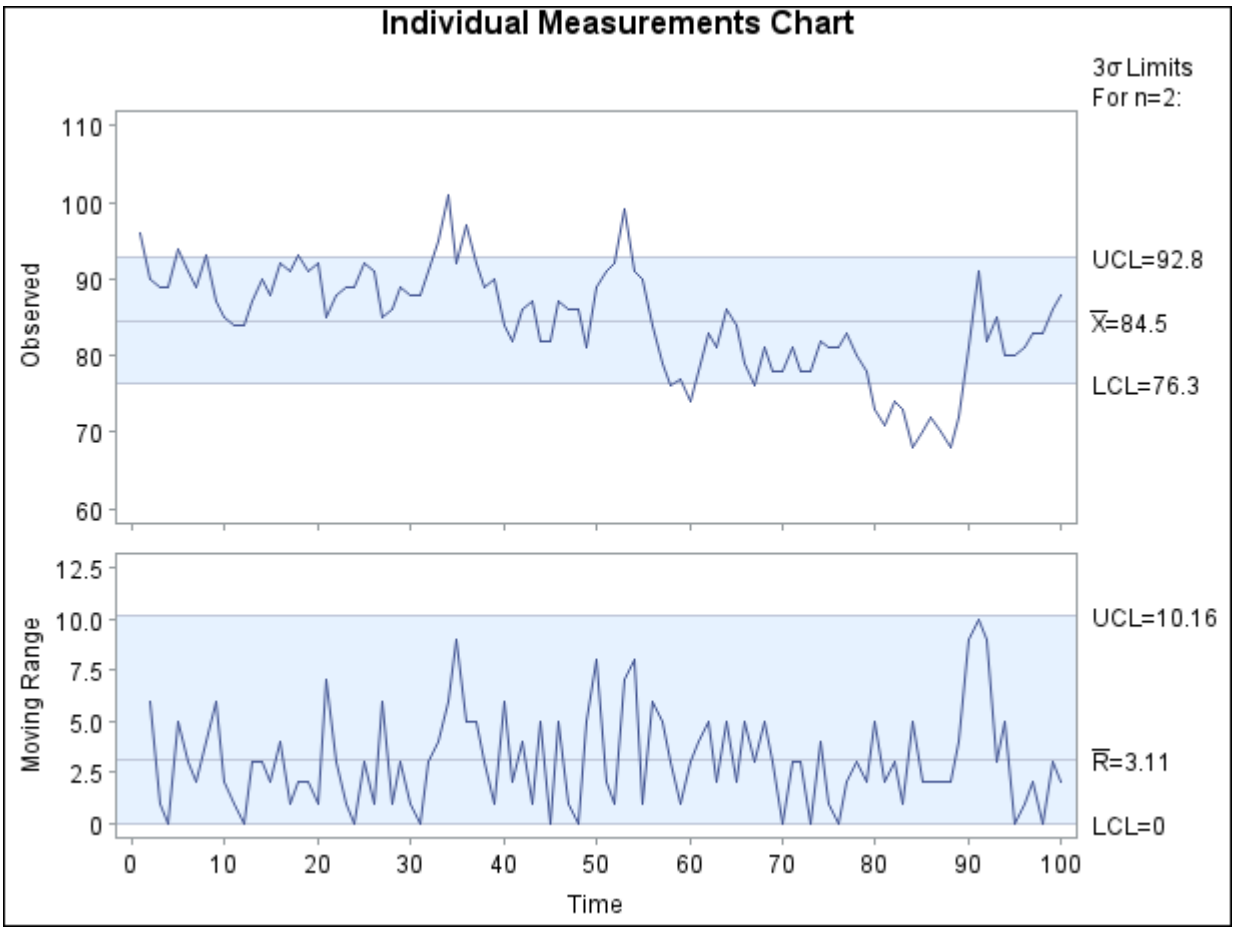
NOTE: See *Autocorrelation in Process Data* in the SAS/QC Sample Library.

Autocorrelation has long been recognized as a natural phenomenon in process industries, where parameters such as temperature and pressure vary slowly relative to the rate at which they are measured. Only in recent years has autocorrelation become an issue in SPC applications, particularly in parts industries, where autocorrelation is viewed as a problem that can undermine the interpretation of Shewhart charts. One reason for this concern is that, as automated data collection becomes prevalent in parts industries, processes are sampled more frequently and it is possible to recognize autocorrelation that was previously undetected. Another reason, noted by Box and Kramer (1992), is that the distinction between parts and process industries is becoming blurred in areas such as computer chip manufacturing. For two other discussions of this issue, refer to Schneider and Pruett (1994) and Woodall (1993).

The standard Shewhart analysis of individual measurements assumes that the process operates with a constant mean μ, and that x_t (the measurement at time t) can be represented as $x_t = \mu + \epsilon_t$, where ϵ_t is a random displacement or error from the process mean μ. Typically, the errors are assumed to be statistically independent in the derivation of the control limits displayed at three standard deviations above and below the central line, which represents an estimate for μ.

When Shewhart charts are constructed from autocorrelated measurements, the result can be too many false signals, making the control limits seem too tight. This situation is illustrated in Figure 19.192, which displays an individual measurement and moving range chart for 100 observations of a chemical process.

Figure 19.192 Conventional Shewhart Chart



The measurements are saved in a SAS data set named Chemical.[18] The chart in Figure 19.192 is created with the following statements:

```
symbol h=2.0 pct;
title 'Individual Measurements Chart';
proc shewhart data=Chemical;
   irchart xt*t / npanelpos = 100
                  split     = '/';
   label xt = 'Observed/Moving Range'
         t = 'Time';
run;
```

Diagnosing and Modeling Autocorrelation

You can diagnose autocorrelation with an autocorrelation plot created with the ARIMA procedure.

```
ods graphics on;
ods select ChiSqAuto SeriesACFPlot SeriesPACFPlot;
proc arima data=Chemical plots(only)=series(acf pacf);
```

[18] The measurements are patterned after the values plotted in Figure 1 of Montgomery and Mastrangelo (1991).

```
   identify var = xt;
run;
quit;
```

Refer to *SAS/ETS User's Guide* for details on the ARIMA procedure. The output, shown in Figure 19.193 and Figure 19.194, indicates that the data are highly autocorrelated with a lag 1 autocorrelation of 0.83.

Figure 19.193 Autocorrelation Check for Chemical Data

Individual Measurements Chart

The ARIMA Procedure

Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	228.15	6	<.0001	0.830	0.718	0.619	0.512	0.426	0.381
12	315.34	12	<.0001	0.360	0.364	0.380	0.347	0.348	0.354
18	406.76	18	<.0001	0.349	0.371	0.348	0.353	0.368	0.341
24	442.15	24	<.0001	0.303	0.261	0.230	0.184	0.141	0.098

Figure 19.194 Autocorrelation Plots for Chemical Data

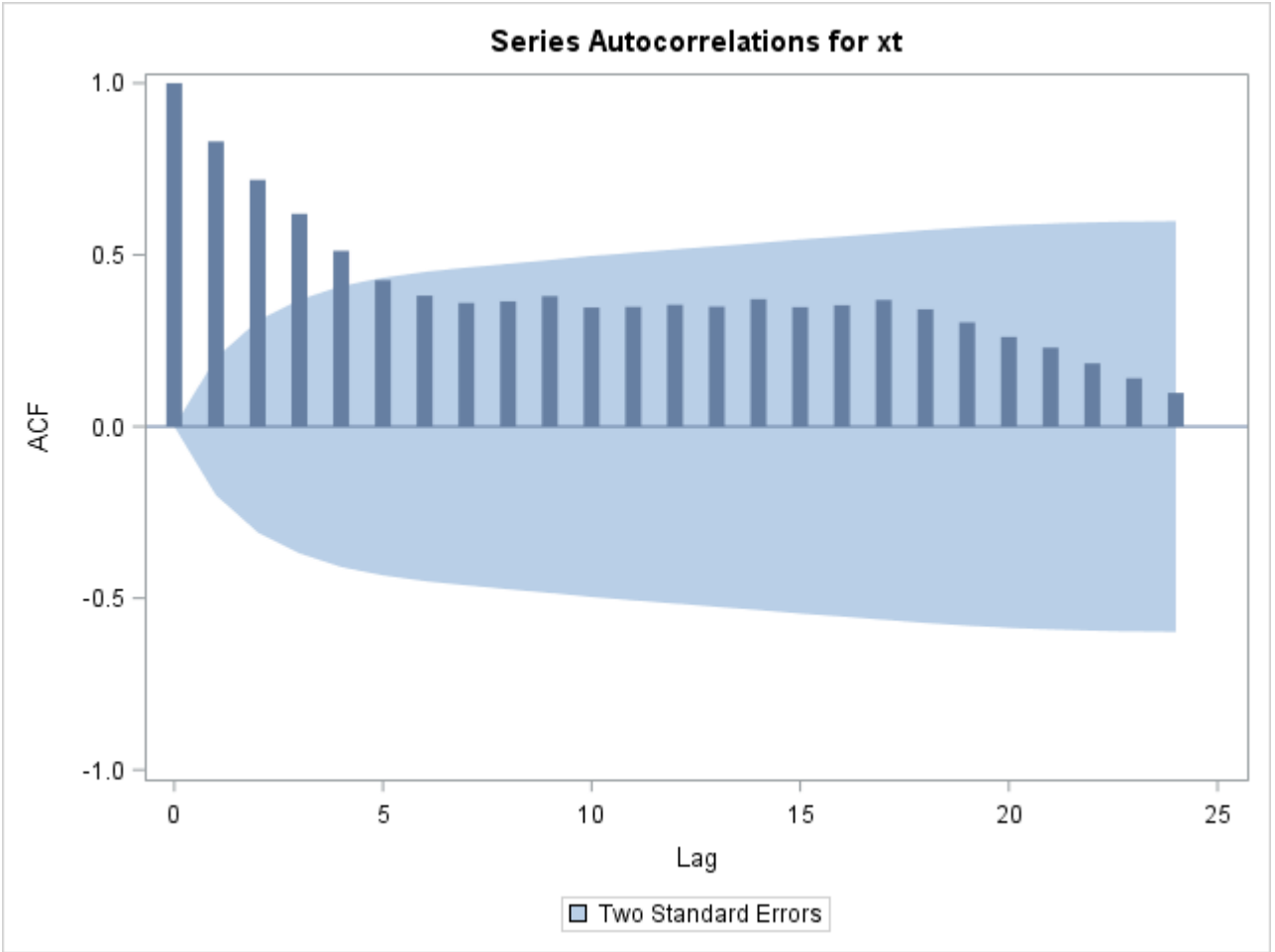
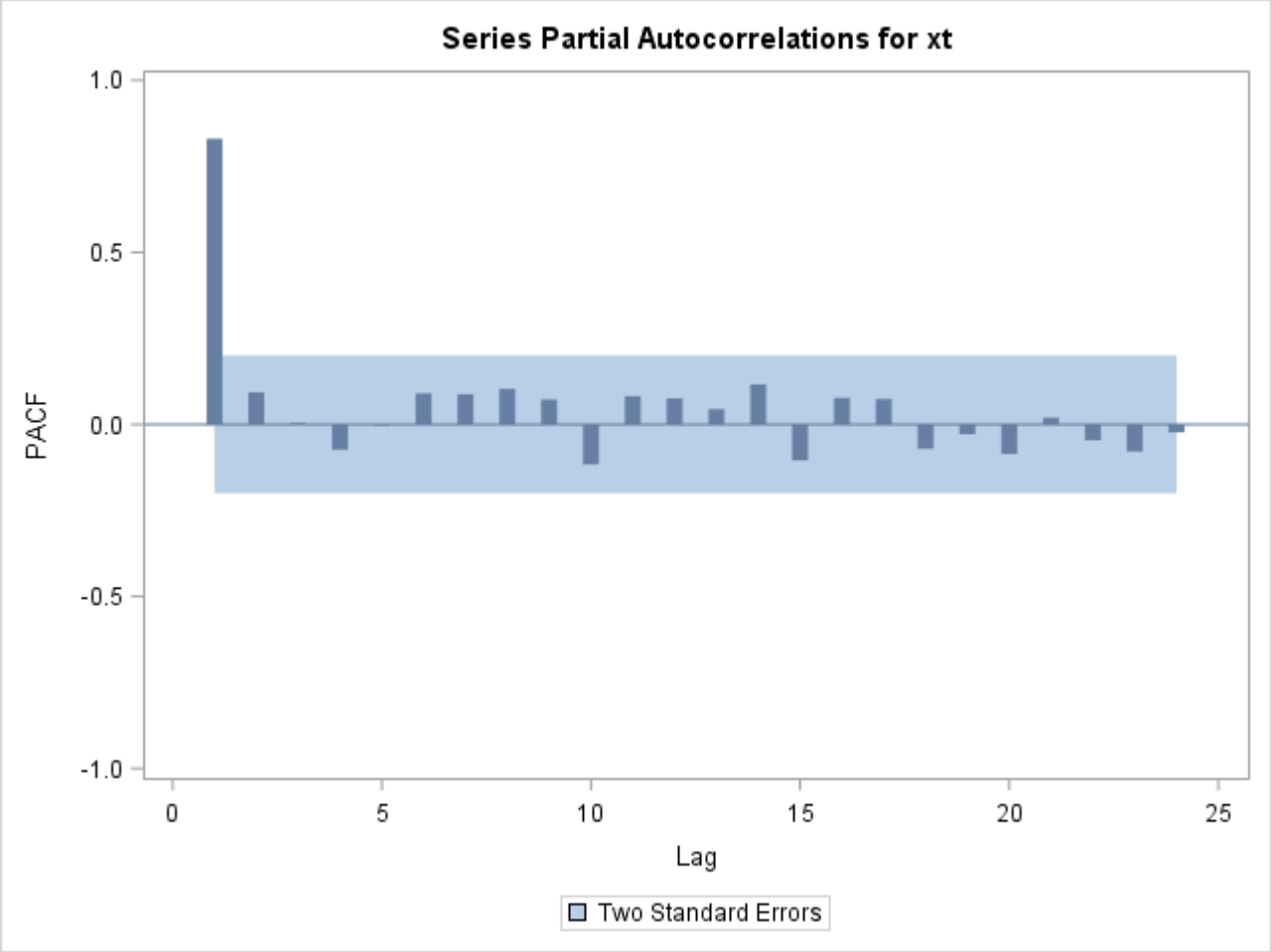


Figure 19.194 *continued*



The partial autocorrelation plot in Figure 19.194 suggests that the data can be modeled with a first-order autoregressive model, commonly referred to as an AR(1) model.

$$\tilde{x}_t \equiv x_t - \mu = \phi_0 + \phi_1 \tilde{x}_{t-1} + \epsilon_t$$

You can fit this model with the ARIMA procedure. The results in Figure 19.195 show that the equation of the fitted model is $\tilde{x}_t = 13.05 + 0.847\tilde{x}_{t-1}$.

```
ods select ParameterEstimates;
proc arima data=Chemical;
   identify var=xt;
   estimate p=1 method=ml;
run;
```

Figure 19.195 Fitted AR(1) Model

Individual Measurements Chart

The ARIMA Procedure

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	85.28375	2.32973	36.61	<.0001	0
AR1,1	0.84694	0.05221	16.22	<.0001	1

Strategies for Handling Autocorrelation

There is considerable disagreement on how to handle autocorrelation in process data. Consider the following three views:

- At one extreme, Wheeler (1991) argues that the usual control limits are contaminated "only when the autocorrelation becomes excessive (say 0.80 or larger)." He concludes that "one need not be overly concerned about the effects of autocorrelation upon the control chart."
- At the opposite extreme, automatic process control (APC), also referred to as engineering process control, views autocorrelation as a phenomenon to be exploited. In contrast to SPC, which assumes that the process remains on target unless an unexpected but removable cause occurs, APC assumes that the process is changing dynamically due to known causes that cannot be eliminated. Instead of avoiding "overcontrol" and "tampering," which have a negative connotation in the SPC framework, APC advocates continuous tuning of the process to achieve minimum variance control. Descriptions of this approach and discussion of the differences between APC and SPC are provided by a number of authors, including Box and Kramer (1992), MacGregor (1987, 1990), MacGregor, Hunter, and Harris (1988), and Montgomery et al. (1994).
- A third strategy advocates removing autocorrelation from the data and constructing a Shewhart chart (or an EWMA chart or a cusum chart) for the residuals; refer, for example, to Alwan and Roberts (1988).

An example of the last approach is presented in the remainder of this section simply to demonstrate the use of the ARIMA procedure in conjunction with the SHEWHART procedure. The ARIMA procedure models the autocorrelation and saves the residuals in an output data set; the SHEWHART procedure creates a control chart using the residuals as input data.

In the chemical data example, the residuals can be computed as forecast errors and saved in an output SAS data set with the FORECAST statement in the ARIMA procedure.

```
proc arima data=Chemical;
   identify var=xt;
   estimate p=1 method=ml;
   forecast out=Results id=t;
run;
```

The output data set (named Results) saves the one-step-ahead forecasts as a variable named forecast, and it also contains the original variables xt and t. You can create a Shewhart chart for the residuals by using the data set Results as input to the SHEWHART procedure.

```
title 'Residual Analysis Using AR(1) Model';
symbol h=2.0 pct;
proc shewhart data=Results(firstobs=4 obs=100);
   xchart xt*t / npanelpos = 100
                 split     = '/'
                 trendvar  = forecast
                 xsymbol   = xbar
                 ypct1     = 40
                 vref2     = 70 to 100 by 10
                 lvref     = 2
                 nolegend;
   label xt = 'Residual/Forecast'
         t = 'Time';
run;
```

The chart is shown in Figure 19.196. Specifying TRENDVAR=forecast plots the values of forecast in the lower chart and plots the residuals (xt – forecast) together with their 3σ limits in the upper chart.[19]

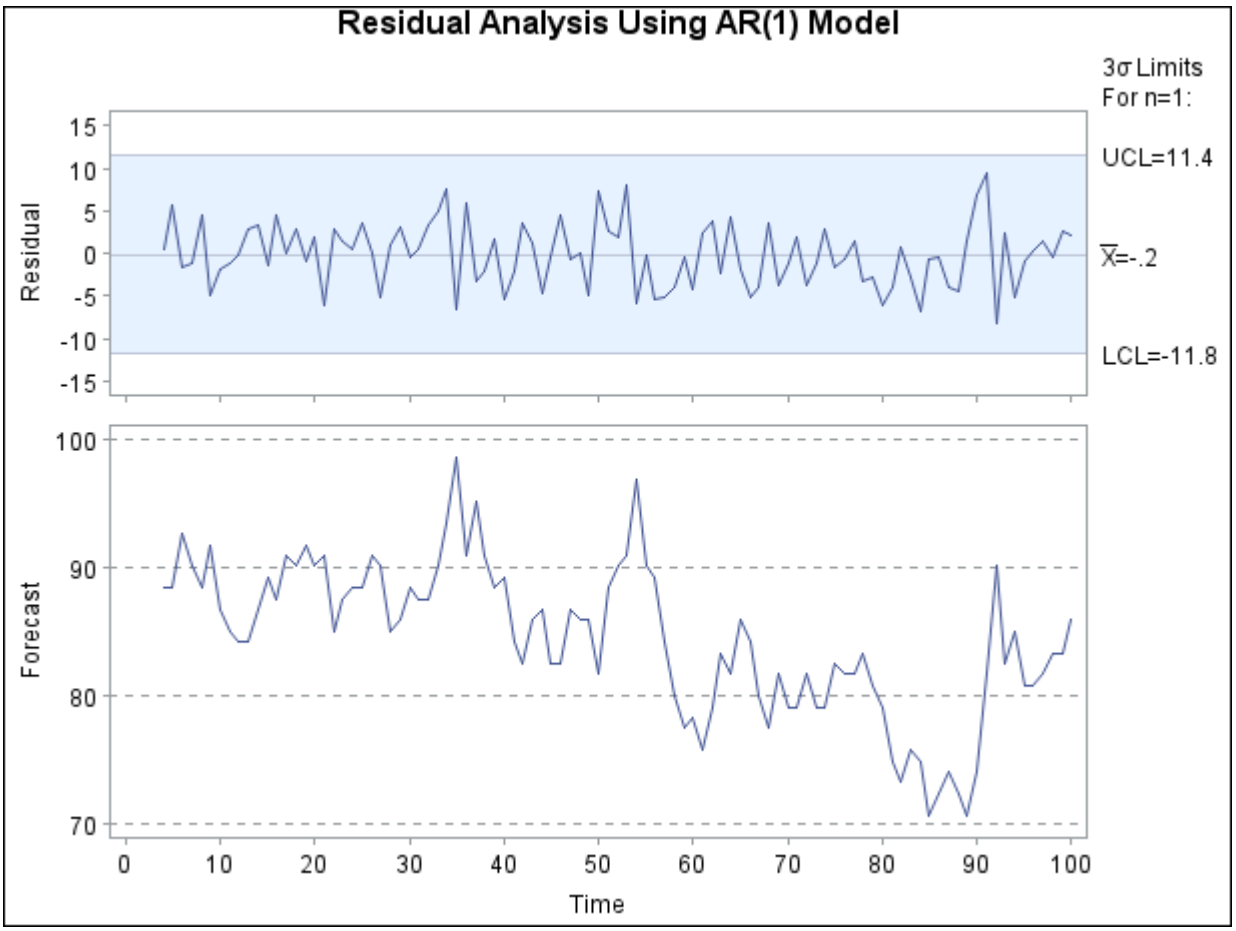
Various other methods can be applied with this data. For example, Montgomery and Mastrangelo (1991) suggest fitting an exponentially weighted moving average (EWMA) model and using this model as the basis for a display that they refer to as an *EWMA central line control chart*.

Before presenting the statements for creating this display, it is helpful to review some terminology. The EWMA *statistic* plotted on a conventional EWMA control chart is defined as

$$z_t = \lambda x_t + (1 - \lambda) z_{t-1}$$

[19]The upper chart in Figure 19.196 resembles Figure 2 of Montgomery and Mastrangelo (1991), who conclude that the process is in control.

Figure 19.196 Residuals from AR(1) Model



The EWMA chart (which you can construct with the MACONTROL procedure) is based on the assumption that the observations x_t are independent. However, in the context of autocorrelated process data (and more generally in time series analysis), the EWMA statistic z_t plays a different role:[20] it is the optimal one-step-ahead forecast for a process that can be modeled by an ARIMA(0,1,1) model

$$x_t = x_{t-1} + \epsilon_t - \theta\epsilon_{t-1}$$

provided that the weight parameter λ is chosen as $\lambda = 1 - \theta$. This statistic is also a good predictor when the process can be described by a subset of ARIMA models for which the process is "positively autocorrelated and the process mean does not drift too quickly."[21]

You can fit an ARIMA(0,1,1) model to the chemical data with the following statements. A summary of the fitted model is shown in Figure 19.197.

[20] For a discussion of these roles, refer to Hunter (1986).

[21] Refer to Montgomery and Mastrangelo (1991) and the discussion that follows their paper.

```
title ;
proc arima data=Chemical;
   identify var=xt(1);
   estimate q=1 method=ml noint;
   forecast out=EWMA id=t;
run;
```

Figure 19.197 Fitted ARIMA(0, 1, 1) Model

The ARIMA Procedure

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.15041	0.10021	1.50	0.1334	1

Variance Estimate	14.97024
Std Error Estimate	3.86914
AIC	549.868
SBC	552.4631
Number of Residuals	99

The forecast values and their standard errors (variables forecast and STD), together with the original measurements, are saved in a data set named EWMA. The EWMA central line control chart plots the forecasts from the ARIMA(0,1,1) model as the central "line," and it uses the standard errors of prediction to determine upper and lower control limits. You can construct this chart, shown in Figure 19.198,[22] with the following statements:

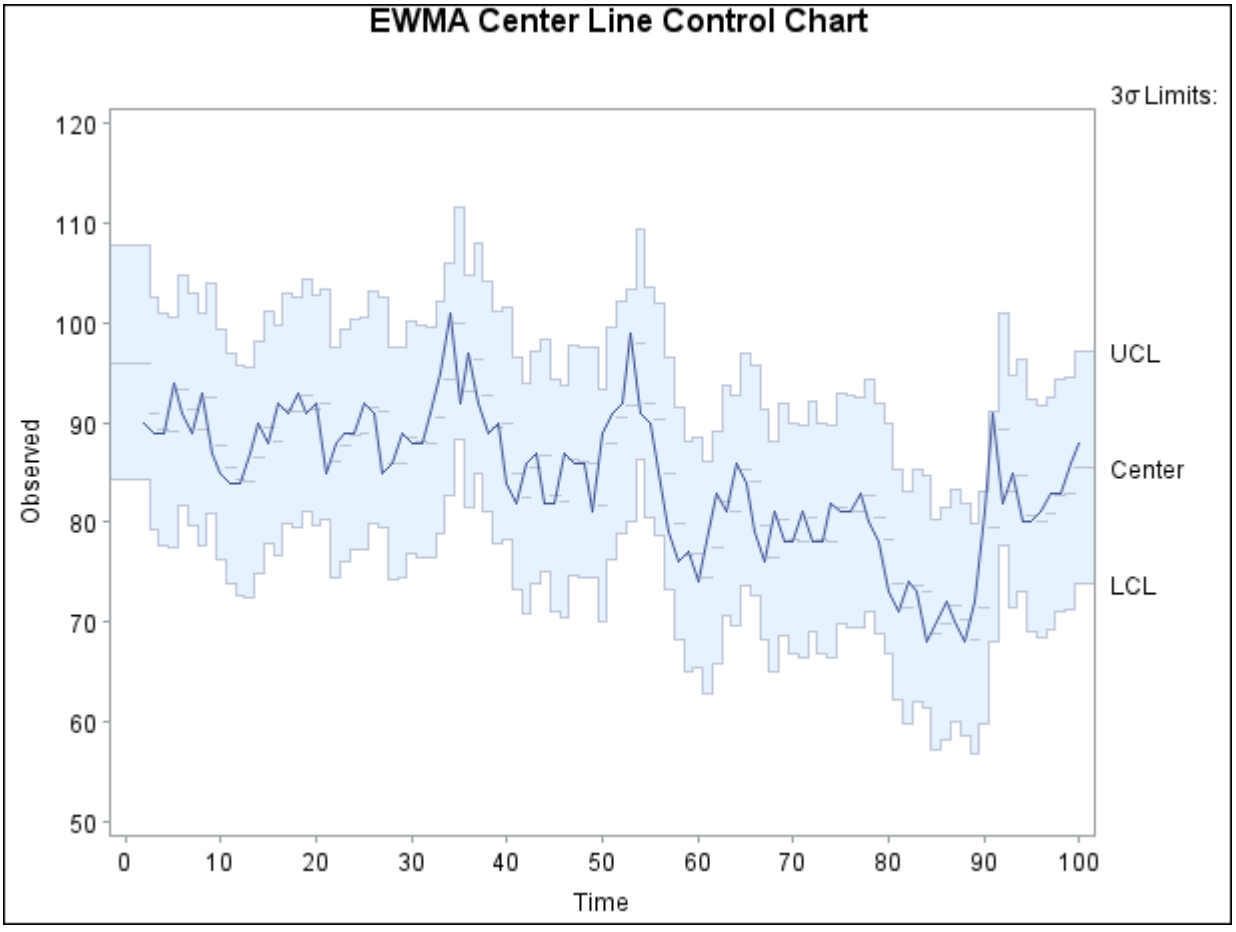
```
data EWMA;
   set EWMA(firstobs=2 obs=100);
run;

data EWMAtab;
   length _var_ $ 8 ;
   set EWMA (rename=(forecast=_mean_ xt=_subx_));
   _var_    = 'xt';
   _sigmas_ = 3;
   _limitn_ = 1;
   _lclx_   = _mean_ - 3 * std;
   _uclx_   = _mean_ + 3 * std;
   _subn_   = 1;
run;

symbol h=2.0 pct;
title 'EWMA Center Line Control Chart';
proc shewhart table=EWMAtab;
   xchart xt*t / npanelpos = 100
                 xsymbol   = 'Center'
                 nolegend;
   label _subx_ = 'Observed'
               t = 'Time' ;
run;
```

[22] Figure 19.198 is similar to Figure 5 of Montgomery and Mastrangelo (1991).

Note that EWMA is read by the SHEWHART procedure as a TABLE= input data set, which has a special structure intended for applications in which both the statistics to be plotted and their control limits are pre-computed. The variables in a TABLE= data set have reserved names beginning and ending with the underscore character; for this reason, forecast and xt are temporarily renamed as _MEAN_ and _SUBX_, respectively. For more information about TABLE= data sets, see "Input Data Sets" in the section for the chart statement in which you are interested.

Again, the conclusion is that the process is in control. While Figure 19.196 and Figure 19.198 are not the only displays that can be considered for analyzing the chemical data, their construction illustrates the conjunctive use of the ARIMA and SHEWHART procedures in process control applications involving autocorrelated data.

Figure 19.198 EWMA Center Line Chart



Multiple Components of Variation

NOTE: See *Multiple Components of Variation* in the SAS/QC Sample Library.

In the preceding section, the excessive variation in the conventional Shewhart chart in Figure 19.192 is the result of positive autocorrelation in the data. The variation is "excessive" not because it is due to special causes of variation, but because the Shewhart model is inappropriate. This section considers another form

of departure from the Shewhart model; here, measurements are *independent* from one subgroup sample to the next, but there are multiple components of variation for each measurement. This is illustrated with an example involving two components.[23]

A company that manufactures polyethylene film monitors the statistical control of an extrusion process that produces a continuous sheet of film. At periodic intervals of time, samples are taken at four locations (referred to as lanes) along a cross section of the sheet, and a test measurement is made of each sample. The test values are saved in a SAS data set named Film. A partial listing of Film is shown in Figure 19.199.

Figure 19.199 Polyethylene Sheet Measurements in the Data Set Film

Sample	Lane	Testval
1	A	93
1	B	87
1	C	92
1	D	78
2	A	87

Preliminary Examination of Variation

As a preliminary step in the analysis, the data are sorted by lane and visually screened for outliers (test values greater than 130) with box plots created as follows:

```
ods graphics off;
proc sort data=Film;
   by Lane;
run;
symbol v = dot h = 2.0 pct;
title 'Outlier Analysis';
proc shewhart data=Film;
   boxchart Testval*Lane / boxstyle = schematicid
                           idsymbol = dot
                           vref     = 130
                           vreflab  = 'Outlier Cutoff'
                           hoffset  = 5
                           nolegend
                           stddevs
                           nolimits ;
   id Sample;
run;
```

Specifying BOXSTYLE=SCHEMATICID requests schematic box plots with outliers identified by the value of the ID variable Sample. The STDDEVS option specifies that the estimate of the process standard deviation is to be based on subgroup standard deviations. Although this estimate is not needed here because control limits are not displayed, it is recommended that you specify the STDDEVS option whenever you are working with subgroup sample sizes greater than ten. The NOLEGEND and NOLIMITS options suppress the subgroup sample size legend and control limits for lane means that are displayed by default. The display is shown in Figure 19.200.

[23] Also refer to Chapter 5 of Wheeler and Chambers (1986) for an explanation of the effects of subgrouping and sources of variation on control charts.

Figure 19.200 Outlier Analysis for the Data Set Film

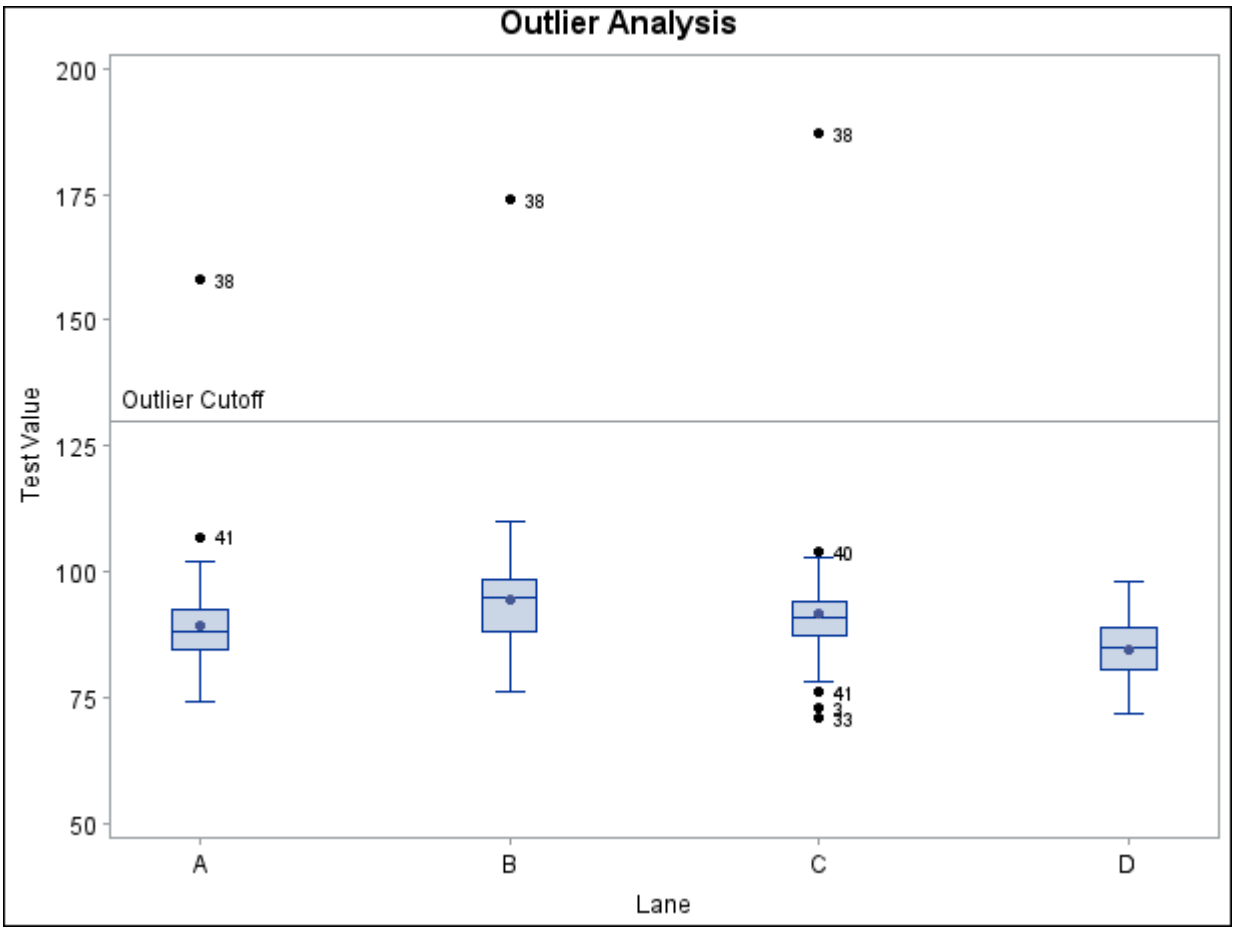
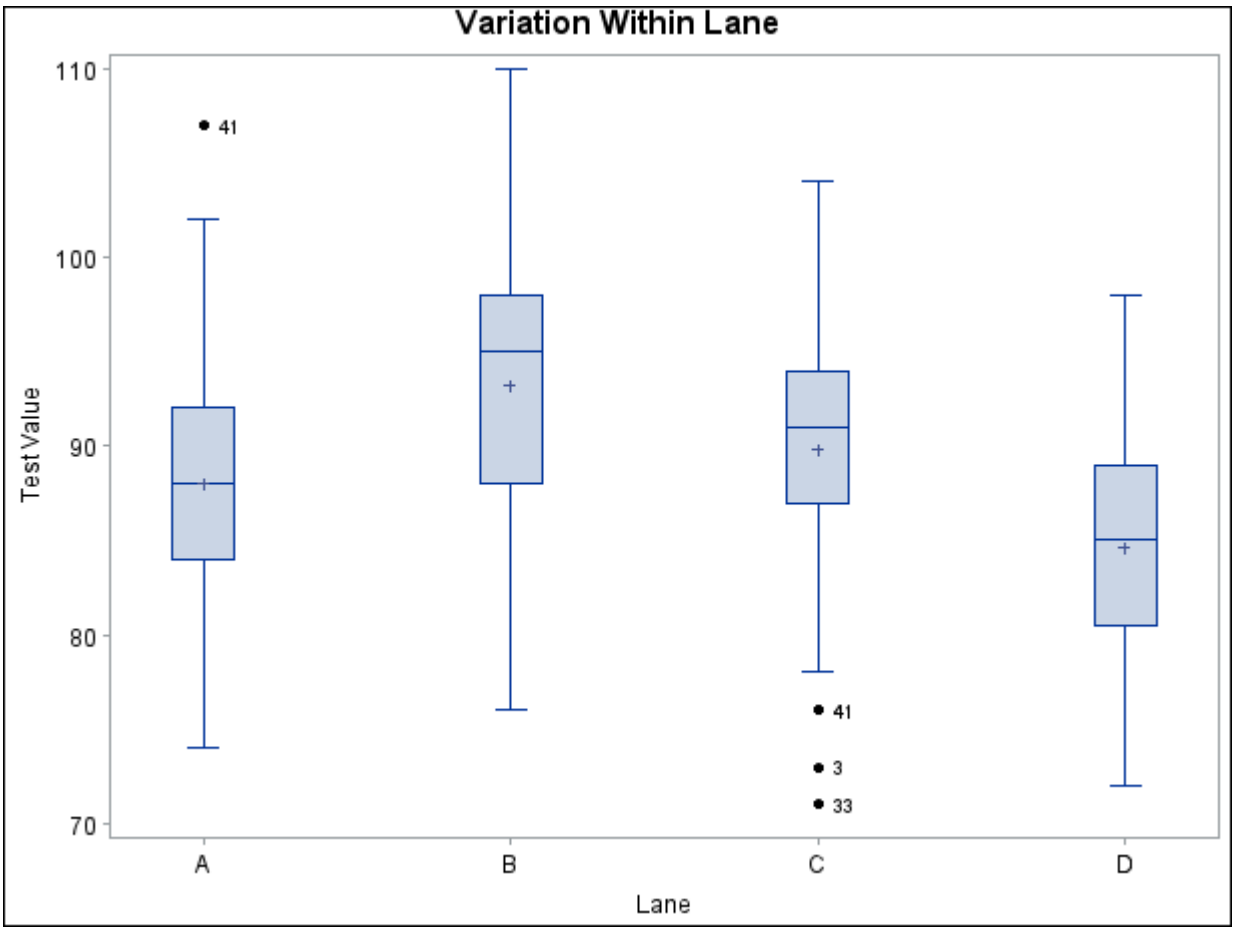


Figure 19.201 shows similarly created box plots for the data in Film2, which is created by removing the outliers from the data set Film.

```
data Film2;
   set Film;
   if Testval < 130;
symbol h = 2.0 pct;
title 'Variation Within Lane';
proc shewhart data=Film2;
   boxchart Testval*Lane / boxstyle = schematicid
                           boxwidth = 5
                           idsymbol = dot
                           hoffset  = 5
                           nolegend
                           stddevs
                           nolimits ;
   id Sample;
run;
```

Figure 19.201 The Data Set Film2 Without Outliers



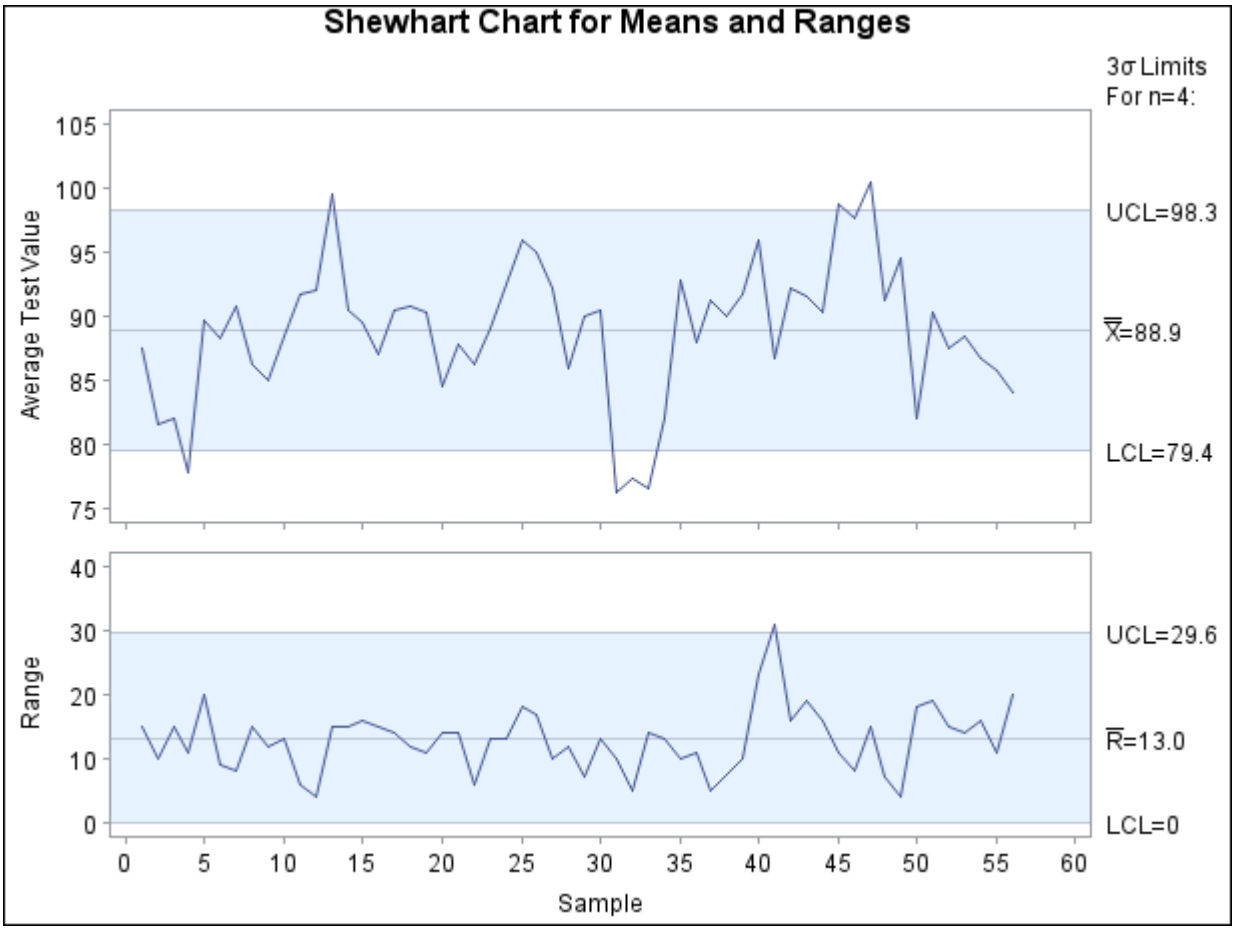
Because you have no additional information about the process, you might want to create a conventional $\bar{X}$ and R chart for the test values grouped by the variable Sample. This is a straightforward application of the XRCHART statement in the SHEWHART procedure.

```
proc sort data=Film2;
   by Sample;
run;
symbol h=2.0 pct;
title 'Shewhart Chart for Means and Ranges';
proc shewhart data=Film2;
   xrchart Testval*Sample /
      split     = '/'
      npanelpos = 60
      limitn    = 4
      outlimits = RLimits
      nolegend
      alln;
   label Testval='Average Test Value/Range';
run;
```

The $\bar{X}$ and R chart is displayed in Figure 19.202. Ordinarily, the out-of-control points in the $\bar{X}$ chart would indicate that the process is not in statistical control. In this situation, however, the process is known to be

quite stable, and the data have been screened for outliers. The problem is that the control limits for the average test value were computed from an inappropriate model. This is discussed in the following section.

Figure 19.202 Conventional $\bar{X}$ and R Chart



Determining the Components of Variation

The standard Shewhart analysis assumes that sampling variation, also referred to as *within-group* variation, is the only source of variation. Writing x_{ij} for the jth measurement within the ith subgroup, you can express the model for the conventional $\bar{X}$ and R chart as

$$x_{ij} = \mu + \sigma_W \epsilon_{ij} \tag{1}$$

for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, n$. The random variables ϵ_{ij} are assumed to be independent with zero mean and unit variance, and σ_W^2 is the within-subgroup variance. The parameter μ denotes the process mean.

In a process such as film manufacturing, this model is not adequate because there is additional variation due to changes in temperature, pressure, raw material, and other factors. A more appropriate model is

$$x_{ij} = \mu + \sigma_B \omega_i + \sigma_W \epsilon_{ij} \tag{2}$$

where σ_B^2 is the *between-subgroup* variance, the random variables ω_i are independent with zero mean and unit variance, and the random variables ω_i are independent of the random variables ϵ_{ij}.[24]

To plot the subgroup averages $\bar{x}_{i.} \equiv \frac{1}{n}\sum_{j=1}^{n} x_{ij}$ on a control chart, you need expressions for the expectation and variance of $\bar{x}_{i.}$. These are

$$\begin{aligned} E(\bar{x}_{i.}) &= \mu \\ \mathrm{Var}(\bar{x}_{i.}) &= \sigma_B^2 + \frac{\sigma_W^2}{n} \end{aligned}$$

Thus, the central line should be located at $\hat{\mu}$, and 3σ limits should be located at

$$\hat{\mu} \pm 3\sqrt{\widehat{\sigma_B^2} + \frac{\widehat{\sigma_W^2}}{n}} \tag{3}$$

where $\widehat{\sigma_B^2}$ and $\widehat{\sigma_W^2}$ denote estimates of the variance components. You can use a variety of SAS procedures for fitting linear models to estimate the variance components. The following statements show how this can be done with the MIXED procedure:

```
title;
proc mixed data=Film2;
   class Sample;
   model Testval = / s;
   random Sample;
   ods output solutionf=sf;
   ods output covparms=cp;
run;
```

The results are shown in Figure 19.203. Note that the parameter estimates are $\widehat{\sigma_B^2} = 19.25$, $\widehat{\sigma_W^2} = 39.68$, and $\hat{\mu} = 88.90$.

Figure 19.203 Partial Output from the MIXED Procedure

The Mixed Procedure

Covariance Parameter Estimates

Cov Parm	Estimate
Sample	19.2526
Residual	39.6825

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	88.8963	0.7250	55	122.61	<.0001

[24] This notation is used in Chapter 3 of Wetherill and Brown (1991), which discusses this issue.

The following statements merge the output data sets from the MIXED procedure into a SAS data set named Newlim that contains the appropriately derived control limit parameters for average test value:

```
data cp;
   set cp sf;
   keep Estimate;
run;

proc transpose data=cp out=Newlim;
run;

data Newlim (keep=_lclx_ _mean_ _uclx_);
   set Newlim;
   _limitn_ = 4;
   _mean_   = col3;
   _stddev_ = sqrt(4*col1 + col2);
   _lclx_   = _mean_ - 3*_stddev_ / sqrt(_limitn_);
   _uclx_   = _mean_ + 3*_stddev_ / sqrt(_limitn_);
   output;
run;
```

Here, the variable _LIMITN_ is assigned the value of n, the variable _MEAN_ is assigned the value of $\hat{\mu}$, and the variable _STDDEV_ is assigned the value of

$$\hat{\sigma}_{\text{adj}} \equiv \sqrt{4\widehat{\sigma_B^2} + \widehat{\sigma_W^2}}$$

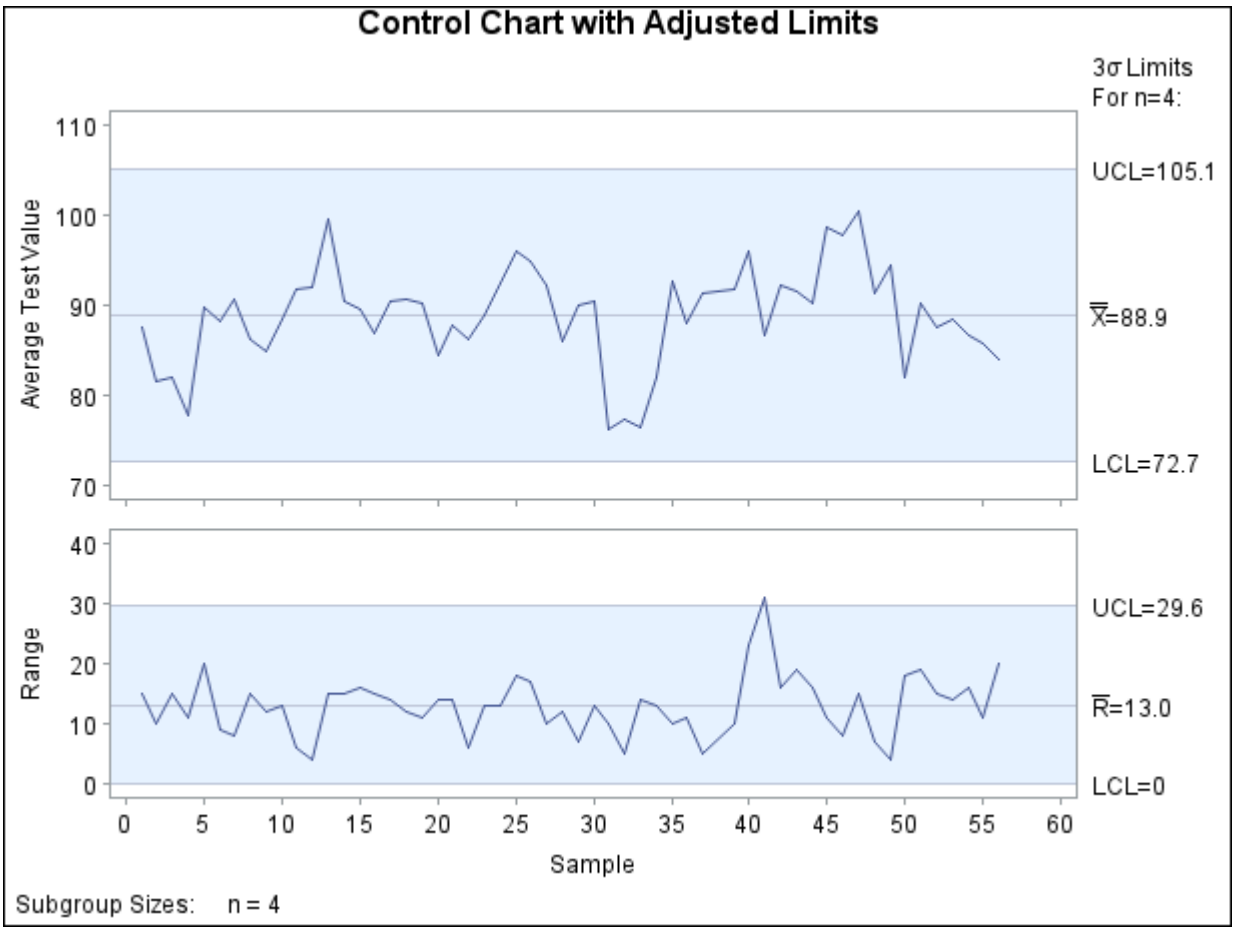
The 3σ limits (_LCLX_ and _UCLX_) are computed according to (3) using $\hat{\sigma}_{\text{adj}}$. The data set Newlim contains the mean and 3σ limits for the average test value.

The following statements compute appropriate control limits for the $\bar{X}$ and R charts, which are shown in Figure 19.204. First, the data set Newlim2 is created by merging the data set RLimits, which contains the original R chart limits computed in "Preliminary Examination of Variation" on page 2153, with Newlim, which saved the appropriate $\bar{X}$ chart limits. The original R chart limits are valid because the range in the ith subgroup is $R_i = \sigma_W(\max_j \epsilon_{ij} - \min_j \epsilon_{ij})$, which is the same for models (1) and (2). The LIMITS= option specifies the data set Newlim2 as the source of the control limits for Figure 19.204.

```
data Newlim2;
   merge Newlim RLimits (drop=_lclx_ _mean_ _uclx_);
run;

title 'Control Chart with Adjusted Limits';
symbol h = 2.0 pct;
proc shewhart data=Film2 limits=Newlim2;
   xrchart Testval*Sample / npanelpos = 60;
   label Testval='Average Test Value';
run;
```

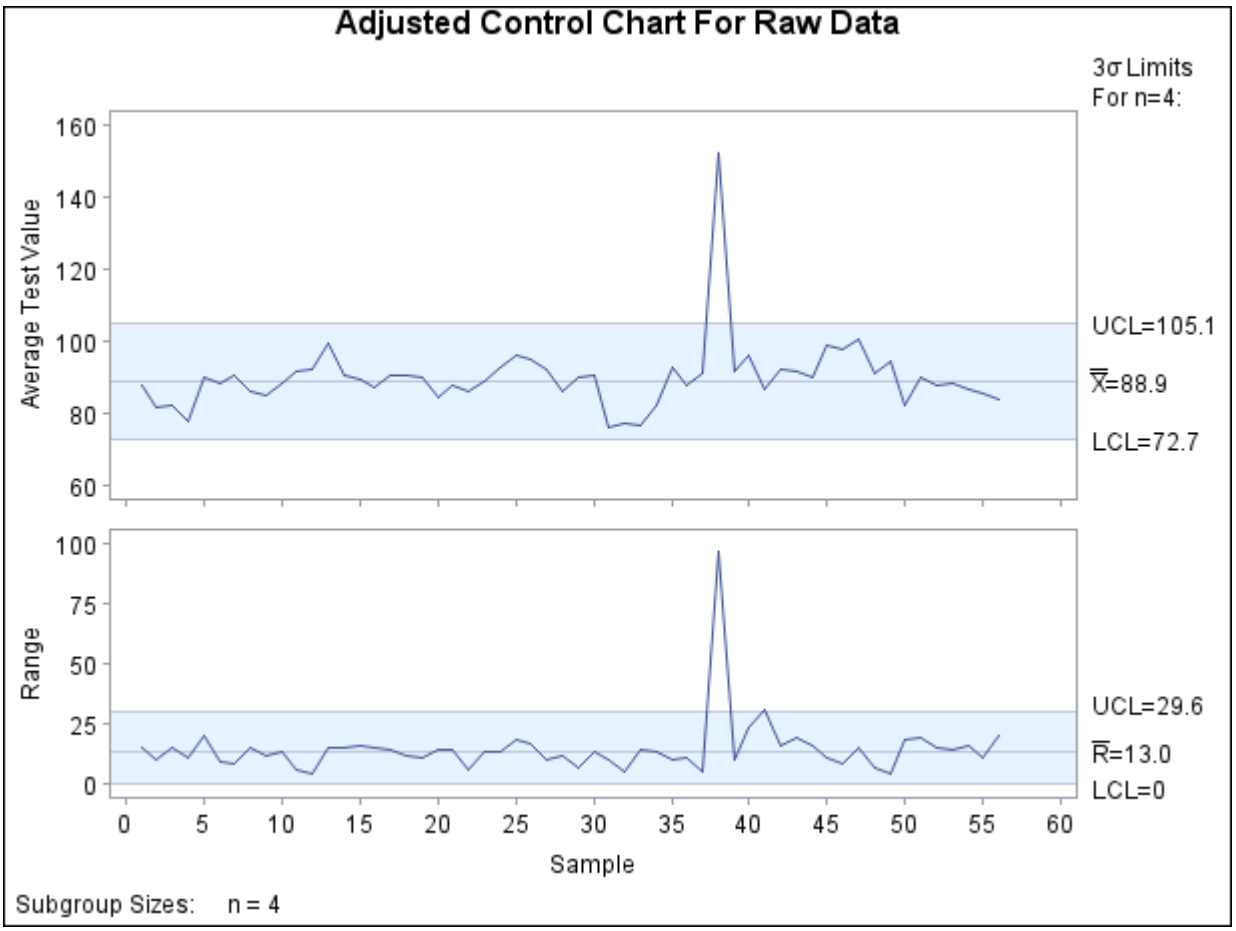
The control limits for the $\bar{X}$ chart in Figure 19.204 are $\hat{\mu} \pm \frac{3}{\sqrt{n}}\hat{\sigma}_{\text{adj}}$. This chart correctly indicates that the variation in the process is due to common causes.

Figure 19.204 $\bar{X}$ and R Chart with Derived Control Limits



You can use a similar set of statements to display the derived control limits in Newlim on an $\bar{X}$ and R chart for the original data (including outliers), as shown in Figure 19.205.

Figure 19.205 $\bar{X}$ and R Chart with Derived Control Limits for Raw Data



A simple alternative to the chart in Figure 19.204 is an "individual measurements" chart for the subgroup means. The advantage of the variance components approach is that it yields separate estimates of the components due to lane and sample, as well as a number of hypothesis tests (these require assumptions of normality). In applying this method, however, you should be careful to use data that represent the process in a state of statistical control.

Short Run Process Control

NOTE: See *Short Run Process Control* in the SAS/QC Sample Library.

When conventional Shewhart charts are used to establish statistical control, the initial control limits are typically based on 25 to 30 subgroup samples. Often, however, this amount of data is not available in manufacturing situations where product changeover occurs frequently or production runs are limited.

A variety of methods have been introduced for analyzing data from a process that is alternating between short runs of multiple products. The methods commonly used in the United States are variations of two basic approaches:[25]

[25] For a review of related methods, refer to Al-Salti and Statham (1994).

- the *difference from nominal* approach. A product-specific nominal value is subtracted from each measured value, and the differences (together with appropriate control limits) are charted. Here it is assumed that the nominal value represents the central location of the process (ideally estimated with historical data) and that the process variability is constant across products.
- the *standardization* approach. Each measured value is standardized with a product-specific nominal and standard deviation values. This approach is followed when the process variability is not constant across products.

These approaches are highlighted in this section because of their popularity, but two alternatives that are technically more sophisticated are worth noting.

- Hillier (1969) provided a method for modifying the usual control limits for $\bar{X}$ and R charts in startup situations where fewer than 25 subgroup samples are available for estimating the process mean μ and standard deviation σ; also refer to Quesenberry (1993).
- Quesenberry (1991b, a) introduced the so-called *Q chart* for short (or long) production runs, which standardizes and normalizes the data using probability integral transformations.

SAS examples illustrating these alternatives are provided in the SAS/QC sample library and are described by Rodriguez and Bynum (1992).

Analyzing the Difference from Nominal

The following example[26] is adapted from an application in aircraft component manufacturing. A metal extrusion process is used to make three slightly different models of the same component. The three product types (labeled M1, M2, and M3) are produced in small quantities because the process is expensive and time-consuming.

Figure 19.206 shows the structure of a SAS data set named Old, which contains the diameter measurements for various short runs. Samples 1 to 30 are to be used to estimate the process standard deviation σ for the differences from nominal.

[26] Refer to Chapter 1 of Wheeler (1991) for a similar example.

Figure 19.206 Diameter Measurements in the Data Set Old

Sample	Prodtype	Diameter
1	M3	13.99
2	M3	14.69
3	M3	13.86
4	M3	14.32
5	M3	13.23
6	M1	17.55
7	M1	14.26
8	M1	14.62
9	M1	12.97
10	M2	16.18
11	M2	15.29
12	M2	16.20
13	M3	13.89
14	M3	12.71
15	M3	14.32
16	M3	15.35
17	M2	15.08
18	M2	14.72
19	M2	14.79
20	M2	15.27
21	M2	15.95
22	M1	14.78
23	M1	15.19
24	M1	15.41
25	M1	16.26
26	M3	16.68
27	M3	15.60
28	M3	14.86
29	M3	16.67
30	M3	14.35

In short run applications involving many product types, it is common practice to maintain a database for the nominal values for the product types. Here, the nominal values are saved in a SAS data set named Nomval, which is listed in Figure 19.207.

Figure 19.207 Nominal Values for Product Types in the Data Set Nomval

Prodtype	Nominal
M1	15.0
M2	15.5
M3	14.8
M4	15.2

To compute the differences from nominal, you must merge the data with the nominal values. You can do this with the following SAS statements. Note that an IN= variable is used in the MERGE statement to allow for

the fact that Nomval includes nominal values for product types that are not represented in Old. Figure 19.208 lists the merged data set Old.

```
proc sort data=Old;
   by Prodtype;
run;

data Old;
   format Diff 5.2 ;
   merge Nomval Old(in = a);
      by Prodtype;
   if a;
   Diff = Diameter - Nominal;
run;

proc sort data=Old;
   by Sample;
run;
```

Figure 19.208 Data Merged with Nominal Values

Sample	Prodtype	Diameter	Nominal	Diff
1	M3	13.99	14.8	-0.81
2	M3	14.69	14.8	-0.11
3	M3	13.86	14.8	-0.94
4	M3	14.32	14.8	-0.48
5	M3	13.23	14.8	-1.57
6	M1	17.55	15.0	2.55
7	M1	14.26	15.0	-0.74
8	M1	14.62	15.0	-0.38
9	M1	12.97	15.0	-2.03
10	M2	16.18	15.5	0.68
11	M2	15.29	15.5	-0.21
12	M2	16.20	15.5	0.70
13	M3	13.89	14.8	-0.91
14	M3	12.71	14.8	-2.09
15	M3	14.32	14.8	-0.48
16	M3	15.35	14.8	0.55
17	M2	15.08	15.5	-0.42
18	M2	14.72	15.5	-0.78
19	M2	14.79	15.5	-0.71
20	M2	15.27	15.5	-0.23
21	M2	15.95	15.5	0.45
22	M1	14.78	15.0	-0.22
23	M1	15.19	15.0	0.19
24	M1	15.41	15.0	0.41
25	M1	16.26	15.0	1.26
26	M3	16.68	14.8	1.88
27	M3	15.60	14.8	0.80
28	M3	14.86	14.8	0.06
29	M3	16.67	14.8	1.87
30	M3	14.35	14.8	-0.45

Assume that the variability in the process is constant across product types. To estimate the common process standard deviation σ, you first estimate σ for each product type based on the average of the moving ranges of the differences from nominal. You can do this in several steps, the first of which is to sort the data and compute the average moving range with the SHEWHART procedure.

```
proc sort data=Old;
   by Prodtype;
run;

proc shewhart data=Old;
   irchart Diff*Sample /
      nochart
      outlimits=Baselim;
   by Prodtype;
run;
```

The purpose of this procedure step is simply to save the average moving range for each product type in the OUTLIMITS= data set Baselim, which is listed in Figure 19.209 (note that Prodtype is specified as a BY variable).

Figure 19.209 Values of $\bar{R}$ by Product Type

Control Limits By Product Type

Prodtype	_VAR_	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_
M1	Diff	Sample	ESTIMATE	2	.002699796	3	-3.13258	0.13000
M2	Diff	Sample	ESTIMATE	2	.002699796	3	-1.77795	-0.06500
M3	Diff	Sample	ESTIMATE	2	.002699796	3	-3.22641	-0.19143

UCLI	_LCLR_	_R_	_UCLR_	_STDDEV_
3.39258	0	1.22714	4.00850	1.08753
1.64795	0	0.64429	2.10458	0.57098
2.84356	0	1.14154	3.72887	1.01166

To obtain a combined estimate of σ, you can use the MEANS procedure to average the average ranges in Baselim and then divide by the unbiasing constant d_2.

```
proc means data=Baselim noprint;
   var _r_;
   output out=Difflim (keep=_r_) mean=_r_;
run;

data Difflim;
   set Difflim;
   drop _r_;
   length _var_ _subgrp_ $ 8;
   _var_    = 'Diff';
   _subgrp_ = 'Sample';
   _mean_   = 0.0;
   _stddev_ = _r_ / d2(2);
   _limitn_ = 2;
   _sigmas_ = 3;
run;
```

The data set Difflim is structured for subsequent use by the SHEWHART procedure as an input LIMITS= data set. The variables in a LIMITS= data set provide pre-computed control limits or—as in this case—the parameters from which control limits are to be computed. These variables have reserved names that begin and end with the underscore character. Here, the variable _STDDEV_ saves the estimate of σ, and the variable _MEAN_ saves the mean of the differences from nominal. Recall that this mean is zero, because the nominal values are assumed to represent the process mean for each product type. The identifier variables _VAR_ and _SUBGRP_ record the names of the process and subgroup variables (these variables are critical in applications involving many product types). The variable _LIMITN_ is assigned a value of 2 to specify moving ranges of two consecutive measurements, and the variable _SIGMAS_ is assigned a value of 3 to specify 3σ limits. The data set Difflim is listed in Figure 19.210.

Figure 19.210 Estimates of Mean and Standard Deviation

Control Limit Parameters For Differences

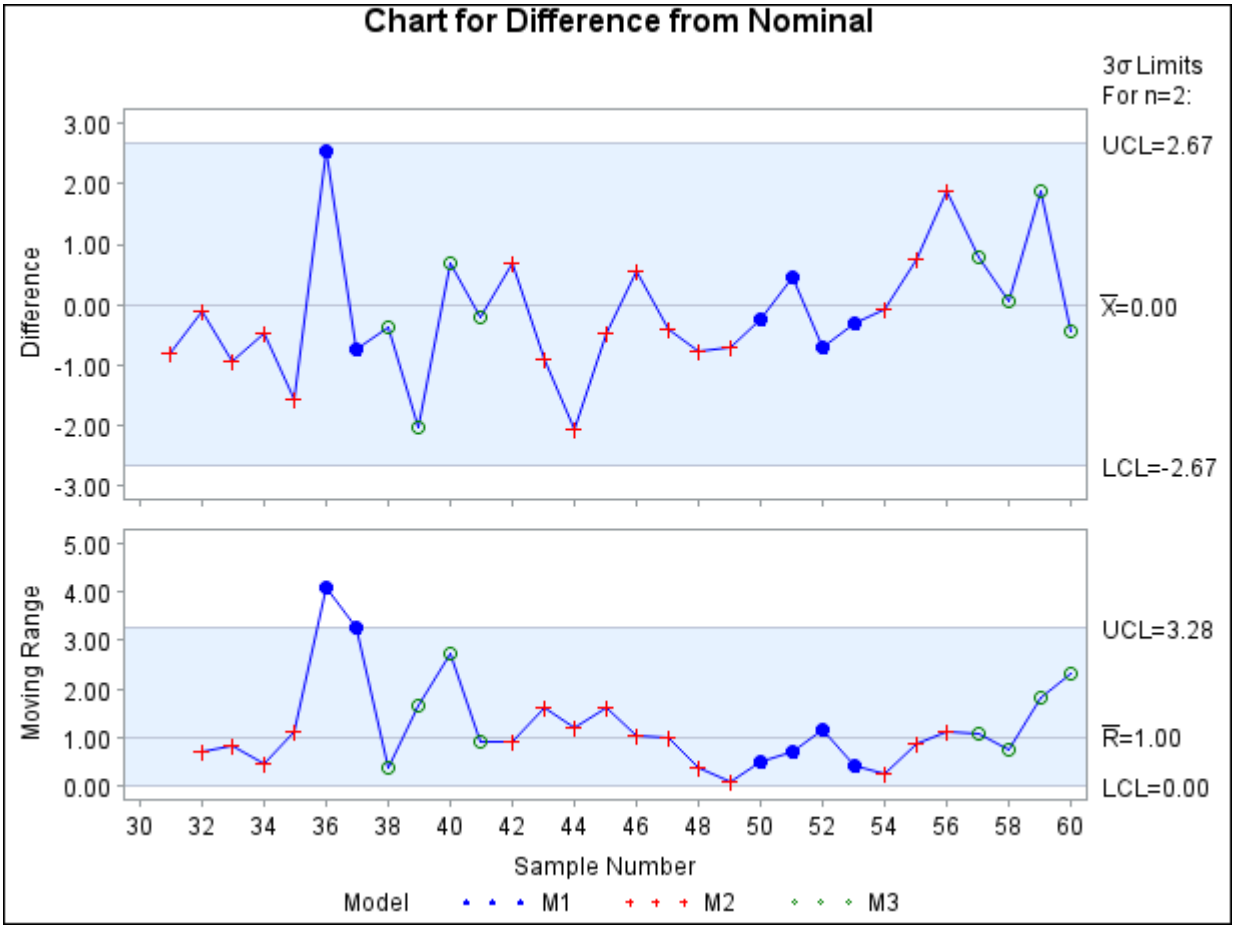
var	_subgrp_	_mean_	_stddev_	_limitn_	_sigmas_
Diff	Sample	0	0.89006	2	3

Now that the control limit parameters are saved in Difflim, diameters for an additional 30 parts (samples 31 to 60) are measured and saved in a SAS data set named New. You can construct short run control charts for this data by merging the measurements in New with the corresponding nominal values in Nomval, computing the differences from nominal, and then constructing the short run individual measurements and moving range charts.

```
proc sort data=new;
   by Prodtype;
run;

data new;
   format Diff 5.2 ;
   merge Nomval new(in = a);
      by Prodtype;
   if a;
   Diff = Diameter - Nominal;
   label Sample   = 'Sample Number'
         Prodtype = 'Model';
run;

proc sort data=new;
   by Sample;
run;

ods graphics off;
symbol1 v=dot    c=blue  h=3.0 pct;
symbol2 v=plus   c=red   h=3.0 pct;
symbol3 v=circle c=green h=3.0 pct;
title 'Chart for Difference from Nominal';
proc shewhart data=new limits=Difflim;
   irchart Diff*Sample=Prodtype / split    = '/';
   label Diff = 'Difference/Moving Range';
run;
```

The chart is displayed in Figure 19.211. Note that the product types are identified with symbol markers as requested by specifying Prodtype as a *symbol-variable*.

Figure 19.211 Short Run Control Chart

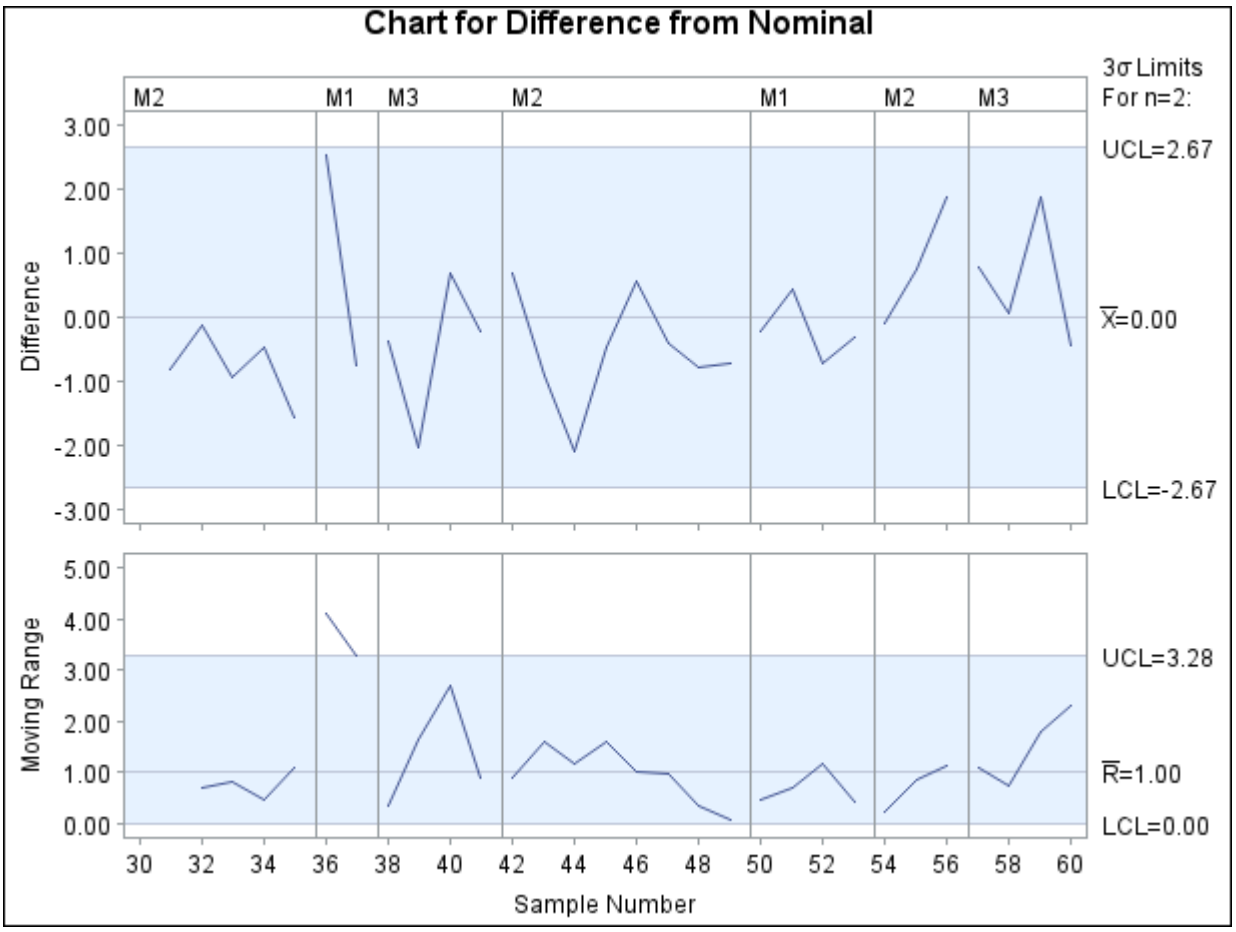


You can also identify the product types with a legend by specifying Prodtype as a _PHASE_ variable.

```
symbol h=3.0 pct;
title 'Chart for Difference from Nominal';
proc shewhart data=new (rename=(Prodtype=_phase_)) limits=Difflim;
   irchart Diff*Sample /
      readphases = all
      phaseref
      phasebreak
      phaselegend
      split      = '/';
   label Diff = 'Difference/Moving Range';
run;
```

The display is shown in Figure 19.212. Note that the PHASEBREAK option is used to suppress the connection of adjacent points in different phases (product types).

Figure 19.212 Identification of Product Types



In some applications, it might be useful to replace the moving range chart with a plot of the nominal values. You can do this with the TRENDVAR= option in the XCHART statement[27] provided that you reset the value of _LIMITN_ to 1 to specify a subgroup sample of size one.

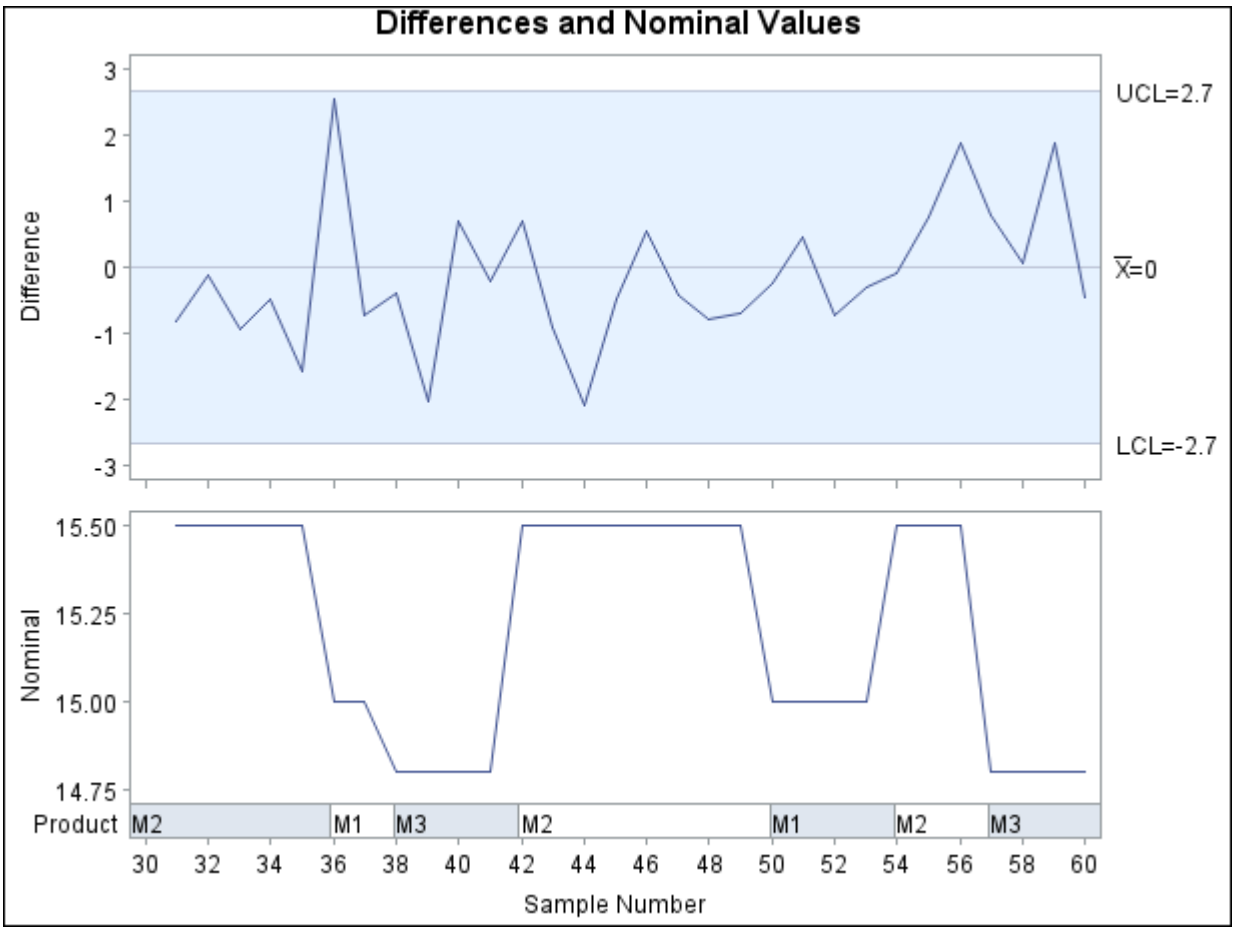
```
data Difflim;
   set Difflim;
   _var_    = 'Diameter';
   _limitn_ = 1;
run;

title 'Differences and Nominal Values';
proc shewhart data=new limits=Difflim;
   xchart Diameter*Sample (Prodtype) /
      nolimitslegend
      nolegend
      split         = '/'
      blockpos      = 3
      blocklabtype  = scaled
      blocklabelpos = left
      xsymbol       = xbar
```

[27] The TRENDVAR= option is not available in the IRCHART statement.

```
        trendvar      = Nominal;
    label Diameter    = 'Difference/Nominal'
          Prodtype    = 'Product';
run;
```

The display is shown in Figure 19.213. Note that you identify the product types by specifying Prodtype as a *block variable* enclosed in parentheses after the subgroup variable Sample. The BLOCKLABTYPE= option specifies that values of the block variable are to be scaled (if necessary) to fit the space available in the block legend. The BLOCKLABELPOS= option specifies that the label of the block variable is to be displayed to the left of the block legend.

Figure 19.213 Short Run Control Chart with Nominal Values



Testing for Constant Variances

The difference-from-nominal chart should be accompanied by a test that checks whether the variances for each product type are identical (homogeneous). Levene's test of homogeneity is particularly appropriate for short run applications because it is robust to departures from normality; refer to Snedecor and Cochran (1980). You can implement Levene's method by using the GLM procedure to construct a one-way analysis of variance for the absolute deviations of the diameters from averages within product types.

```
proc sort data=Old;
   by Prodtype;
run;

proc means data=Old noprint;
   var Diameter;
   by Prodtype;
   output out=Oldmean (keep=Prodtype diammean) mean=diammean;
run;

data Old;
   merge Old Oldmean;
      by Prodtype;
   absdev = abs( Diameter - diammean );
run;

proc means data=Old noprint;
  var absdev;
  by Prodtype;
  output out=stats n=n mean=mean css=css std=std;
run;

title;
proc glm data=Old outstat=glmout;
   class Prodtype;
   model absdev = Prodtype;
run;
```

A partial listing of the results is displayed in Figure 19.214. The large p-value (0.3386) indicates that the data do not reject the hypothesis of homogeneity.

Figure 19.214 Levene's Test of Variance Homogeneity

The GLM Procedure

Dependent Variable: absdev

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	1.02901063	0.51450532	1.13	0.3373
Error	27	12.27381243	0.45458565		
Corrected Total	29	13.30282306			

Standardizing Differences from Nominal

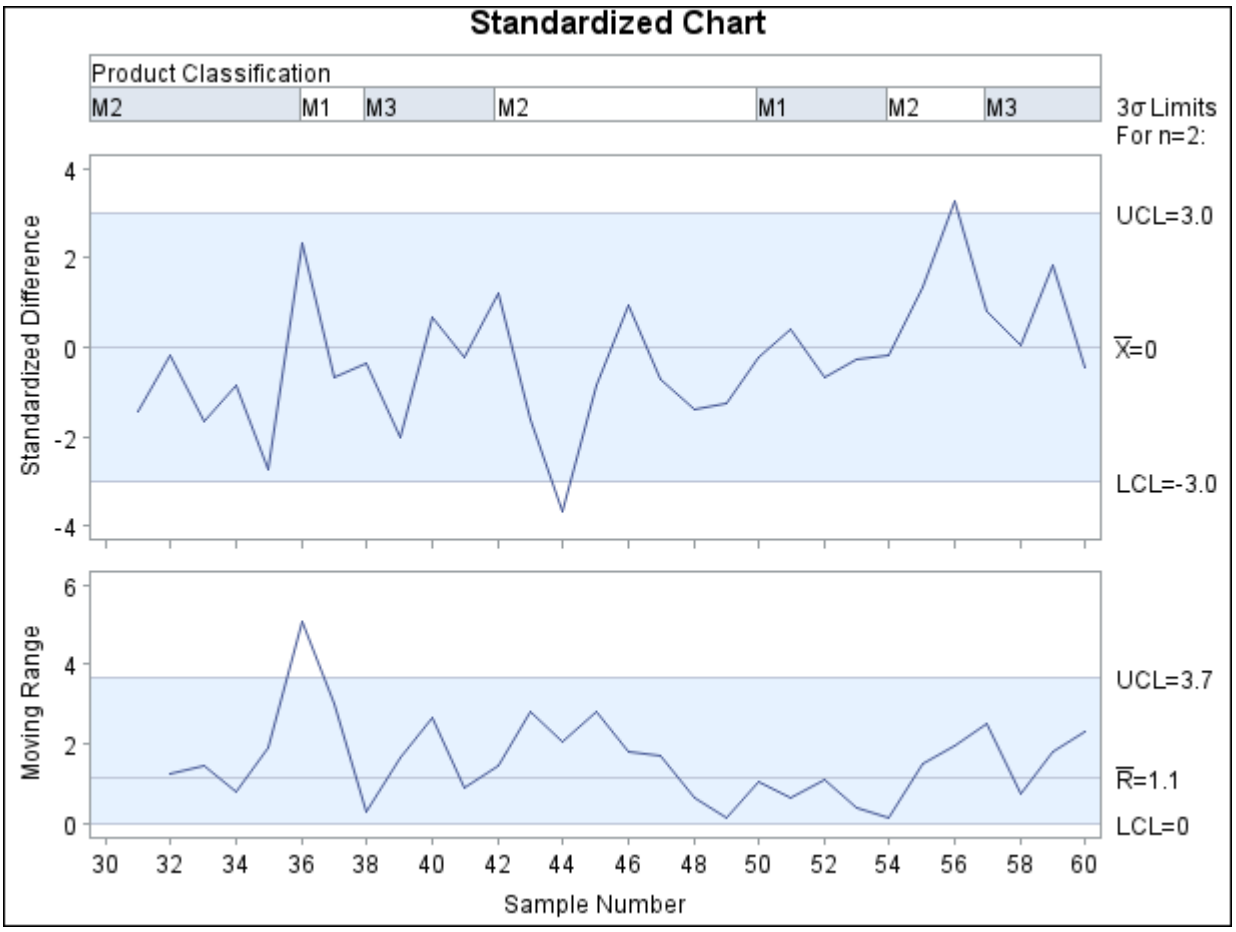
When the variances across product types are *not* constant, various authors recommend standardizing the differences from nominal and displaying them on a common chart with control limits at ± 3.

To illustrate this method, assume that the hypothesis of homogeneity is rejected for the differences in Old. Then you can use the product-specific estimates of σ in Baselim to standardize the differences from nominal in New and create the standardized chart as follows:

```
proc sort data=new;
   by Prodtype;
run;

data new;
   keep Sample Prodtype z Diff Diameter Nominal _stddev_;
   label Sample = 'Sample Number';
   format Diff 5.2 ;
   merge Baselim new(in = a);
      by Prodtype;
   if a;
   z = (Diameter - Nominal) / _stddev_ ;
run;

proc sort data=new;
   by Sample;
run;

title 'Standardized Chart';
proc shewhart data=new;
   irchart z*Sample (Prodtype) /
      blocklabtype  = scaled
      mu0           = 0
      sigma0        = 1
      split         = '/';
   label Prodtype = 'Product Classification'
         z = 'Standardized Difference/Moving Range';
run;
```

Note that the options MU0= and SIGMA= specify that the control limits for the standardized differences from nominal are to be based on the parameters $\mu = 0$ and $\sigma = 1$. The chart is displayed in Figure 19.215.

Figure 19.215 Standardized Difference Chart



Nonnormal Process Data

NOTE: See *Nonnormal Process Data* in the SAS/QC Sample Library.

A number of authors have pointed out that Shewhart charts for subgroup means work well whether the measurements are normally distributed or not.[28] On the other hand, the interpretation of standard control charts for individual measurements (X charts) is affected by departures from normality.

In situations involving a large number of measurements, it might be possible to subgroup the data and construct an $\bar{X}$ chart instead of an X chart. However, the measurements should not be subgrouped arbitrarily for this purpose.[29] If subgrouping is not possible, two alternatives are to transform the data to normality (preferably with a simple transformation such as the log transformation) or modify the usual limits based on a suitable model for the data distribution.

The second of these alternatives is illustrated here with data from a study conducted by a service center. The time taken by staff members to answer the phone was measured, and the delays were saved as values of a variable named Time in a SAS data set named Calls. A partial listing of Calls is shown in Figure 19.216.

[28] Refer to Schilling and Nelson (1976) and Wheeler (1991).

[29] Refer to Wheeler and Chambers (1986) for a discussion of subgrouping.

Figure 19.216 Answering Times from the Data Set Calls

Recnum	Time
1	3.233
2	3.110
3	3.136
4	2.899
5	2.838
6	2.459
7	3.716
8	2.740
9	2.487
10	2.635
11	2.676
12	2.905
13	3.431
14	2.663
15	3.437
16	2.823
17	2.596
18	2.633
19	3.235
20	2.701
21	3.202
22	2.725
23	3.151
24	2.464
25	2.662
26	3.188
27	2.640
28	2.541
29	3.033
30	2.993
31	2.636
32	2.481
33	3.191
34	2.662
35	2.967
36	3.300
37	2.530
38	2.777
39	3.353
40	3.614
41	4.288
42	2.442
43	2.552
44	2.613
45	2.731
46	2.780
47	3.588
48	2.612

Figure 19.216 *continued*

Recnum	Time
49	2.579
50	2.871

Creating a Preliminary Individual Measurements Chart

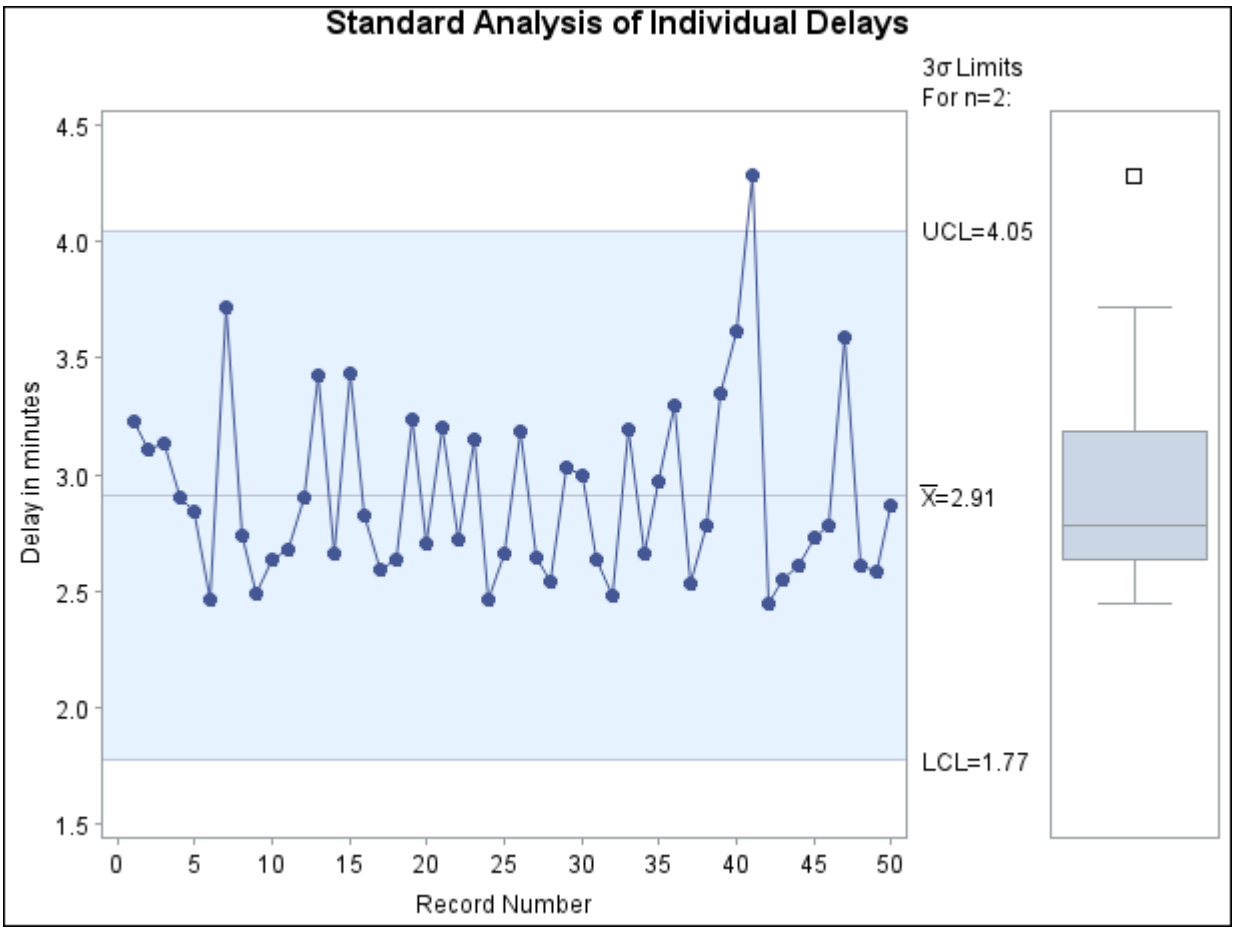
As a first step, the delays were analyzed using an *X* chart created with the following statements. The chart is displayed in Figure 19.217.

```
ods graphics off;
title 'Standard Analysis of Individual Delays';
proc shewhart data=Calls;
   irchart Time * Recnum /
      rtmplot   = schematic
      outlimits = delaylim
      nochart2 ;
   label Recnum = 'Record Number'
         Time   = 'Delay in minutes' ;
run;
```

You might be inclined to conclude that the 41st point signals a special cause of variation. However, the box plot in the right margin (requested with the RTMPLOT= option) indicates that the distribution of delays is skewed. Thus, the reason that the measurements are grouped well within the control limits is that the limits are incorrect and not that the process is too good for the limits.

NOTE: This example assumes the process is in statistical control; otherwise, the box plot could not be interpreted as a representation of the process distribution. You can check the assumption of normality with goodness-of-fit tests by using the CAPABILITY procedure, as shown in the statements that follow.

Figure 19.217 Standard Control Limits for Delays



Calculating Probability Limits

The OUTLIMITS= option saves the control limits from the chart in Figure 19.217 in a SAS data set named delaylim, which is listed in Figure 19.218.

Figure 19.218 Control Limits for Standard Chart from the Data Set Calls

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_	_UCLI_	_STDDEV_
Time	Recnum	ESTIMATE	2	.002699796	3	1.77008	2.91038	4.05068	0.38010

The control limits can be replaced with the corresponding percentiles from a fitted lognormal distribution. The equation for the lognormal density function is

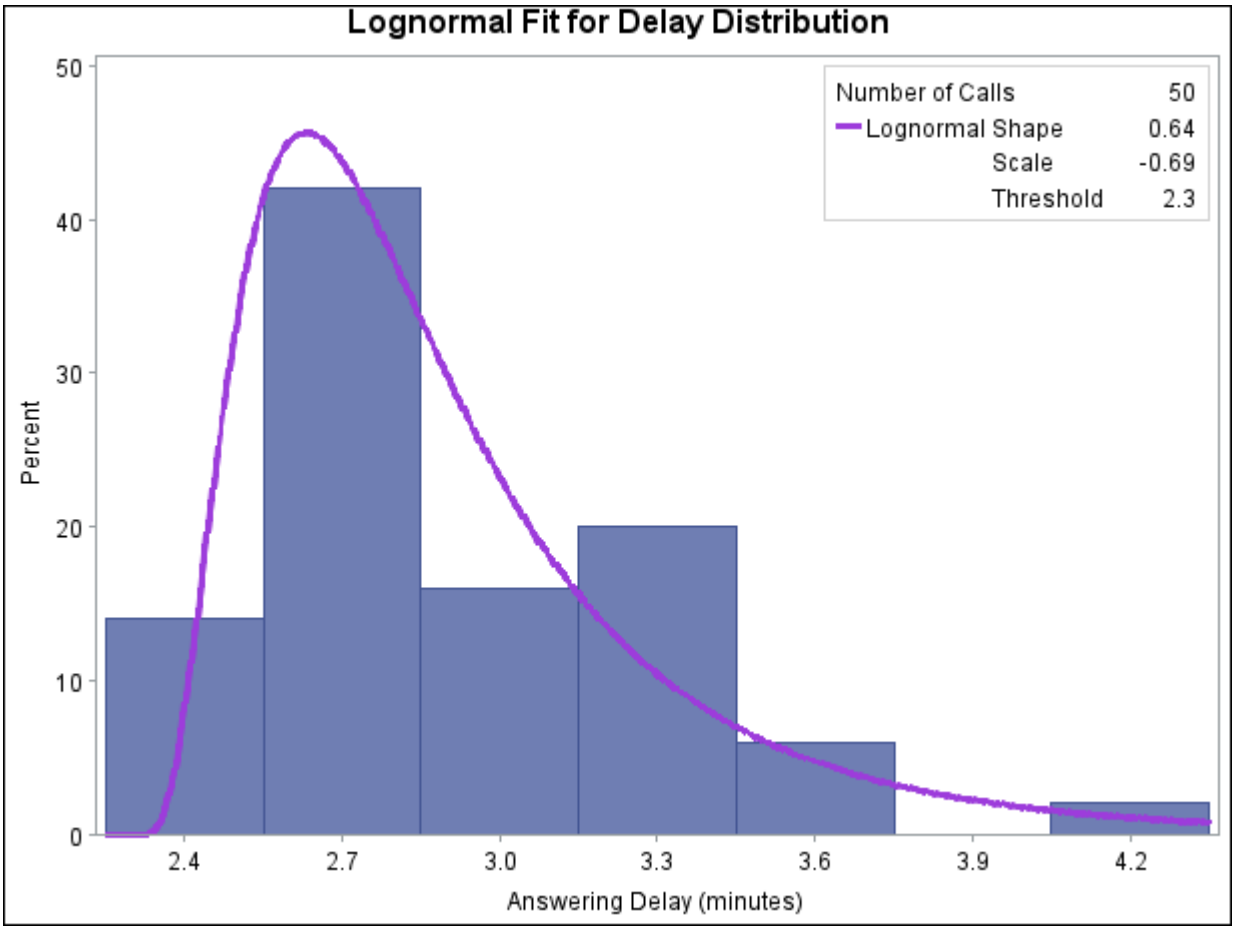
$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} \exp\left(-\frac{(\log(x)-\zeta)^2}{2\sigma^2}\right) \quad x > 0$$

where σ denotes the shape parameter and ζ denotes the scale parameter.

The following statements use the CAPABILITY procedure to fit a lognormal model and superimpose the fitted density on a histogram of the data, shown in Figure 19.219:

```
title 'Lognormal Fit for Delay Distribution';
proc capability data=Calls noprint;
   histogram Time /
      lognormal(threshold=2.3 w=2)
      outfit = Lnfit
      nolegend ;
   inset n = 'Number of Calls'
      lognormal( sigma = 'Shape' (4.2)
                 zeta  = 'Scale' (5.2)
                 theta ) / pos    = ne;
   label Time  = 'Answering Delay (minutes)';
run;
```

Figure 19.219 Distribution of Delays



Parameters of the fitted distribution and results of goodness-of-fit tests are saved in the data set Lnfit, which is listed in Figure 19.220. The large *p*-values for the goodness-of-fit tests are evidence that the lognormal model provides a good fit.

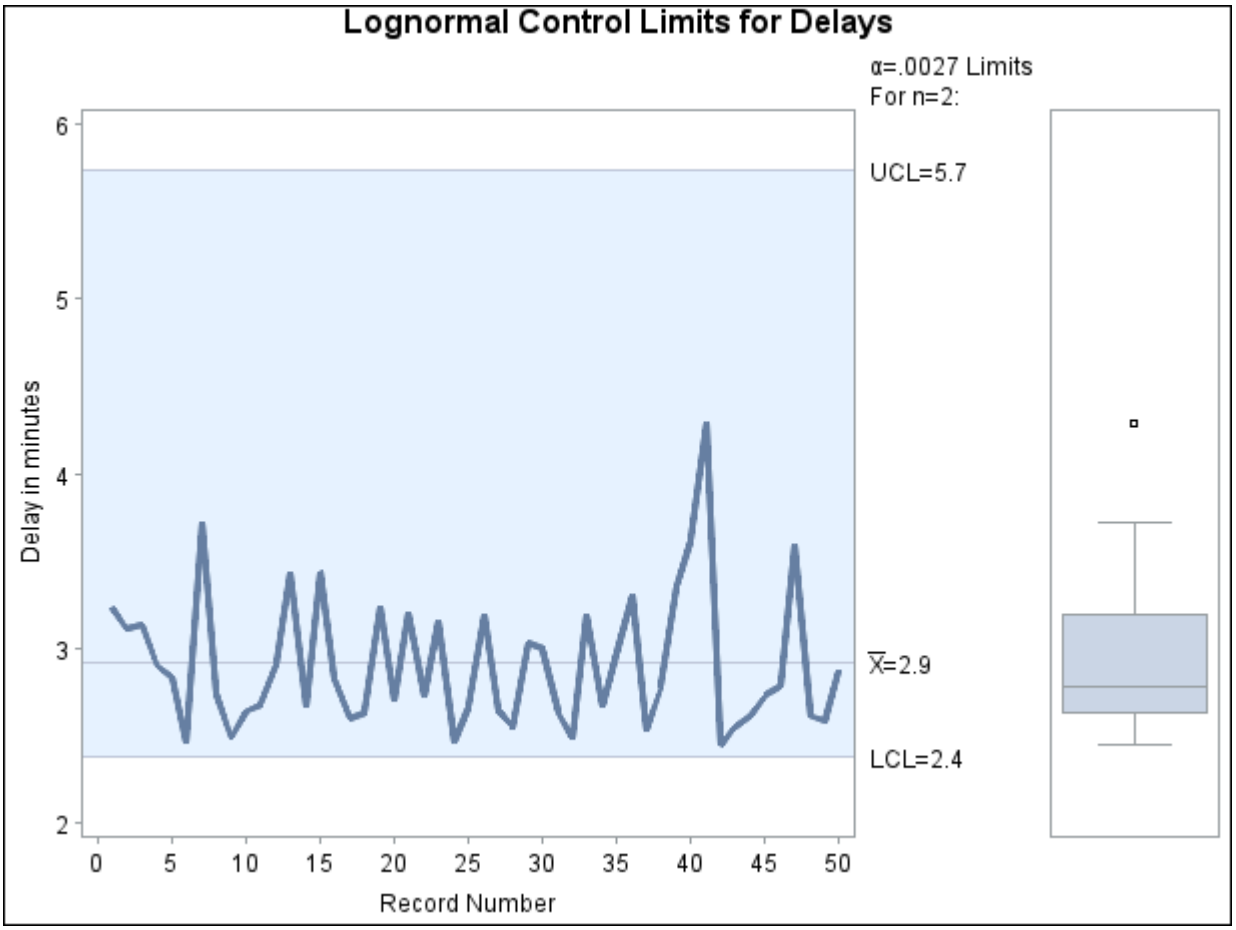
Figure 19.220 Parameters of Fitted Lognormal Model in the Data Set Lnfit

VAR	_CURVE_	_LOCATN_	_SCALE_	_SHAPE1_	_MIDPTN_	_ADASQ_	_ADP_	_CVMWSQ_	_CVMP_	_KSD_	_KSP_
Time	LNORMAL	2.3	-0.68910	0.64110	4.2	0.34854	0.47465	0.058737	0.40952	0.092223	0.15

The following statements replace the control limits in DELAYLIM with limits computed from percentiles of the fitted lognormal model. The 100αth percentile of the lognormal distribution is $P_\alpha = \exp(\sigma\Phi^{-1}(\alpha) + \zeta)$, where Φ^{-1} denotes the inverse standard normal cumulative distribution function. The SHEWHART procedure constructs an X chart with the modified limits, displayed in Figure 19.221.

```
data delaylim;
   merge delaylim Lnfit;
   drop _sigmas_ ;
   _lcli_ = _locatn_ + exp(_scale_+probit(0.5*_alpha_)*_shape1_);
   _ucli_ = _locatn_ + exp(_scale_+probit(1-.5*_alpha_)*_shape1_);
   _mean_ = _locatn_ + exp(_scale_+0.5*_shape1_*_shape1_);
run;

title 'Lognormal Control Limits for Delays';
proc shewhart data=Calls limits=delaylim;
   irchart Time*Recnum /
      rtmplot   = schematic
      nochart2 ;
   label Recnum = 'Record Number'
         Time   = 'Delay in minutes' ;
run;
```

Figure 19.221 Adjusted Control Limits for Delays



Clearly the process is in control, and the control limits (particularly the lower limit) are appropriate for the data. The particular probability level $\alpha = 0.0027$ associated with these limits is somewhat immaterial, and other values of α such as 0.001 or 0.01 could be specified with the ALPHA= option in the original IRCHART statement.

Multivariate Control Charts

NOTE: See *Creating Multivariate Control Charts* in the SAS/QC Sample Library.

In many industrial applications, the output of a process characterized by p variables that are measured simultaneously. Independent variables can be charted individually, but if the variables are correlated, a multivariate chart is needed to determine whether the process is in control.

Many types of multivariate control charts have been proposed; refer to Alt (1985) for an overview. Denote the ith measurement on the jth variable as X_{ij} for $i = 1, 2, \ldots, n$, where n is the number of measurements, and $j = 1, 2, \ldots, p$. Standard practice is to construct a chart for a statistic T_i^2 of the form

$$T_i^2 = (\mathbf{X}_i - \bar{\mathbf{X}}_n)' \mathbf{S}_n^{-1} (\mathbf{X}_i - \bar{\mathbf{X}}_n)$$

where

$$\bar{X}_j = \frac{1}{n} \sum_{i=1}^{n} X_{ij} \quad , \quad \mathbf{X}_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{ip} \end{bmatrix}, \quad \bar{\mathbf{X}}_n = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_p \end{bmatrix}$$

and

$$\mathbf{S}_n = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{X}_i - \bar{\mathbf{X}}_n)(\mathbf{X}_i - \bar{\mathbf{X}}_n)'$$

It is assumed that $\mathbf{X}_i$ has a p-dimensional multivariate normal distribution with mean vector — $= (\mu_1 \mu_2 \cdots \mu_p)'$ and covariance matrix $\mathbf{\Sigma}$ for $i = 1, 2, \ldots, n$. Depending on the assumptions made about the parameters, a χ^2, Hotelling T^2, or beta distribution is used for T_i^2, and the percentiles of this distribution yield the control limits for the multivariate chart.

In this example, a multivariate control chart is constructed using a beta distribution for T_i^2. The beta distribution is appropriate when the data are individual measurements (rather than subgrouped measurements) and when — and $\mathbf{\Sigma}$ are estimated from the data being charted. In other words, this example illustrates a start-up phase chart where the control limits are determined from the data being charted.

Calculating the Chart Statistic

In this situation, it was shown by Gnanadesikan and Kettenring (1972), using a result of Wilks (1962), that T_i^2 is exactly distributed as a multiple of a variable with a beta distribution. Specifically,

$$T_i^2 \sim \frac{(n-1)^2}{n} \, B\left(\frac{p}{2}, \frac{n-p-1}{2}\right)$$

Tracy, Young, and Mason (1992) used this result to derive initial control limits for a multivariate chart based on three quality measures from a chemical process in the start-up phase: percent of impurities, temperature, and concentration. The remainder of this section describes the construction of a multivariate control chart using their data, which are given here by the data set Startup.

```
data Startup;
   input Sample Impure Temp Conc;
   label Sample = 'Sample Number'
         Impure = 'Impurities'
         Temp   = 'Temperature'
         Conc   = 'Concentration' ;
   datalines;
 1  14.92  85.77  42.26
 2  16.90  83.77  43.44
 3  17.38  84.46  42.74
 4  16.90  86.27  43.60
 5  16.92  85.23  43.18
 6  16.71  83.81  43.72
 7  17.07  86.08  43.33
 8  16.93  85.85  43.41
 9  16.71  85.73  43.28
10  16.88  86.27  42.59
11  16.73  83.46  44.00
12  17.07  85.81  42.78
13  17.60  85.92  43.11
14  16.90  84.23  43.48
;
```

In preparation for the computation of the control limits, the sample size is calculated and parameter variables are defined.

```
proc means data=Startup noprint ;
   var Impure Temp Conc;
   output out=means n=n;
run;

data Startup;
   if _n_ = 1 then set means;
   set Startup;
   p        = 3;
   _subn_   = 1;
   _limitn_ = 1;
run;
```

Next, the PRINCOMP procedure is used to compute the principal components of the variables and save them in an output data set named Prin.

```
proc princomp data=Startup out=Prin outstat=scores std cov;
   var Impure Temp Conc;
run;
```

The following statements compute T_i^2 and its exact control limits, using the fact that T_i^2 is the sum of squares of the principal components.[30] Note that these statements create several special SAS variables so that the data set Prin can subsequently be read as a TABLE= input data set by the SHEWHART procedure. These special variables begin and end with an underscore character. The data set Prin is listed in Figure 19.222.

```
data Prin (rename=(tsquare=_subx_));
   length _var_ $ 8 ;
   drop prin1 prin2 prin3 _type_ _freq_;
   set Prin;
   comp1   = prin1*prin1;
   comp2   = prin2*prin2;
   comp3   = prin3*prin3;
   tsquare = comp1 + comp2 + comp3;
   _var_   = 'tsquare';
   _alpha_ = 0.05;
   _lclx_  = ((n-1)*(n-1)/n)*betainv(_alpha_/2, p/2, (n-p-1)/2);
   _mean_  = ((n-1)*(n-1)/n)*betainv(0.5, p/2, (n-p-1)/2);
   _uclx_  = ((n-1)*(n-1)/n)*betainv(1-_alpha_/2, p/2, (n-p-1)/2);
   label tsquare = 'T Squared'
         comp1   = 'Comp 1'
         comp2   = 'Comp 2'
         comp3   = 'Comp 3';
run;
```

[30] Refer to Jackson (1980).

Figure 19.222 The Data Set Prin

T2 Chart For Chemical Example

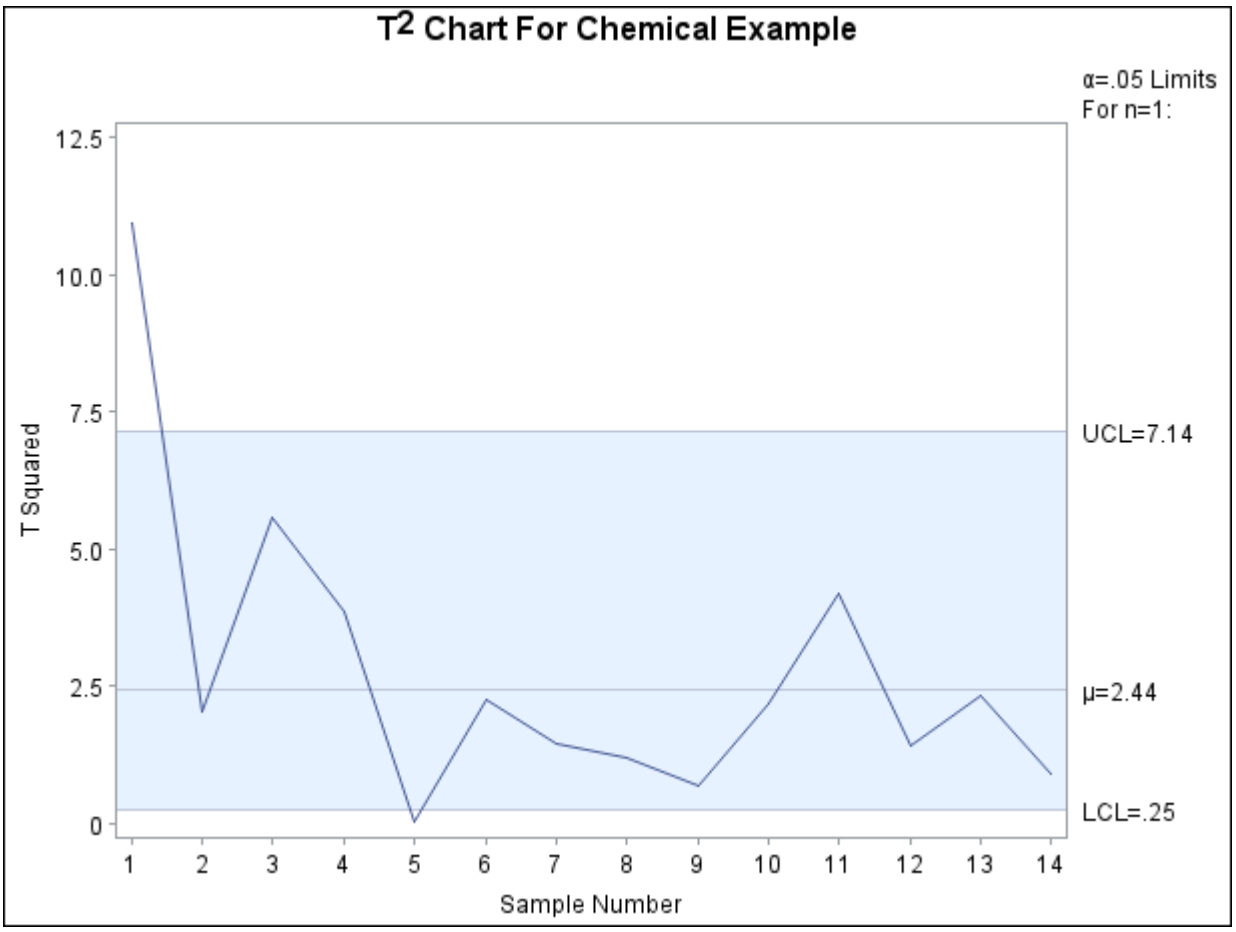
var	n	Sample	Impure	Temp	Conc	p	_subn_	_limitn_	comp1	comp2	comp3
tsquare	14	1	14.92	85.77	42.26	3	1	1	0.79603	10.1137	0.01606
tsquare	14	2	16.90	83.77	43.44	3	1	1	1.84804	0.0162	0.17681
tsquare	14	3	17.38	84.46	42.74	3	1	1	0.33397	0.1538	5.09491
tsquare	14	4	16.90	86.27	43.60	3	1	1	0.77286	0.3289	2.76215
tsquare	14	5	16.92	85.23	43.18	3	1	1	0.00147	0.0165	0.01919
tsquare	14	6	16.71	83.81	43.72	3	1	1	1.91534	0.0645	0.27362
tsquare	14	7	17.07	86.08	43.33	3	1	1	0.58596	0.4079	0.44146
tsquare	14	8	16.93	85.85	43.41	3	1	1	0.29543	0.1729	0.73939
tsquare	14	9	16.71	85.73	43.28	3	1	1	0.23166	0.0001	0.44483
tsquare	14	10	16.88	86.27	42.59	3	1	1	1.30518	0.0004	0.86364
tsquare	14	11	16.73	83.46	44.00	3	1	1	3.15791	0.0274	0.98639
tsquare	14	12	17.07	85.81	42.78	3	1	1	0.43819	0.0823	0.87976
tsquare	14	13	17.60	85.92	43.11	3	1	1	0.41494	1.6153	0.30167
tsquare	14	14	16.90	84.23	43.48	3	1	1	0.90302	0.0001	0.00010

subx	_alpha_	_lclx_	_mean_	_uclx_
10.9257	0.05	0.24604	2.44144	7.13966
2.0410	0.05	0.24604	2.44144	7.13966
5.5827	0.05	0.24604	2.44144	7.13966
3.8640	0.05	0.24604	2.44144	7.13966
0.0372	0.05	0.24604	2.44144	7.13966
2.2534	0.05	0.24604	2.44144	7.13966
1.4354	0.05	0.24604	2.44144	7.13966
1.2077	0.05	0.24604	2.44144	7.13966
0.6766	0.05	0.24604	2.44144	7.13966
2.1692	0.05	0.24604	2.44144	7.13966
4.1717	0.05	0.24604	2.44144	7.13966
1.4003	0.05	0.24604	2.44144	7.13966
2.3320	0.05	0.24604	2.44144	7.13966
0.9032	0.05	0.24604	2.44144	7.13966

You can now use the data set Prin as input to the SHEWHART procedure to create the multivariate control chart displayed in Figure 19.223.

```
ods graphics off;
title 'T' m=(+0,+0.5) '2'
          m=(+0,-0.5) ' Chart For Chemical Example';
proc shewhart table=Prin;
   xchart tsquare*Sample /
      xsymbol  = mu
      nolegend ;
run;
```

Figure 19.223 Multivariate Control Chart for Chemical Process



The methods used in this example easily generalize to other types of multivariate control charts. You can create charts using the χ^2 and F distributions by using the appropriate CINV or FINV function in place of the BETAINV function. For details, refer to Alt (1985), Jackson (1980, 1991), and Ryan (1989).

Examining the Principal Component Contributions

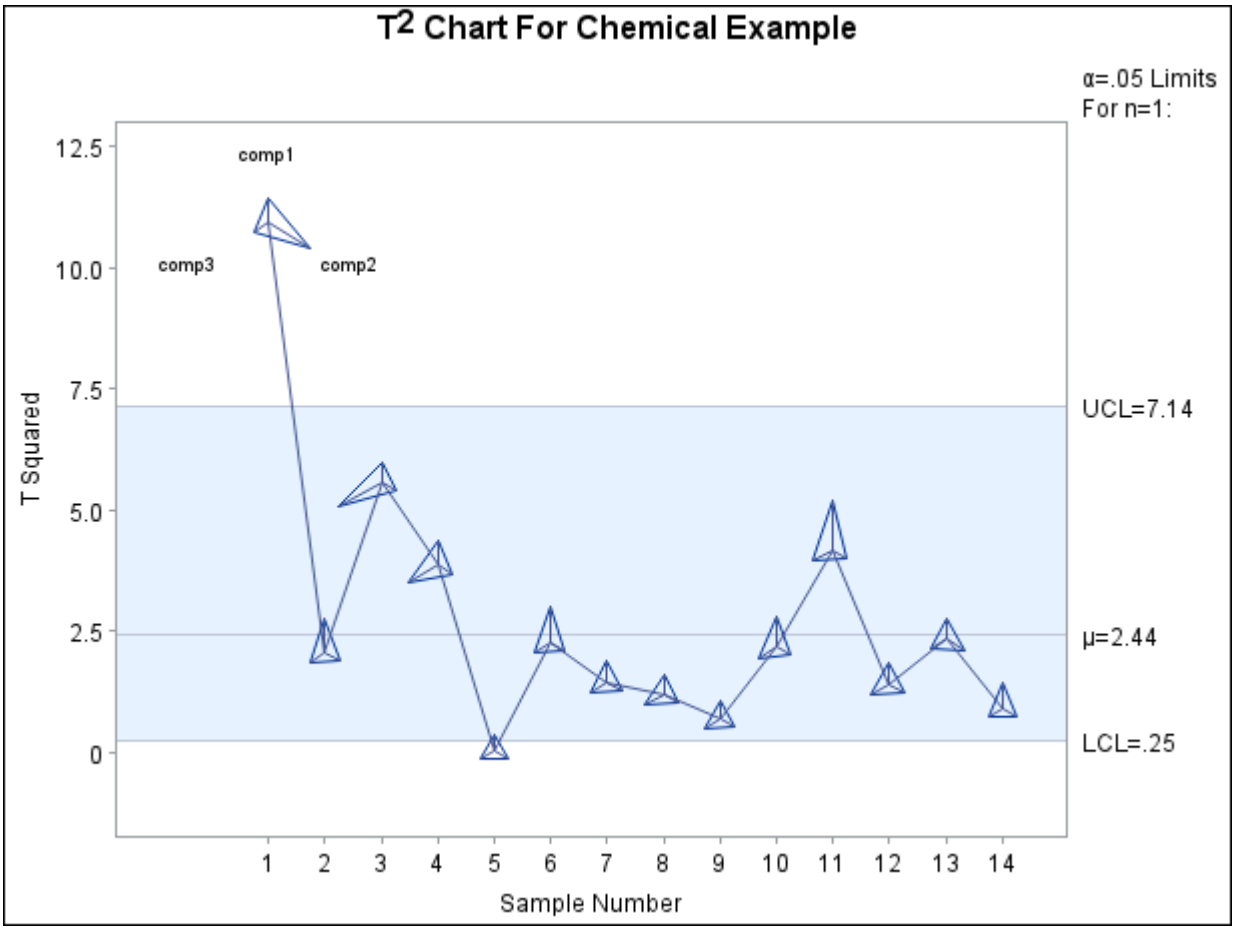
You can use the *star options* in the SHEWHART procedure to superimpose points on the chart with stars whose vertices represent standardized values of the squares of the three principal components used to determine T_i^2.

```
title 'T' m=(+0,+0.5) '2'
          m=(+0,-0.5) ' Chart For Chemical Example';
symbol value=none;
proc shewhart table=Prin;
   xchart tsquare*Sample /
      starvertices  = (comp1 comp2 comp3)
      startype      = wedge
      starlegend    = none
      starlabel     = first
      staroutradius = 4
      npanelpos     = 14
```

```
        xsymbol       = mu
        nolegend ;
run;
```

The chart is displayed in Figure 19.224. In situations where the principal components have a physical interpretation, the star chart can be a helpful diagnostic for determining the relative contributions of the different components.

Figure 19.224 Multivariate Control Chart Displaying Principal Components



For more information about star charts, see the section "Displaying Auxiliary Data with Stars" on page 2090, or consult the entries for the STARVERTICES= and related options in "Dictionary of Options: SHEWHART Procedure" on page 1993.

Principal components are not the only approach that can be used to interpret multivariate control charts. This problem has recently been studied by a number of authors, including Doganaksoy, Faltin, and Tucker (1991), Hawkins (1991, 1993), and Mason, Tracy, and Young (1993).

Interactive Control Charts: SHEWHART Procedure

Overview: Interactive Control Charts

This section describes two approaches for creating an interactive control chart which enables an end user to "drill down" into subgroup data points and display information not contained in the chart itself. For example, the end user might want to be able to click on a subgroup to

- list the individual measurements in the subgroup
- diagnose an out-of-control point by viewing a Pareto chart of the most common problems affecting the process
- view a list of recommended corrective actions
- trace the raw materials used to manufacture a batch of product

The two approaches for creating interactive control charts are as follows:

- saving graphics coordinate data from control charts for use in creating SAS/AF applications
- associating Uniform Resource Locators (URLs) with subgroups to produce "clickable" control charts in HTML

The options described in this section can be specified in all the chart statements available in the SHEWHART procedure.

Details: Interactive Control Charts

Saving Graphics Coordinates in a Control Chart

You can specify an WEBOUT= data set in any chart statement to save graphics coordinate information for a control chart. The WEBOUT= data set is an extension of the OUTTABLE= data set, which contains the subgroup summary statistics, control limits and related information found in an OUTTABLE= data set, as well as coordinate data. The additional coordinate variables are listed in Table 19.102.

Table 19.102 WEBOUT= Data Set

Variable	Description
_X1_	x-coordinate of lower left corner of primary chart subgroup bounding box
_Y1_	y-coordinate of lower left corner of primary chart subgroup bounding box
_X2_	x-coordinate of upper right corner of primary chart subgroup bounding box
_Y2_	y-coordinate of upper right corner of primary chart subgroup bounding box
_X*n*_	x-coordinate for point *n* of the subgroup shape
_Y*n*_	y-coordinate for point *n* of the subgroup shape

Table 19.102 (continued)

Variable	Description
_X1_2_	x-coordinate of lower left corner of secondary chart subgroup bounding box
_Y1_2_	y-coordinate of lower left corner of secondary chart subgroup bounding box
_X2_2_	x-coordinate of upper right corner of secondary chart subgroup bounding box
_Y2_2_	y-coordinate of upper right corner of secondary chart subgroup bounding box
SHAPE	shape of primary chart subgroup bounding area
NXY	number of points defining primary chart subgroup bounding area
GRAPH	name of primary chart graphics entry
GRAPH2	name of secondary chart graphics entry
DXMIN	value of lowest major tick mark on horizontal axis
DXMAX	value of highest major tick mark on horizontal axis
XMIN	x-coordinate of lowest major tick mark on horizontal axis
XMAX	x-coordinate of highest major tick mark on horizontal axis
DYMIN	value of lowest major tick mark on vertical axis
DYMAX	value of highest major tick mark on vertical axis
YMIN	y-coordinate of lowest major tick mark on vertical axis
YMAX	y-coordinate of highest major tick mark on vertical axis
XMIN2	x-coordinate of lowest major tick mark on secondary chart horizontal axis
XMAX2	x-coordinate of highest major tick mark on secondary chart horizontal axis
DYMIN2	value of lowest major tick mark on secondary chart vertical axis
DYMAX2	value of highest major tick mark on secondary chart vertical axis
YMIN2	y-coordinate of lowest major tick mark on secondary chart vertical axis
YMAX2	y-coordinate of highest major tick mark on secondary chart vertical axis

You can use the coordinate data saved in the WEBOUT= data set to create a "clickable" control chart in a SAS/AF application. The variables _X1_, _Y1_, _X2_ and _Y2_ contain the coordinates of the lower left and upper right corners of a rectangular *bounding box* associated with each subgroup on the primary chart. This box defines the clickable area associated with the subgroup when the chart is incorporated into a SAS/AF application. It contains the symbol used to plot the subgroup data, or the junction of line segments representing the subgroup if no plotting symbol is used. The variables _X1_2_, _Y1_2_, _X2_2_ and _Y2_2_ contain coordinates of the corners of subgroup bounding boxes for a secondary chart.

If you use the BOXCHART statement, each subgroup is represented by a box-and-whisker plot rather than a single symbol. The subgroup's bounding box is defined by the sides of the box-and-whisker plot and its lower and upper quartiles, regardless of the BOXSTYLE= value in effect.

If you specify the STARVERTICES= option, each subgroup is represented by a polygon or star with a vertex corresponding to each of the STARVERTICES= variables. The clickable area for a subgroup is the polygon with these vertices, regardless of the STARTYPE= value specified. In the WEBOUT= data set the value of the _SHAPE_ variable is POLY and the _NXY_ variable contains the number of vertices in the polygon. The variables _X*n*_ and _Y*n*_, where $n = 1$ to the value of _NXY_, contain the coordinates of the vertices of a subgroup's polygon. When the STARVERTICES= option is not used, the value of _SHAPE_ is always RECT and the value of _NXY_ is always 2.

When a control chart spans multiple panels (pages), the panels reside in separate SAS graphics entries. The _GRAPH_ character variable records the name of the graphics entry containing the panel on which a given subgroup is plotted. This is the same name that appears in the PROC GREPLAY menu. When the SEPARATE option is used, primary and secondary charts are displayed on different graphics entries. The _GRAPH2_ variable records the name of the graphics entry containing the secondary chart panel where a subgroup appears. When the SEPARATE option is not used, the values of _GRAPH_ and _GRAPH2_ will be the same for a given subgroup.

The variables _DXMIN_, _DXMAX_, _XMIN_ and _XMAX_ provide the data values and graphics coordinates associated with the lowest and highest major tick marks on the horizontal (subgroup) axis. The variables _DYMIN_, _DYMAX_, _YMIN_ and _YMAX_ provide the analogous values for the vertical axis. Through a simple linear transformation in your SAS/AF application you can use this information to convert from percent screen units to "data" units and vice versa.

The variables _XMIN2_ and _XMAX2_ contain the graphics coordinates associated with the lowest and highest major tick marks on the horizontal axis of a secondary chart. No variables for the corresponding data values are required, because they are always identical to those for the primary chart.

The variables _DYMIN2_, _DYMAX2_, _YMIN2_ and _YMAX2_ contain the data and coordinate values for the lowest and highest tick marks on the vertical axis of a secondary chart. A SAS/AF program receives the (x,y) coordinates for the location of the cursor when the user clicks on a subgroup data point. The application can determine whether (x,y) lies within any of the boxes whose coordinates are saved in the WEBOUT= data set. If so, the program can determine which subgroup was selected on the primary or secondary chart and can check the _TESTS_ and _TESTS2_ variables included in the WEBOUT= data set to determine whether an out-of-control condition has been signaled.

Notes:

1. Graphics coordinates are scaled in percent screen units from 0 to 100, where (0,0) represents the lower-left corner of the screen and (100,100) represents the upper-right corner of the screen. Because SAS/AF applications define the origin of the vertical axis at the top of the screen, it will be necessary to subtract the y-coordinates from 100 in your SCL program.

2. The variables _X1_2_, _Y1_2_, _X2_2_, _Y2_2_, _GRAPH2_, _XMIN2_, _XMAX2_, _YMIN2_, _YMAX2_, _DYMIN2_ and _DYMAX2_ appear in the WEBOUT= data set only when a secondary chart is produced. A secondary chart is produced by the IRCHART, MRCHART, XRCHART and XSCHART statements and by the BOXCHART, MCHART and XCHART statements when the TRENDVAR= option is specified.

3. When the subgroup variable is a character variable, the value of _DXMIN_ is zero and the value of _DXMAX_ is the number of subgroups in the input data set minus one.

4. A bounding box circumscribes a point displayed on a chart and its dimensions depend on the size of the symbol marker used to display the point. If no symbol marker is specified, a small default size is used for the box. If a large number of subgroups are displayed on a panel, the subgroup symbols might overlap, so it is possible for a user to inadvertently select more than one point.

Associating URLs with Subgroups in HTML

You can use the Output Delivery System (ODS) to produce an HTML file containing a control chart created by the SHEWHART procedure. The HTML= option provides a way to associate Uniform Resource Locators (URLs) with subgroups plotted on a control chart. It specifies a variable in the input data set containing HTML syntax providing the URLs to be associated with different subgroups. The HTML= variable can be a character variable or a numeric variable with an associated character format.

The following statements generate an $\bar{X}$ chart that is saved to a GIF file and included in an HTML file. The formatted values of the numeric HTML= variable Web specify URLs that link subgroups in the input data set to various web pages.

```
goptions target = gif;
ods html body = "example1.html";
proc format;
   value webfmt
   1='href="http://www.sas.com/"'
   2='href="http://www.sas.com/service/techsup/faq/qc/shewproc.html"'
   3='href="http://www.sas.com/rnd/app/qc.html"'
   4='href="http://www.sas.com/rnd/app/qc/qcnew.html"'
   5='href="http://www.sas.com/rnd/app/qc/qc.html"'
   ;

data wafers;
   format Web webfmt.;
   input Batch Web @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
 1  1  35.00 34.99 34.99 34.98 35.00
 2  1  35.00 34.99 34.99 34.98 35.00
 3  1  34.99 34.99 35.00 34.99 35.00
 4  1  35.00 35.00 34.99 34.99 35.00
 5  2  35.00 34.99 34.98 34.99 35.00
 6  2  34.99 34.99 35.00 35.00 35.00
 7  2  35.01 34.98 35.00 35.00 34.99
 8  2  35.00 35.00 34.99 34.98 34.99
 9  3  34.99 34.98 34.99 35.01 35.00
10  3  34.99 35.00 35.00 34.99 35.00
11  3  35.01 35.00 35.00 34.98 34.99
12  3  34.99 34.99 35.00 34.98 35.01
13  4  35.01 34.99 34.98 34.99 34.99
14  4  35.00 35.00 34.99 35.00 34.99
15  4  34.98 35.00 34.99 35.00 34.99
16  4  34.99 35.00 35.00 35.01 35.00
17  5  34.98 34.98 34.98 34.99 34.98
18  5  35.01 35.02 35.00 34.98 35.00
19  5  34.99 34.98 35.00 34.99 34.98
20  5  34.99 35.00 35.00 34.99 34.99
;
```

```
symbol1 v=square;
proc shewhart data=wafers;
   xchart Diameter*Batch / html = ( Web );
run;

ods html close;
run;
```

In this example five different URLs are each associated with a set of four subgroup values. When you view the ODS HTML output with a browser, you can click on a subgroup data point and the browser will bring up the page specified by the subgroup's URL. These URLs happen to point to pages at SAS Institute's web site which might be of interest to SAS/QC users.

NOTE: The value of the HTML= variable must be the same for each observation belonging to a given subgroup.

Links and Tests for Special Causes

The TESTHTML= data set provides a way to associate a link with each subgroup in a control chart for which a given test for special causes is positive:

Table 19.103 Variables Required in a TESTHTML= Data Set

Variable	Type	Description
TEST	Character or numeric	Test identifier
CHART	Numeric	Primary (1) or secondary (2) chart
URL	Character	HTML specifying URL for subgroups with positive test

The variable _TEST_ identifies a test for special causes (see "Tests for Special Causes: SHEWHART Procedure" on page 2119). A standard test is identified by its number (1 to 8) and a nonstandard test is identified by the CODE= character in its pattern specification. The _TEST_ variable must be a character variable if nonstandard tests are included in the TESTHTML= data set. The value of _CHART_ is 1 or 2, specifying whether the test applies to the primary or secondary chart. The character variable _URL_ contains the HTML syntax for the link to be associated with subgroups for which the test is positive.

The following statements create a TESTHTML= data set and an $\bar{X}$ chart using the same DATA= data set as the previous example:

```
ods html body = "example2.html";

data testlink;
   length _URL_ $ 75;
   input _TEST_ _CHART_ _URL_;
   datalines;
1  1  href="http://www.sas.com/"
2  1  href="http://www.sas.com/service/techsup/faq/qc/shewproc.html"
3  1  href="http://www.sas.com/rnd/app/qc.html"
4  1  href="http://www.sas.com/rnd/app/qc/qcnew.html"
5  1  href="http://www.sas.com/products/qc/index.html"
6  1  href="http://www.sas.com/rnd/app/qc/qc/qcspc.html"
```

```
7  1  href="http://www.sas.com/software/components/qc.html"
8  1  href="http://www.sas.com/rnd/app/qc/qc.html"
;

symbol1 v=dot;
proc shewhart data=wafers testhtml=testlink;
   xchart Diameter*Batch / tests = 1 to 8;
run;

ods html close;
run;
```

In this example only subgroups triggering tests for special causes have URLs associated with them.

NOTE: If a TESTHTML= data set and an HTML= variable are both specified, the link from the TESTHTML= data set is associated with any subgroup for which the test is positive.

References

Al-Salti, M., and Statham, A. (1994). "A Review of the Literature on the Use of SPC in Batch Production." *Quality and Reliability Engineering International* 10:49–62.

Alt, F. (1985). "Multivariate Quality Control." In *Encyclopedia of Statistical Sciences,* vol. 6, edited by S. Kotz, N. L. Johnson, and C. B. Read. New York: John Wiley & Sons.

Alwan, L. C., and Roberts, H. V. (1988). "Time Series Modeling for Statistical Process Control." *Journal of Business and Economic Statistics* 6:87–95.

American Society for Testing and Materials (1976). *ASTM Manual on Presentation of Data and Control Chart Analysis*. Philadelphia: ASTM.

ASQC Automotive Division/AIAG (1990). *Fundamental Statistical Process Control: Reference Manual*. Southfield, MI: Automotive Industry Action Group.

Austin, J. A. (1973). "Control Chart Constants for Largest and Smallest in Sampling from a Normal Distribution Using the Generalized Burr Distribution." *Technometrics* 15:931–933.

Bissell, A. F. (1990). "How Reliable Is Your Capability Index?" *Journal of the Royal Statistical Society, Series C* 39:331–340.

Box, G. E. P., and Kramer, T. (1992). "Statistical Process Monitoring and Feedback Adjustment: A Discussion." *Technometrics* 34:251–285. With discussion.

Boyles, R. A. (1997). "Estimating Common-Cause Sigma in the Presence of Special Causes." *Journal of Quality Technology* 29:381–395.

Burr, I. W. (1969). "Control Charts for Measurements with Varying Sample Sizes." *Journal of Quality Technology* 1:163–167.

Burr, I. W. (1976). *Statistical Quality Control Methods*. New York: Marcel Dekker.

Champ, S. W., and Woodall, W. H. (1987). "Exact Results for Shewhart Control Charts with Supplementary Runs Rules." *Technometrics* 29:393–401.

Champ, S. W., and Woodall, W. H. (1990). "A Program to Evaluate the Run Length Distribution of a Shewhart Control Chart with Supplementary Run Rules." *Journal of Quality Technology* 29:393–399.

Deming, W. E. (1982). *Out of the Crisis*. Cambridge, MA: Center for Advanced Engineering Study, Massachusetts Institute of Technology.

Doganaksoy, N., Faltin, F. W., and Tucker, W. T. (1991). "Identification of Out-of-Control Quality Characteristics in a Multivariate Manufacturing Environment." *Communications in Statistics—Theory and Methods* 20:2775–2790.

Draper, N. R., and Smith, H. (1981). *Applied Regression Analysis*. 2nd ed. New York: John Wiley & Sons.

Gnanadesikan, R., and Kettenring, J. R. (1972). "Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data." *Biometrics* 28:81–124.

Grant, E. L., and Leavenworth, R. S. (1988). *Statistical Quality Control*. 6th ed. New York: McGraw-Hill.

Hawkins, D. M. (1991). "Multivariate Quality Control Based on Regression-Adjusted Variables." *Technometrics* 33:61–75.

Hawkins, D. M. (1993). "Regression Adjustment for Variables in Multivariate Quality Control." *Journal of Quality Technology* 25:170–182.

Hillier, F. S. (1969). "$\bar{X}$- and *R*-Chart Control Limits Based on a Small Number of Subgroups." *Journal of Quality Technology* 1:17–26.

Hunter, J. S. (1986). "The Exponentially Weighted Moving Average." *Journal of Quality Technology* 18:203–210.

Hunter, J. S. (1988). "The Digidot Plot." *American Statistician* 42:54.

Iglewicz, B., and Hoaglin, D. C. (1987). "Use of Boxplots for Process Evaluation." *Journal of Quality Technology* 19:180–190.

Jackson, J. E. (1980). "Principal Components and Factor Analysis, Part 1: Principal Components." *Journal of Quality Technology* 12:201–213.

Jackson, J. E. (1991). *A User's Guide to Principal Components*. New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Kemp, A. W. (1992). *Univariate Discrete Distributions*. 2nd ed. New York: John Wiley & Sons.

Kendall, M. G. (1955). *Rank Correlation Methods*. 2nd ed. London: Charles Griffin.

Kume, H. (1985). *Statistical Methods for Quality Improvement*. Tokyo: AOTS Chosakai.

MacGregor, J. F. (1987). "Interfaces between Process Control and Online Statistical Process Control." *Computing and Systems Technology Division Communications* 10:9–20.

MacGregor, J. F. (1990). "A Different View of the Funnel Experiment." *Journal of Quality Technology* 22:255–259.

MacGregor, J. F., Hunter, J. S., and Harris, T. (1988). "SPC Interfaces." Short course notes.

Mason, R. L., Tracy, N. D., and Young, J. C. (1993). "Use of Hotelling's T^2 Statistic in Multivariate Control Charts." Unpublished paper.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978). "Variations of Box Plots." *American Statistician* 32:12–16.

Montgomery, D. C. (1996). *Introduction to Statistical Quality Control*. 3rd ed. New York: John Wiley & Sons.

Montgomery, D. C., Keats, J. B., Runger, G. C., and Messina, W. S. (1994). "Integrating Statistical Process Control and Engineering Process Control." *Journal of Quality Technology* 26:79–87.

Montgomery, D. C., and Mastrangelo, C. M. (1991). "Some Statistical Process Control Methods for Autocorrelated Data." *Journal of Quality Technology* 23:179–204. With discussion.

Nelson, L. S. (1982). "Control Charts for Individual Measurements." *Journal of Quality Technology* 14:172–174.

Nelson, L. S. (1984). "The Shewhart Control Chart—Tests for Special Causes." *Journal of Quality Technology* 15:237–239.

Nelson, L. S. (1985). "Interpreting Shewhart $\bar{X}$ Control Charts." *Journal of Quality Technology* 17:114–116.

Nelson, L. S. (1989). "Standardization of Shewhart Control Charts." *Journal of Quality Technology* 21:287–289.

Nelson, L. S. (1994). "Shewhart Control Charts with Unequal Subgroup Sizes." *Journal of Quality Technology* 26:64–67.

Quesenberry, C. P. (1991a). "SPC Q Charts for a Binomial Parameter p: Short or Long Runs." *Journal of Quality Technology* 23:239–246.

Quesenberry, C. P. (1991b). "SPC Q Charts for Start-Up Processes and Short or Long Runs." *Journal of Quality Technology* 23:213–224.

Quesenberry, C. P. (1993). "The Effect of Sample Size on Estimated Effects." *Journal of Quality Technology* 25:237–247.

Rocke, D. M. (1989). "Robust Control Charts." *Technometrics* 31:173–184.

Rodriguez, R. N., and Bynum, R. A. (1992). "Examples of Short Run Process Control Methods with the SHEWHART Procedure in SAS/QC Software." Unpublished manuscript available from the authors.

Ryan, T. P. (1989). *Statistical Methods for Quality Improvement*. New York: John Wiley & Sons.

Schilling, E. G., and Nelson, P. R. (1976). "The Effect of Non-normality on the Control Limits of $\bar{X}$ Charts." *Journal of Quality Technology* 8:183–187.

Schneider, H., and Pruett, J. M. (1994). "Control Charting Issues in the Process Industries." *Quality Engineering* 6:347–373.

Shewhart, W. A. (1931). *Economic Control of Quality Manufactured Product*. New York: D. Van Nostrand; republished in 1980 by the American Society for Quality Control.

Snedecor, G. W., and Cochran, W. G. (1980). *Statistical Methods*. 7th ed. Ames: Iowa State University Press.

Teichroew, D. (1962). "Tables of Expected Values of Order Statistics and Products of Order Statistics for Samples of Size 20 and Less from the Normal Distribution." In *Contributions to Order Statistics*, edited by A. E. Sarhan, and B. G. Greenberg, 190–205. New York: John Wiley & Sons.

Tracy, N. D., Young, J. C., and Mason, R. L. (1992). "Multivariate Control Charts for Individual Observations." *Journal of Quality Technology* 24:88–95.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Western Electric Company (1956). *Statistical Quality Control Handbook*. Indianapolis: Western Electric Company.

Westgard, J. O. (2002). *Basic QC Practices: Training in Statistical Quality Control for Healthcare Laboratories*. Madison, WI: Westgard QC.

Wetherill, G. B., and Brown, D. B. (1991). *Statistical Process Control: Theory and Practice*. London: Chapman & Hall.

Wheeler, D. J. (1991). "Shewhart's Chart: Myths, Facts, and Competitors." *Annual Quality Congress Transactions* 45:533–538.

Wheeler, D. J. (1995). *Advanced Topics in Statistical Process Control*. Knoxville, TN: SPC Press.

Wheeler, D. J., and Chambers, D. S. (1986). *Understanding Statistical Process Control*. Knoxville, TN: SPC Press.

Wilks, S. S. (1962). *Mathematical Statistics*. New York: John Wiley & Sons.

Woodall, W. H. (1993). "Autocorrelated Data and SPC." *ASQC Statistics Division Newsletter* 13:18–21.

Appendix A
Measurement Systems Analysis

Contents

Overview **2193**
Terminology 2193
Syntax **2194**
%basicemp Macro 2194
%hongrr Macro 2195
%msagrr Macro 2196
%shortemp Macro 2197
Examples **2198**
Example 1.1: A Short EMP Study 2198
Example 1.2: A Basic EMP Study 2200
Example 1.3: Gauge R&R for Gasket Thickness 2205
Example 1.4: Honest Gauge R&R for Gasket Thickness 2207
References **2208**

Overview

Measurement systems are essential to the quality of a manufacturing process. The instruments that take measurements are subject to variation. Therefore, the variation in measured quantities consists of the variation in the product that is being measured plus the variation in the measurement system. Too much variation in the measurement system can mask variation in the manufacturing process.

The SAS autocall macro library provides two macros for gauge repeatability and reproducibility (R&R) and two macros for evaluating the measurement process (EMP). Both of these methods examine the precision (reproducibility), consistency (repeatability) and bias of a measurement system. This appendix describes the syntax for the macros and provides simple examples that show how they are used. See Wheeler (2006) for a thorough discussion of evaluating the measurement process.

Terminology

The following definitions describe terms used in measurement systems analysis.

Condition — typically an operator, but can be thought of more generically as any condition that could affect the measurements. For example, with an automated process, condition might be a set-up procedure or an environmental condition such as temperature. A condition represents a potential nuisance source of variation.

Gauge	any device used to obtain measurements, for example, a micrometer or a gasket thickness gauge.
Measurement System	the complete process used to obtain measurements. This includes people, gauges, operations, and procedures.
Part	the item that is measured, for example, a gasket. The parts selected should represent the entire operating range (variability) of the process.
Repeatability	the variation resulting from repeated measurements taken on the same part with the same gauge by the same operator. Repeatability is the gauge or equipment variation. This is also called *test-retest variation*.
Reproducibility	the variation in the average of the measurements resulting when different operators using the same gauge take measurements on the same part. Reproducibility is the operator-to-operator variability.
Trial	a set of measurements on all parts taken by one operator. Multiple trials help separate the gauge variability (repeatability) from the variability contributed by operators (reproducibility).

Syntax

Two macros are provided for gauge repeatability and reproducibility:

- %hongrr
- %msagrr

Two macros are provided for evaluating the measurement process:

- %basicemp
- %shortemp

%basicemp Macro

%basicemp *(parameters)* **;**

You can use the %basicemp macro to perform a Basic EMP Study, which determines whether a condition has a detrimental effect on the measurement process. Common conditions include different operators and different measurement instruments.

The %basicemp macro produces the following outputs:

- average and range charts
- a main effect chart

- a mean range chart
- a table of summary statistics for the study
- a table of statistics that characterize the measurement system and the relative utility for each level of the condition

The parameters for this macro are as follows:

SAS-data-set
is the name of the data set that contains the measurement data. You must specify a value for this parameter.

NR=*n*
specifies the number of measurements that were taken to make one reported value. By default, $n = 1$.

SAMPLE=*variable*
specifies the variable in the input data set that identifies the subgroups of measurements. By default, *variable* is Sample.

CONDITION=*variable*
specifies the variable in the input data set that identifies a condition that potentially affects measurement variation. By default, *variable* is Condition.

VALUE=*variable*
specifies the variable in the input data set that contains the reported values. By default, *variable* is Value.

DISCRETE=YES | NO
determines whether values of the sample variable are treated as discrete values in the mean and range chart. This option applies when the sample variable is numeric. If you specify DISCRETE=YES, the sample values will be displayed at regular intervals on the horizontal axis, even if the intervals between the sample values are not equal. By default, DISCRETE=YES;

%hongrr Macro

%hongrr *(parameters)* **;**

The %hongrr macro produces a summary report for an Honest Gauge R&R Study, as described by Wheeler (2006). The parameters for this macro are as follows:

SAS-data-set
is the name of the data set that contains the measurement data. You must specify a value for this parameter.

SAMPLE=*variable*
specifies the variable in the input data set that identifies the subgroups of measurements. By default, *variable* is Sample.

CONDITION=*variable*
specifies the variable in the input data set that identifies a condition that potentially affects measurement variation. By default, *variable* is Condition.

VALUE=*variable*
specifies the variable in the input data set that contains the reported values. By default, *variable* is Value.

%msagrr Macro

%msagrr *(parameters)* **;**

The %msagrr macro performs a gauge R&R analysis based on the methods described in ASQC Automotive Division/AIAG (2010). It produces average and range charts and a table of estimates of the following sources of variation:

EV	equipment variation, also called test-retest variation. This is a measure of repeatability.
AV	appraiser variation. This is variation due to differences among operators and is a measure of reproducibility.
IV	interaction between operators and parts.
PV	part variation. This is the variation in the manufacturing process that the measurement system is intended to measure.

The table contains estimates that are produced by the average and range method and by the variance components method.

Wheeler (2006) describes problems with the AIAG Gauge R&R Study, and recommends better approaches that are supported by the %basicemp, %hongrr, and %shortemp macros.

The parameters for this macro are as follows:

SAS-data-set
is the name of the data set that contains the measurement data. The default is the most recently created SAS data set.

NR=*n*
specifies the number of measurements that were taken to make one reported value. By default, $n = 1$.

SAMPLE=*variable*
specifies the variable in the input data set that identifies the subgroups of measurements. By default, *variable* is Sample.

CONDITION=*variable*
specifies the variable in the input data set that identifies a condition that potentially affects measurement variation. By default, *variable* is Condition.

VALUE=*variable*
specifies the variable in the input data set that contains the reported values. By default, *variable* is Value.

NU=*k*
specifies the multiple of σ to be used to compute the limits on the average and range charts. By default, $k = 5.15$. Other commonly-used values are 4 and 6.

CHARTS=YES | NO
determines whether mean and range charts are produced. By default, CHARTS=YES.

VARCOMP=YES | NO
determines whether variance components are estimated by using the VARCOMP procedure. By default, VARCOMP=YES

DISCRETE=YES | NO
determines whether mean and range charts are produced. By default, DISCRETE=YES

%shortemp Macro

%shortemp *(parameters)* **;**

You an use the %shortemp macro to perform a Short EMP Study, as described by Wheeler (2006). A Short EMP Study characterizes the relative utility of a particular measurement system for use with a particular product. The macro produces average and range charts and a table of statistics that summarize the study.

The parameters for this macro are as follows:

SAS-data-set
is the name of the data set that contains the measurement data. You must specify a value for this parameter.

NR=*n*
specifies the number of measurements that were taken to make one reported value. By default, $n = 1$.

SAMPLE=*variable*
specifies the variable in the input data set that identifies the subgroups of measurements. By default, *variable* is Sample.

VALUE=*variable*
specifies the variable in the input data set that contains the reported values. By default, *variable* is Value.

DISCRETE=YES | NO
determines whether values of the sample variable are treated as discrete values in the mean and range chart. This option applies when the sample variable is numeric. If you specify DISCRETE=YES, the sample values will be displayed at regular intervals on the horizontal axis, even if the intervals between the sample values are not equal. By default, DISCRETE=YES;

Examples

Example 1.1: A Short EMP Study

The purpose of a Short EMP Study is to determine the suitability of a measurement system for measuring a particular product. The study is designed to eliminate sources of measurement variation other than pure test-retest variation (for example, operator effects). In this example from Wheeler (2006), ten samples of Product 833 were selected for measurement. All ten samples was measured in each of three trials. The same operator used the same measurement instrument (Gauge 702) to perform each measurement. The following statements create a SAS data set named Gauge702Product833, which contains the measurements:

```
data Gauge702Product833;
   input Trial Sample Value;
   datalines;
1  1  3.66
1  2  4.50
1  3  3.63
1  4  4.28
1  5  5.66
1  6  3.36
1  7  4.20
1  8  6.95
1  9  3.41
1 10  2.43
2  1  4.26
2  2  3.85
2  3  3.03
2  4  5.08
2  5  4.81
2  6  3.91
2  7  4.35
2  8  5.60
2  9  2.81
2 10  2.98
3  1  5.41
3  2  3.35
3  3  3.53
3  4  4.63
3  5  5.31
3  6  4.06
3  7  5.05
3  8  5.70
3  9  3.21
3 10  2.68
;
```

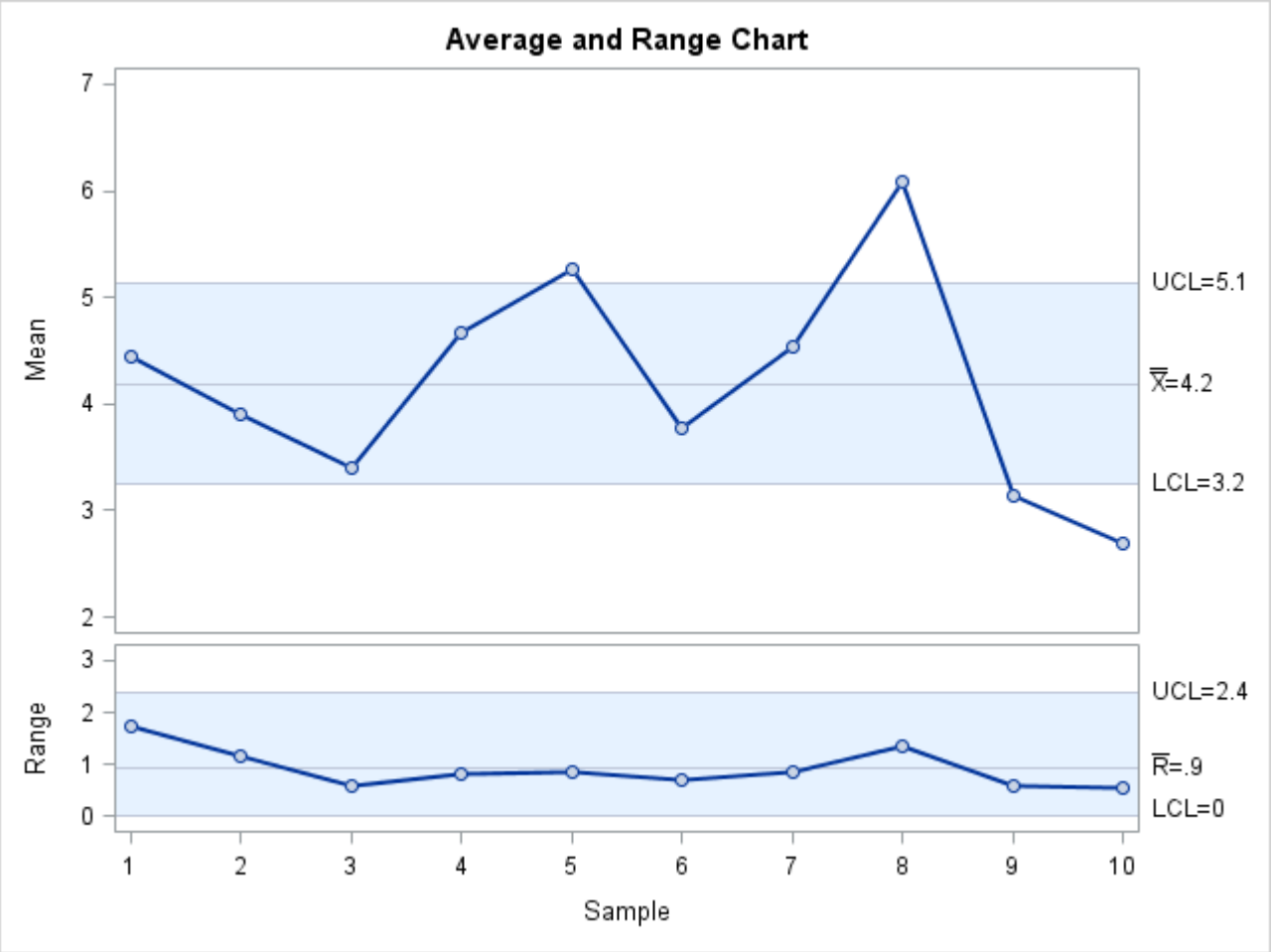
You can use the %shortemp macro to perform the study:

```
%shortemp(Gauge702Product833);
```

The default values for all the macro parameters are appropriate in this case, so you only need to specify the data set that contains the measurements.

Output 1.1.1 shows the average and range ($\bar{X}$ and R) chart that is produced by the macro.

Output 1.1.1 Short EMP Study Results



Each subgroup in the average and range charts consists of the three measurements for a particular sample. Because the same part is measured three times, the subgroup ranges indicate the test-retest variation. With no ranges outside the control limits, there is no evidence of inconsistency in the measurements.

The ranges are also used to construct the limits on the average chart. Therefore the limits characterize measurement errors. Subgroup means outside or near the control limits indicate that the measurement error is not large enough to mask the process error. This measurement system should be adequate to monitor unusual variation in the process.

Output 1.1.2 shows the summary table produced by the macro.

Output 1.1.2 Short EMP Study Results

Short EMP Study for Gauge702Product833

Short EMP Analysis	
Upper-Range Limit Check	Not Exceeded
Number of Samples	10
Number of Replications	3
Probable Error for Single Determination	0.36690
Measurement Increment	0.01
Measurement Increment Action	Might Drop a Digit
Test-Retest Error for Reported Value	0.29545
Estimated Variance of Product	0.93992
Intraclass Correlation	0.7608
System Classification	Second Class

Note the probable error and measurement increment values. According to Wheeler (2006), the smallest effective measurement increment is 0.2 times the probable error, and the largest effective measurement increment is 2 times the probable error. In this case the measurement increment is only about 1/36 of the probable error. The second decimal place in the measurements is suspect, and the report recommends that the last digit could be dropped.

Example 1.2: A Basic EMP Study

This example is taken from Wheeler (2006).

Three different operators each conducted two trails in which they measured a characteristic of six wafers. The following statements create a SAS data set named WaferMeasurements, which contains the measurements they made:

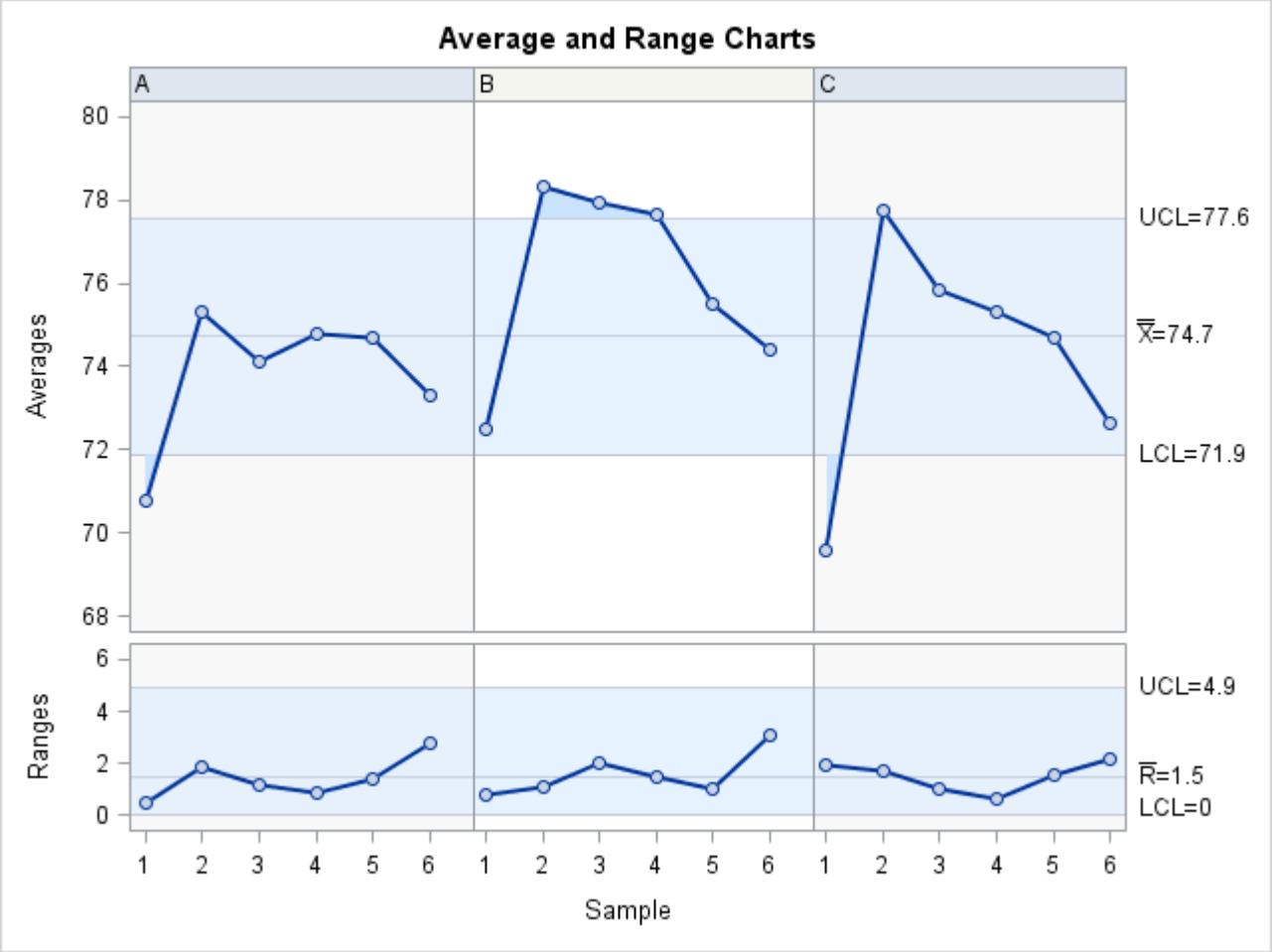
```
data WaferMeasurements;
   input Operator $ Trial Sample Value;
   datalines;
A 1 1 70.52
A 1 2 74.40
A 1 3 73.54
A 1 4 75.20
A 1 5 73.99
A 1 6 71.89
A 2 1 71.03
A 2 2 76.24
A 2 3 74.68
A 2 4 74.33
A 2 5 75.39
A 2 6 74.70
B 1 1 72.08
B 1 2 77.78
B 1 3 76.93
B 1 4 78.40
B 1 5 76.04
B 1 6 75.98
B 2 1 72.89
B 2 2 78.90
B 2 3 78.93
B 2 4 76.90
B 2 5 75.01
B 2 6 72.87
C 1 1 70.56
C 1 2 76.88
C 1 3 75.34
C 1 4 75.65
C 1 5 73.91
C 1 6 73.73
C 2 1 68.61
C 2 2 78.61
C 2 3 76.35
C 2 4 75.02
C 2 5 75.46
C 2 6 71.55
;
```

You can use the %basicemp macro to perform the study:

```
%basicemp(WaferMeasurements, condition=Operator);
```

The average and range chart is shown is Output 1.2.1.

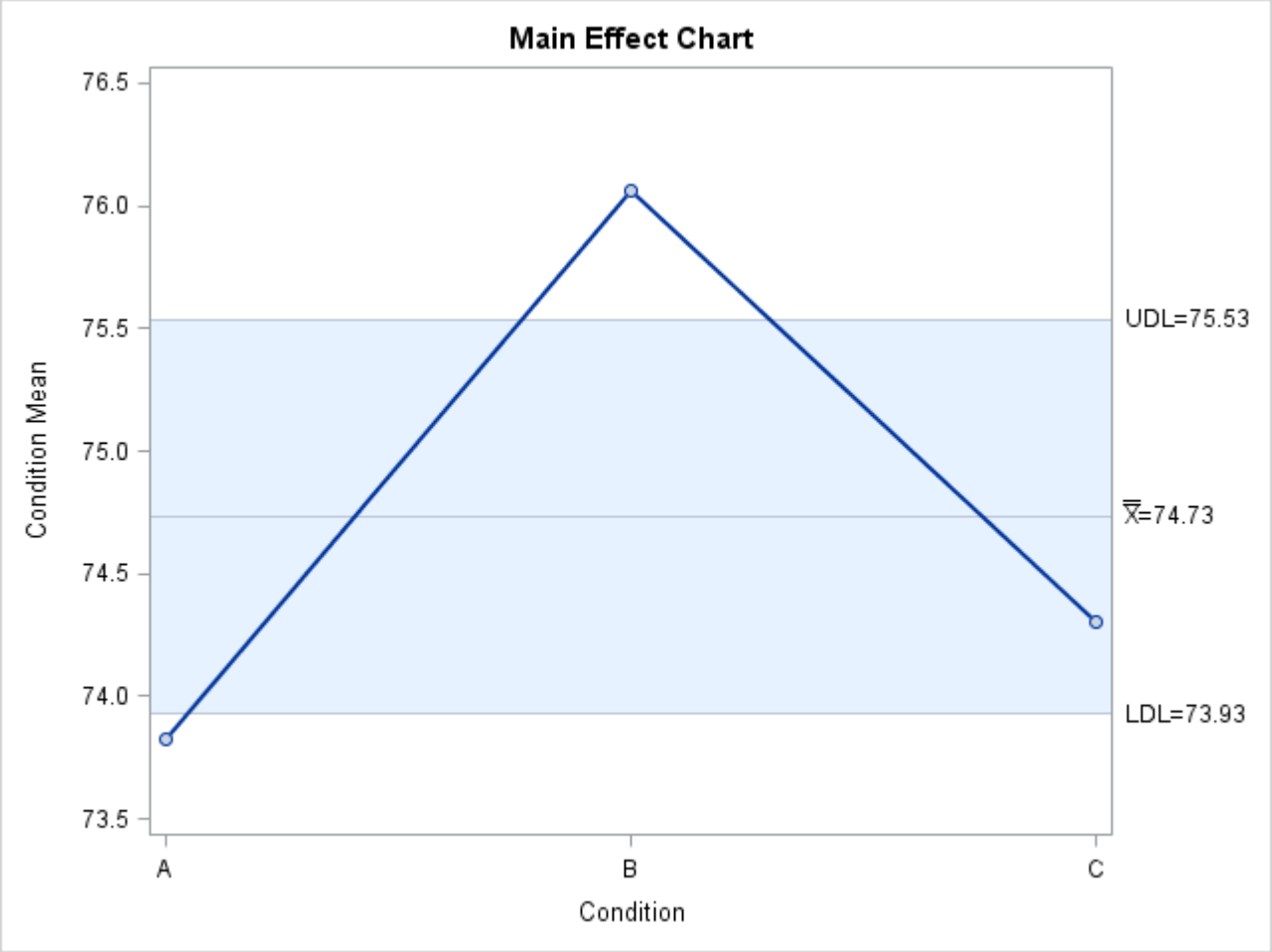
Output 1.2.1 Basic EMP Study Results



All the ranges are within the limits on the range chart, which means there is no indication of measurement inconsistency. However, the averages for the different operators do not appear to be consistent.

The main effect chart is shown in Output 1.2.2.

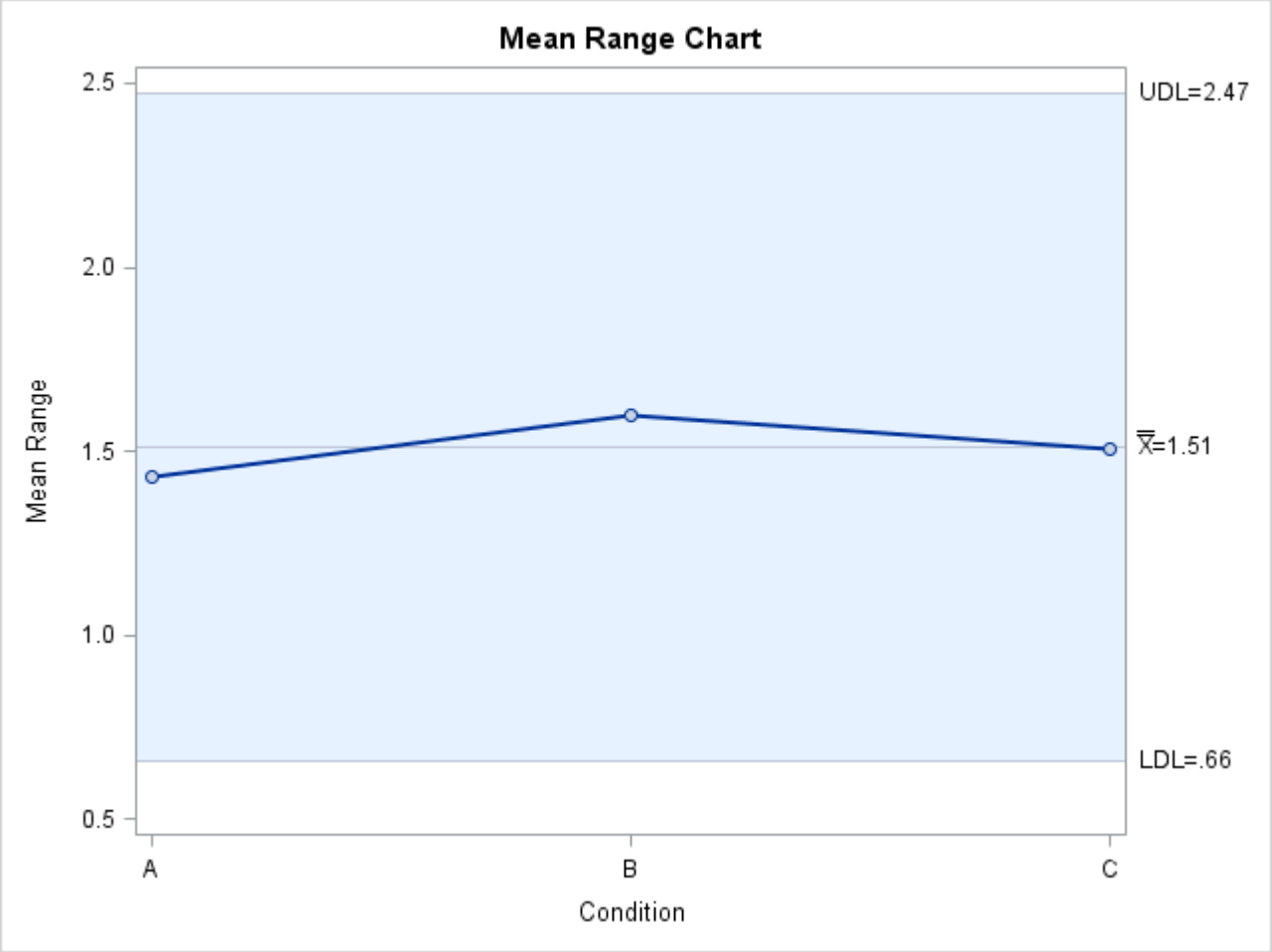
Output 1.2.2 Basic EMP Study Results



Because the means for operators A and B are both outside the limits, the mean range chart indicates operator biases in the measurements.

The mean range chart is shown in Output 1.2.3.

Output 1.2.3 Basic EMP Study Results



The fact that the mean range for each operator is well within the limits reinforces the conclusion that the measurement process is consistent.

The Basic EMP reports are shown in Output 1.2.4.

Output 1.2.4 Basic EMP Study Results

Basic EMP Study for WaferMeasurements

EMP Study for One Nuisance Component	
Number of Samples	6
Number of Nuisance Component Levels	3
Number of Times Samples Measured	2
Number of Measurements Averaged in Reported Values	1
Estimated Test-Retest Error	1.33870
Probable Error	0.90362
Intraclass Correlation (Without Bias)	0.7214
System Classification	Second Class
Intraclass Correlation (With Bias)	0.6054

Output 1.2.4 *continued*

Basic EMP Study for WaferMeasurements

Analysis of Nuisance Component		
Level of Nuisance Component	Estimated PE Variance	Intraclass Correlation
A	1.60232	0.7433
B	1.99807	0.6990
C	1.78684	0.7220

Note that the values listed for the intraclass correlations with and without bias are different from those given by Wheeler. That is because the %basicemp macro estimates the product and operator variations by using standard deviations instead of ranges.

Example 1.3: Gauge R&R for Gasket Thickness

This example is patterned after an example given in ASQC Automotive Division/AIAG (1990).

Suppose the ABC Company needs to evaluate a gasket thickness gauge. Three operators (George, Jane, and Robert) are selected for this study. Using the same gauge, each operator measures ten parts (gaskets) in a random order. Each part is measured by each operator twice (two trials). The following statements create a data set called ABC that contains the measurements (gasket thicknesses) collected by each operator.

```
data ABC;
   input Operator $ Sample @;
   do i=1 to 2;
      input Trial @;
      output;
   end;
   datalines;
George    1   0.65  0.60
George    2   1.00  1.00
George    3   0.85  0.80
George    4   0.85  0.95
George    5   0.55  0.45
George    6   1.00  1.00
George    7   0.95  0.95
George    8   0.85  0.80
George    9   1.00  1.00
George   10   0.60  0.70
Jane      1   0.55  0.55
Jane      2   1.05  0.95
Jane      3   0.80  0.75
Jane      4   0.80  0.75
Jane      5   0.40  0.40
Jane      6   1.00  1.05
Jane      7   0.95  0.90
Jane      8   0.75  0.70
Jane      9   1.00  0.95
```

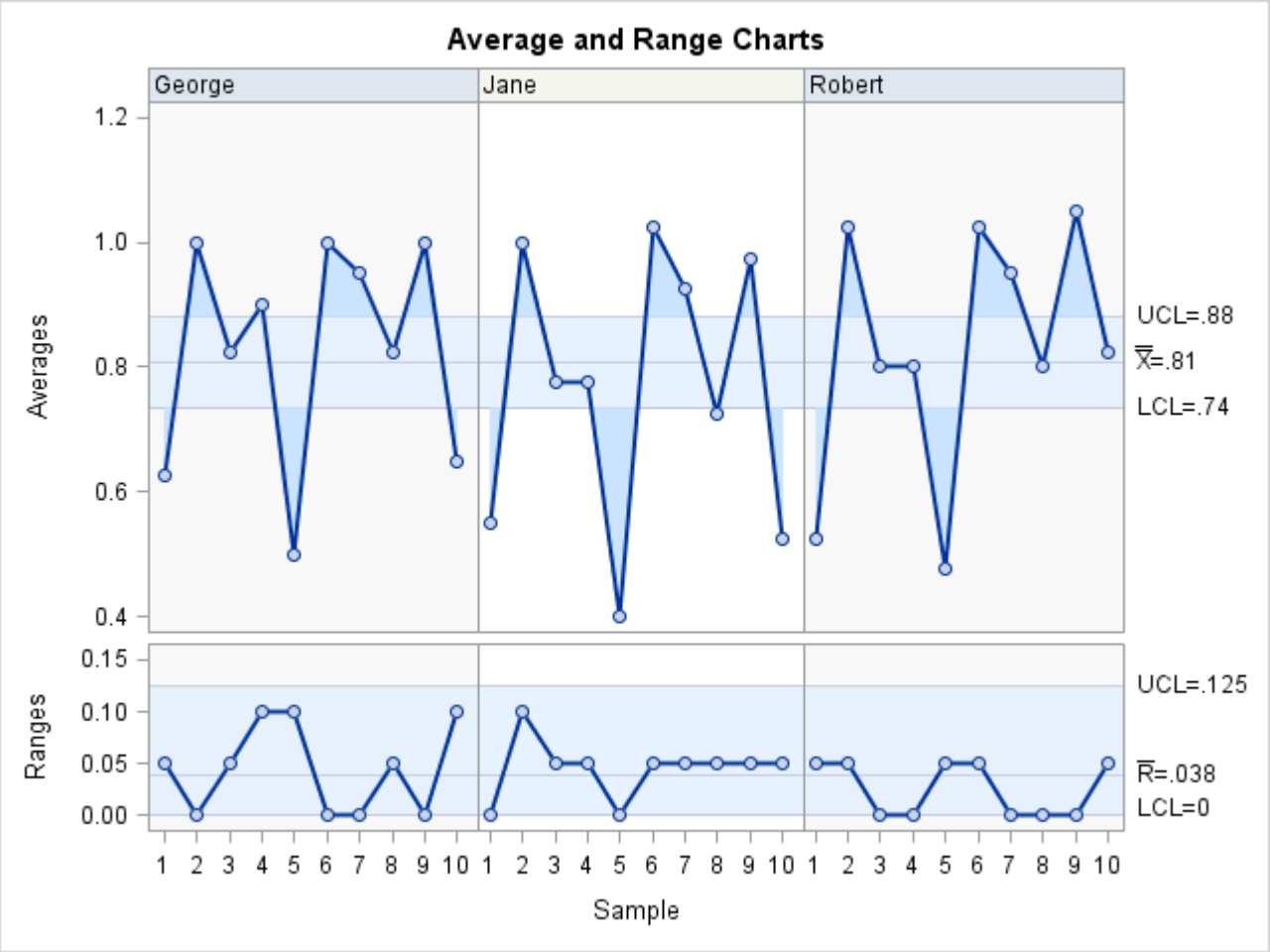
```
Jane      10    0.55   0.50
Robert     1    0.50   0.55
Robert     2    1.05   1.00
Robert     3    0.80   0.80
Robert     4    0.80   0.80
Robert     5    0.45   0.50
Robert     6    1.00   1.05
Robert     7    0.95   0.95
Robert     8    0.80   0.80
Robert     9    1.05   1.05
Robert    10    0.85   0.80
;
```

You can use the %msagrr macro to perform the analysis:

```
%msagrr(ABC, condition=Operator, value=Trial);
```

The average and range chart is shown is Output 1.3.1.

Output 1.3.1 Gasket Thickness Average and Range Charts



No points on the range chart are outside the limits, and the variability across operators is fairly comparable. This indicates that all operators are using the gauge in the same way. If there are any points outside the limits, they should be investigated and dealt with before you proceed.

The Gauge R&R report is shown in Output 1.3.2.

Output 1.3.2 Gauge R&R Results

Gauge R&R Study for ABC

Gauge RR Variance Estimate Table

Estimation Method	EV Estimated Variance	AV Estimated Variance	IV Estimated Variance	PV Estimated Variance
Average and Range	0.17471	0.15699	.	0.90422
Variance Component	0.18509	0.15553	0.24340	0.99282

Example 1.4: Honest Gauge R&R for Gasket Thickness

You can use the %hongrr macro to perform an Honest Gauge R&R Study on the data from the ABC Company from Example 1.3:

```
%hongrr(ABC, condition=Operator, value=Trial);
```

The results are shown in Output 1.4.1.

Output 1.4.1 Honest Gauge R&R Results

Honest Gauge R&R Study for ABC

Honest Gauge RR Analysis Table	
Upper-Range Limit Check	Not Exceeded
Number of Samples	10
Number of Nuisance Component Levels	3
Number of Replications	2
Estimated PE Variance	0.001154
Estimated O Variance	0.001142
Estimated E Variance	0.002296
Estimated P Variance	0.031838
Estimated X Variance	0.034134
Repeatability Proportion of Variation	0.033810
Reproducibility Proportion of Variation	0.033465
Combined RR Proportion of Variation	0.067275
Probable Error	0.022931
Intraclass Correlation	0.9327
System Classification	First Class
Measurement Increment	0.01
Measurement Increment Action	Correct number of digits

See Wheeler (2006) for a description of the Honest Gauge R&R Study and instructions for interpreting its results.

References

ASQC Automotive Division/AIAG (1990). *Measurement Systems Analysis Reference Manual*. Southfield, MI: Automotive Industry Action Group.

ASQC Automotive Division/AIAG (2010). *Measurement Systems Analysis Reference Manual*. 4th ed. Troy, MI: Automotive Industry Action Group.

Burdick, R. K., Borror, C. M., and Montgomery, D. C. (2005). *Design and Analysis of Gauge R&R Studies: Making Decisions with Confidence Intervals in Random and Mixed ANOVA Models*. Philadelphia, PA, and Alexandria, VA: SIAM and ASA.

Wheeler, D. J. (2006). *EMP III: Evaluating the Measurement Process and Using Imperfect Data*. Knoxville, TN: SPC Press.

Appendix B
The RELIABILITY Graphical Interface

An experimental graphical interface for the RELIABILITY procedure is implemented using FRAME entries in SAS/AF software. The application is available as a SAS/QC sample library program and is stored in the reliab catalog. (File extensions for SAS catalogs differ based on the operating system.)

Assume that you are using the SAS System under Microsoft Windows and that the SAS/QC sample library is stored in the `c:\sas\qc\sample` directory. (Check with your SAS site representative for the location of the SAS/QC sample library on your system.) You invoke the RELIABILITY application as follows:

1. First, for SAS 9.3 and later releases, you must enter the following statements to ensure proper operation of the RELIABILITY application:

```
ods graphics off;
ods html close;
ods listing;
```

2. Next you must tell the SAS System where the catalog is stored:

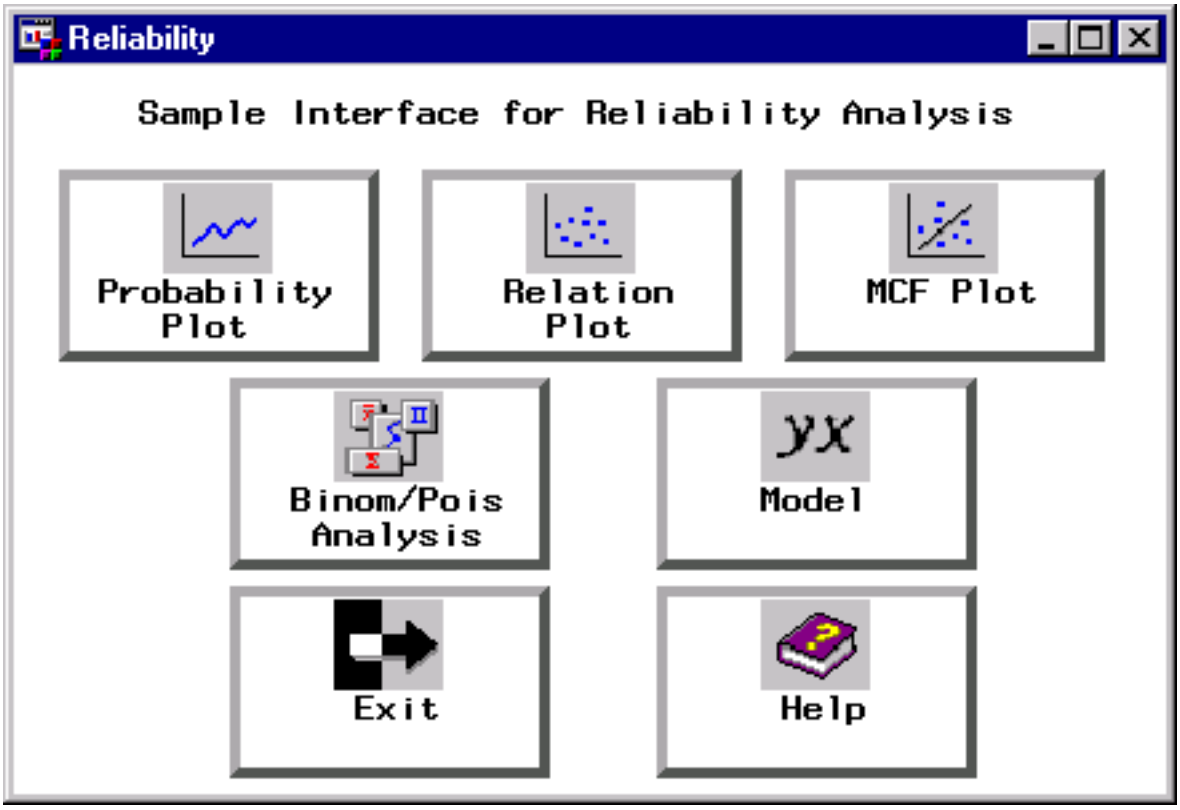
```
libname rel 'c:\sas\qc\sample';
```

3. You then issue the following command from any SAS display manager window:

```
af c=rel.reliab.reliab.frame aws=no
```

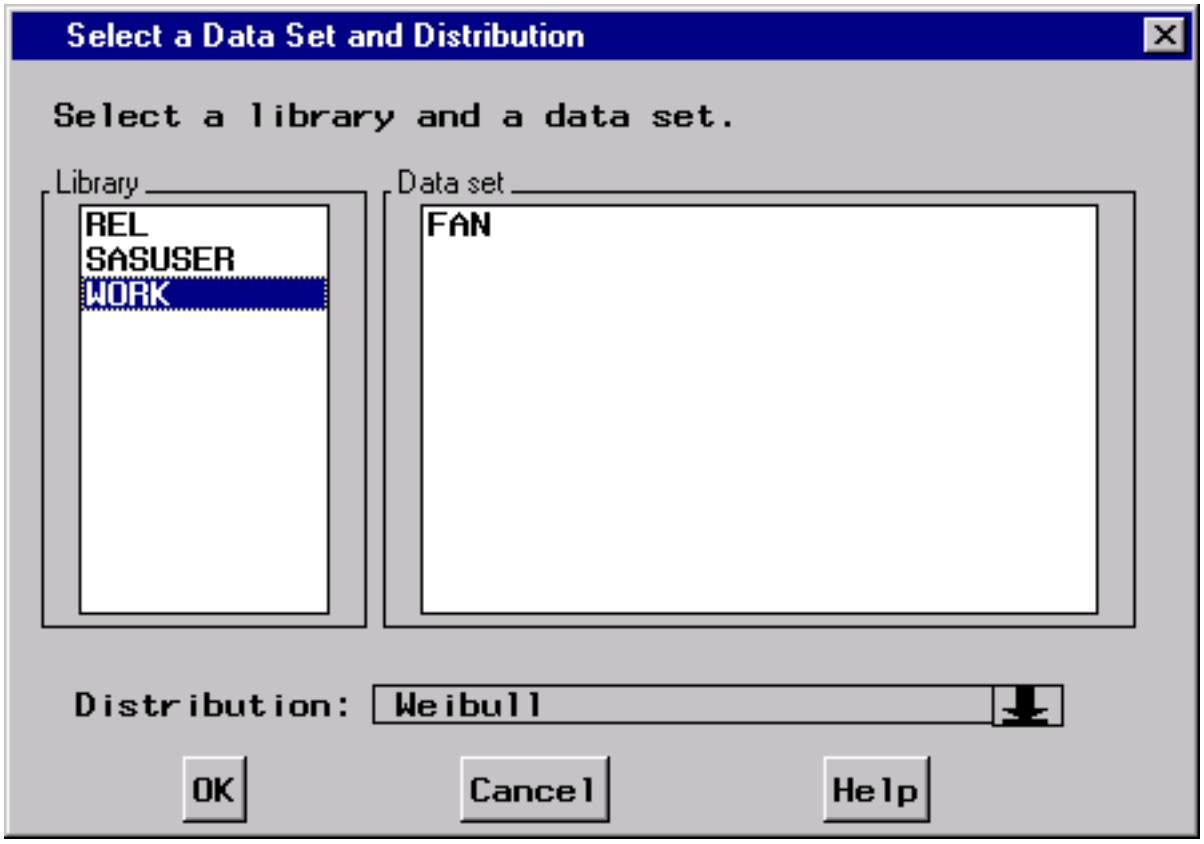
The main application window appears, as shown in Figure B.1. You select a type of analysis from the main window. For example, you can select a probability plot by clicking the Probability Plot button.

Figure B.1 Main Window



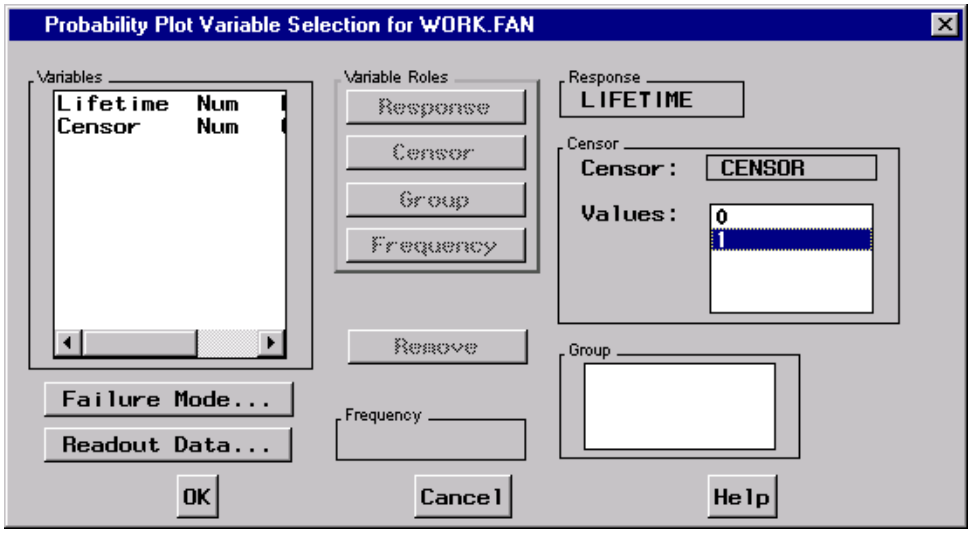
The next window to appear enables you to select the SAS data set that contains your data. You also specify a probability distribution for the probability plot and associated analysis. In Figure B.2, the data set WORK.FAN that contains the data for the engine fan example is selected, and the Weibull distribution is specified. For more information about the data for the engine fan example, see the section "Analysis of Right-Censored Data from a Single Population" on page 1206 in Chapter 18, "The RELIABILITY Procedure."

Figure B.2 Data Set and Distribution Window



Click the OK button, and the variable selection screen shown in Figure B.3 appears. The variable LIFETIME from the input data set is selected in Figure B.3 as the response variable, and the variable CENSOR is selected as the censoring indicator, with a value of 1 indicating censored lifetimes.

Figure B.3 Variable Selection Window



Clicking the OK button produces the probability plot window shown in Figure B.4. The RESULTS button enables you to view the tabular output from the RELIABILITY procedure.

You can choose procedure options and other analyses by selecting one of the menus at the top of the plot window. For example, you can specify additional plot options by selecting the plot menu, as shown in Figure B.5.

Figure B.4 Probability Plot Window

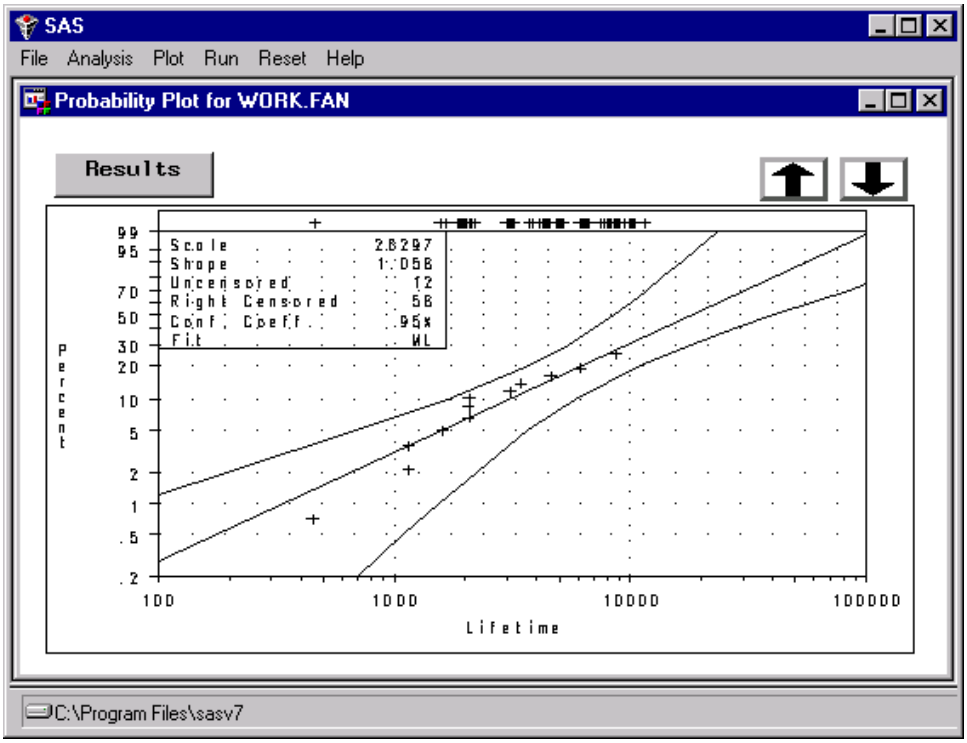
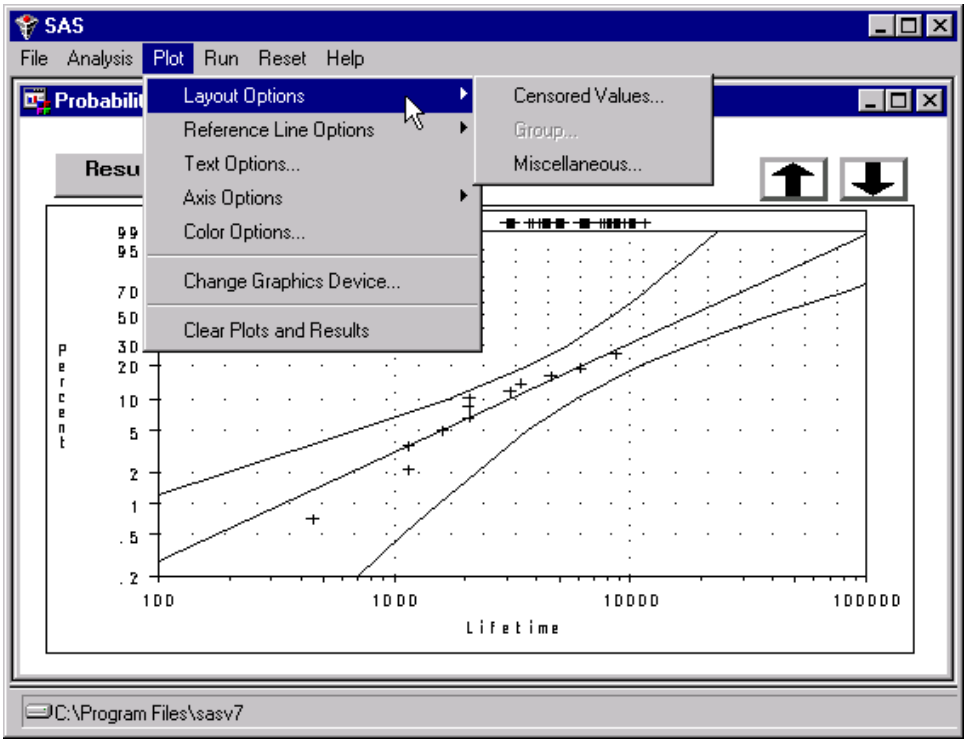


Figure B.5 Probability Plot Window



Appendix C
Functions

Contents

Introduction **2215**
Function Descriptions **2216**
AOQ2 Function 2217
ASN2 Function 2218
ATI2 Function 2220
BAYESACT Call 2221
C4 Function 2223
CUSUMARL Function 2224
D2 Function 2226
D3 Function 2227
EWMAARL Function 2228
PROBACC2 Function 2229
PROBBNML Function 2230
PROBHYPR Function 2232
PROBMED Function 2234
STDMED Function 2235
Details **2237**
Types of Sampling Plans 2237
Evaluating Single-Sampling Plans 2237
Evaluating Double-Sampling Plans 2239
Deriving Control Chart Constants 2239
References **2240**

Introduction

SAS/QC software provides specialized DATA step functions for computations related to control chart analysis, for Bayes analysis of screening designs, and for sampling plan evaluation. You can use these functions in DATA step programming statements. The following lists summarize these functions:

Table C.1 Functions for Control Chart Analysis

Function	Description
C4	expected value c_4 of the standard deviation of a sample from a normal population with unit standard deviation
CUSUMARL	average run length for a cumulative sum control chart scheme
D2	expected value d_2 of the range of a sample from a normal population with unit standard deviation
D3	standard deviation d_3 of the range of a sample from a normal population with unit standard deviation
EWMAARL	average run length for an EWMA scheme
PROBMED	cumulative distribution function of sample median
STDMED	standard deviation of median of a standard normal sample

Table C.2 Function for Bayes Analysis of Screening Designs

Function	Description
BAYESACT	posterior probabilities of variance contamination

Table C.3 Functions for Sampling Plan Evaluation

Function	Description
AOQ2	average outgoing quality for double-sampling plan
ASN2	average sample number for double-sampling plan
ATI2	average total inspection for double-sampling plan
PROBACC2	acceptance probability for double-sampling plan

In addition, the PROBBNML and PROBHYPR functions, which are provided in Base SAS software, are useful when evaluating single-sampling plans.

The twelve SAS/QC functions, together with the PROBBNML and PROBHYPR functions, are described in the "Function Descriptions" section. The section "Details" on page 2237, summarizes types of sampling plans and gives additional definitions.

Function Descriptions

This section describes the twelve SAS/QC functions and the related functions PROBBNML and PROBHYPR in alphabetical order.

AOQ2 Function

computes average outgoing quality for a double-sampling plan.

Syntax

AOQ2(*replacement*, $N, a_1, r_1, a_2, n_1, n_2, p$)

where

replacement	has the value 'REP' or 'NOREP', respectively, depending on whether nonconforming items are replaced with conforming items.
N	is the lot size, where $N \geq 2$.
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The AOQ2 function returns the average outgoing quality for a Type B double-sampling plan in which nonconforming items are replaced with conforming items (*replacement* is 'REP') or not replaced (*replacement* is 'NOREP'). For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 2237.

For replacement, the average outgoing quality is

$$\text{AOQ} = \frac{pP_{a_1}(N - n_1) + pP_{a_2}(N - n_1 - n_2)}{N}$$

and for no replacement, the average outgoing quality is

$$\text{AOQ} = \frac{pP_{a_1}(N - n_1)}{N - n_1 p} + \frac{pP_{a_2}(N - n_1 - n_2)}{N - n_1 p - n_2 p}$$

where, in both situations,

$$\begin{aligned} P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\ &= \text{probability of acceptance for first sample} \\ P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1)F(a_2 - d|n_2) \\ &= \text{probability of acceptance for second sample} \end{aligned}$$

and

$$
\begin{aligned}
f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\
&= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\
F(a|n) &= \sum_{d=0}^{a} f(d|n) \\
&= \text{probability that the number of nonconforming items is less than or equal to } a
\end{aligned}
$$

Examples

The first set of statements results in a value of 0.0148099904. The second set of statements results in a value of 0.0144743043.

```
data;
   aoq=aoq2('norep',120,0,2,1,13,13,0.18);
   put aoq;
run;

data;
   aoq=aoq2('rep',120,0,2,1,13,13,0.18);
   put aoq;
run;
```

ASN2 Function

computes the average sample number for a double-sampling plan.

Syntax

ASN2(*mode*,$a_1, r_1, a_2, n_1, n_2, p$)

where

mode	identifies whether sampling is under full inspection (*mode* is 'FULL') or semicurtailed inspection (*mode* is 'SEMI').
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$.
n_2	is the size of the second sample, where $n_2 \geq 1$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The ASN2 function returns the average sample number for a Type B double-sampling plan under full inspection (*mode* is 'FULL') or semicurtailed inspection (*mode* is 'SEMI'). For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 2237.

For full inspection, the average sample number is

$$\text{ASN} = n_1 + n_2[F(r_1 - 1|n_1) - F(a_1|n_1)]$$

and for semicurtailed inspection, the average sample number is

$$\text{ASN} = n_1 + \sum_{d=a_1+1}^{r_1-1} f(d|n_1)\left(n_2 F(a_2 - d|n_2) + \frac{r_2 - d}{p}[1 - F(r_2 - d|n_2 + 1)]\right)$$

where

$$\begin{aligned} f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\ &= \text{binomial probability that the number of nonconforming items} \\ &\quad \text{in a sample of size } n \text{ is exactly } d \\ F(a|n) &= \sum_{d=0}^{a} f(d|n) \\ &= \text{probability that the number of nonconforming items is less} \\ &\quad \text{than or equal to } a \end{aligned}$$

Examples

The first set of statements results in a value of 15.811418112. The second set of statements results in a value of 14.110408695.

```
data;
   asn=asn2('full',0,2,1,13,13,0.18);
   put asn;
run;

data;
   asn=asn2('semi',0,2,1,13,13,0.18);
   put asn;
run;
```

ATI2 Function

computes the average total inspection for a double-sampling plan.

Syntax

ATI2$(N, a_1, r_1, a_2, n_1, n_2, p)$

where

N	is the lot size, where $N \geq 2$.
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The ATI2 function returns the average total inspection for a Type B double-sampling plan. For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 2237.

The average total inspection is

$$\text{ATI} = n_1 P_{a_1} + (n_1 + n_2) P_{a_2} + N(1 - P_{a_1} - P_{a_2})$$

where

$$\begin{aligned} P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\ &= \text{probability of acceptance for first sample} \\ P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1) F(a_2 - d|n_2) \\ &= \text{probability of acceptance for second sample} \end{aligned}$$

and

$$
\begin{aligned}
f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\
&= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\
F(a|n) &= \sum_{d=0}^{a} f(d|n) \\
&= \text{probability that the number of nonconforming items is less than or equal to } a
\end{aligned}
$$

Examples

The following statements result in a value of 110.35046381:

```
data;
   ati=ati2(120,0,2,1,13,13,0.18);
   put ati;
run;
```

BAYESACT Call

computes posterior probabilities that observations are contaminated with a larger variance.

Syntax

CALL BAYESACT$(k, s, df, \alpha_1, \ldots, \alpha_n, y_1, \ldots, y_n, \beta_1, \ldots, \beta_n, p_0)$;

where

k	is the contamination coefficient, where $k \geq 1$.
s	is an independent estimate of σ, where $s \geq 0$.
df	is the number of degrees of freedom for s, where $df \geq 0$.
α_i	is the prior probability of contamination for the ith observation in the sample, where $i = 1, \ldots, n$ and n is the number of observations in the sample. Note that $0 \leq \alpha_i \leq 1$.
y_i	is the ith observation in the sample, where $i = 1, \ldots, n$ and n is the number of observations in the sample. When the BAYESACT call is used to perform a Bayes analysis of designs (see "Description" below), the y_is are estimates for effects.
β_i	is the variable that contains the returned posterior probability of contamination for the ith observation in the sample, where $i = 1, \ldots, n$ and n is the number of observations in the sample.
p_0	is the variable that contains the posterior probability that the sample is uncontaminated.

Description

The BAYESACT call computes posterior probabilities (β_i) that observations in a sample are *contaminated* with a larger variance than other observations and computes the posterior probability (p_0) that the entire sample is uncontaminated.

Specifically, the BAYESACT call assumes a normal random sample of n independent observations, with a mean of 0 (a centered sample) where some of the observations may have a larger variance than others:

$$\mathrm{Var}(y_i) = \begin{cases} \sigma^2 & \text{with probability } 1-\alpha_i \\ k^2\sigma^2 & \text{with probability } \alpha_i \end{cases}$$

where $i = 1, \ldots, n$. The parameter k is called the *contamination coefficient*. The value of α_i is the *prior probability* of contamination for the ith observation. Based on the prior probability of contamination for each observation, the call gives the posterior probability of contamination for each observation and the posterior probability that the entire sample is uncontaminated.

Box and Meyer (1986) suggest computing posterior probabilities of contamination for the analysis of saturated orthogonal factorial designs. Although these designs give uncorrelated estimates for effects, the significance of effects cannot be tested in an analysis of variance since there are no degrees of freedom for error. Box and Meyer suggest computing posterior probabilities of contamination for the effect estimates. The prior probabilities (α_i) give the likelihood that an effect will be significant, and the contamination coefficient (k) gives a measure of how large the significant effect will be. Box and Meyer recommend using $\alpha = 0.2$ and $k = 10$, implying that about 1 in 5 effects will be about 10 times larger than the remaining effects. To adequately explore posterior probabilities, examine them over a range of values for prior probabilities and a range of contamination coefficients.

If an independent estimate of σ is unavailable (as is the case when the y_is are effects from a saturated orthogonal design), use 0 for s and df in the BAYESACT call. Otherwise, the call assumes s is proportional to the square root of a χ^2 random variable with df degrees of freedom. For example, if the y_is are estimated effects from an orthogonal design that is not saturated, then use the BAYESACT call with s equal to the estimated standard error of the estimates and df equal to the degrees of freedom for error.

From Bayes' theorem, the posterior probability that y_i is contaminated is

$$\beta_i(\sigma) = \frac{\alpha_i f(y_i; 0, k^2\sigma^2)}{\alpha_i f(y_i; 0, k^2\sigma^2) + (1-\alpha_i) f(y_i; 0, \sigma^2)}$$

for a given value of σ, where $f(x; \mu, \sigma)$ is the density of a normal distribution with mean μ and variance σ^2.

The probability that the sample is uncontaminated is

$$p = \prod_{i=1}^{n} (1 - \beta_i(\sigma))$$

Posterior probabilities that are independent of σ are derived by integrating $\beta_i(\sigma)$ and p over a noninformative prior for σ. If an estimate of σ is available (when $df > 0$), it is appropriately incorporated. Refer to Box and Meyer (1986) for details.

Examples

The statements

```
data;
   retain post1-post7 postnone;
   call bayesact(10,0,0,
          0.2,   0.2,   0.2,   0.2,   0.2,     0.2,   0.2,
      -5.4375,1.3875,8.2875,0.2625,1.7125,-11.4125,1.5875,
        post1, post2, post3, post4, post5,   post6, post7,
      postnone);
run;
```

return the following posterior probabilities:

```
POST1      0.42108
POST2      0.037412
POST3      0.53438
POST4      0.024679
POST5      0.050294
POST6      0.64329
POST7      0.044408
POSTNONE   0.28621
```

The probability that the sample is uncontaminated is 0.28621. A situation where this BAYESACT call would be appropriate is a saturated 2^7 design in 8 runs, where the estimates for main effects are as shown in the function above (-5.4375, 1.3875, . . . , 1.5875).

C4 Function

computes the expected value of the standard deviation of n independent normal random variables.

Syntax

C4(n)

where n is the sample size, with $n \geq 2$.

Description

The C4 function returns the expected value of the standard deviation of n independent, normally distributed random variables with the same mean and with standard deviation of 1. This expected value is referred to as the control chart constant c_4.

The value c_4 is calculated as

$$c_4 = \frac{\Gamma(\frac{n}{2})\sqrt{2/(n-1)}}{\Gamma(\frac{n-1}{2})}$$

where $\Gamma(\cdot)$ is the gamma function. As n grows, c_4 is asymptotically equal to $(4n-4)/(4n-3)$.

For more information, refer to the American Society for Quality Control (1983), the American Society for Testing and Materials (1976), Montgomery (1996), and Wadsworth, Stephens, and Godfrey (1986).

In other chapters, c_4 is written as $c_4(n)$ to emphasize the dependence on n.

You can use the constant c_4 to calculate an unbiased estimate ($\hat{\sigma}$) of the standard deviation σ of a normal distribution from the sample standard deviation of n observations:

$$\hat{\sigma} = (\text{sample standard deviation})/c_4$$

where the sample standard deviation is calculated using $n - 1$ in the denominator. In the SHEWHART procedure, c_4 is used to calculate control limits for s charts, and it is used in the estimation of the process standard deviation based on subgroup standard deviations.

Examples

The following statements result in a value of 0.939985603:

```
data;
   constant=c4(5);
   put constant;
run;
```

CUSUMARL Function

computes the average run length for a cumulative sum control chart scheme.

Syntax

CUSUMARL(*type*,δ, h, k <,*headstart*>)

where

type	indicates a one-sided or two-sided scheme. Valid values are 'ONESIDED' or 'O' for a one-sided scheme, and 'TWOSIDED' or 'T' for a two-sided scheme.
δ	is the shift to be detected, expressed as a multiple of the process standard deviation (σ).
h	is the decision interval (one-sided scheme) or the vertical distance between the origin and the upper arm of the V-mask (two-sided scheme), each time expressed as a positive value in standard units (a multiple of $\sigma/\sqrt{n}$, where n is the subgroup sample size).
k	is the reference value (one-sided scheme) or the slope of the lower arm of the V-mask (two-sided scheme), each time expressed as a positive value in standard units (a multiple of $\sigma/\sqrt{n}$, where n is the subgroup sample size).
headstart	is the headstart value (optional) expressed in standard units (a multiple of $\sigma/\sqrt{n}$, where n is the subgroup sample size). The default *headstart* is zero. For details, refer to Lucas and Crosier (1982).

Description

The CUSUMARL function returns the average run length of one-sided and two-sided cumulative sum schemes with parameters as described above. The notation is consistent with that used in the CUSUM procedure.

For a one-sided scheme, the average run length is calculated using the integral equation method (with 24 Gaussian points) described by Goel and Wu (1971) and Lucas and Crosier (1982).

For a two-sided scheme with no *headstart*, the average run length (ARL) is calculated using the fact that

$$(\mathrm{ARL})^{-1} = (\mathrm{ARL}_+)^{-1} + (\mathrm{ARL}_-)^{-1}$$

where ARL_+ and ARL_- denote the average run lengths of the equivalent one-sided schemes for detecting a shift of the same magnitude in the positive direction and in the negative direction, respectively.

For a two-sided scheme with a nonzero *headstart*, the ARL is calculated by combining average run lengths for one-sided schemes as described in Appendix A.1 of Lucas and Crosier 1982, p. 204.

For a specified shift δ, you can use the CUSUMARL function to design a cusum scheme by first calculating average run lengths for a range of values of h and k and then choosing the combination of h and k that yields a desired average run length.

You can also use the CUSUMARL function to interpolate published tables of average run lengths.

Examples

The following three sets of statements result in the values 4.1500826715, 4.1500836225, and 4.1061588131, respectively.

```
data;
   arl=cusumarl('twosided',2.5,8,0.25);
   put arl;
run;

data;
   arl=cusumarl('onesided',2.5,8,0.25);
   put arl;
run;

data;
   arl=cusumarl('o',2.5,8,0.25,0.1);
   put arl;
run;
```

D2 Function

computes the expected value of the sample range.

Syntax

D2(n)

where n is the sample size, with $2 \leq n \leq 25$.

Description

The D2 function returns the expected value of the sample range of n independent, normally distributed random variables with the same mean and a standard deviation of 1. This expected value is referred to as the control chart constant d_2. The values returned by the D2 function are accurate to ten decimal places.

The value d_2 can be expressed as

$$d_2 = \int_{-\infty}^{\infty} \left[1 - (1 - \Phi(x))^n - (\Phi(x))^n\right] \, dx$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Refer to Tippett (1925). In other chapters, d_2 is written as $d_2(n)$ to emphasize the dependence on n.

In the SHEWHART procedure, d_2 is used to calculate control limits for r charts, and it is used in the estimation of the process standard deviation based on subgroup ranges. Also refer to the American Society for Quality Control (1983), the American Society for Testing and Materials (1976), Kume (1985), Montgomery (1996), and Wadsworth, Stephens, and Godfrey (1986).

You can use the constant d_2 to calculate an unbiased estimate ($\hat{\sigma}$) of the standard deviation σ of a normal distribution from the sample range of n observations:

$$\hat{\sigma} = (\text{sample range})/d_2$$

Note that the statistical efficiency of this estimate relative to that of the sample standard deviation decreases as n increases.

Examples

The following statements result in a value of 2.3259289473:

```
data;
   constant=d2(5);
   put constant;
run;
```

D3 Function

computes the standard deviation of the range of n independent normal random variables.

Syntax

D3(n)

where n is the sample size, with $2 \leq n \leq 25$.

Description

The D3 function returns the standard deviation of the range of n independent, normally distributed random variables with the same mean and with unit standard deviation. The standard deviation returned is referred to as the control chart constant d_3. The values returned by the D3 function are accurate to ten decimal places.

The value d_3 can be expressed as

$$d_3 = \sqrt{2\int_{-\infty}^{\infty}\int_{-\infty}^{y} f(x,y)\,dx\,dy - d_2^2}$$

where

$$f(x,y) = 1 - (\Phi(y))^n - (1-\Phi(x))^n + (\Phi(y) - \Phi(x))^n$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and d_2 is the expected range. Refer to Tippett (1925).

In other chapters d_3 is written as $d_3(n)$ to emphasize the dependence on n.

In the SHEWHART procedure, d_3 is used to calculate control limits for r charts, and it is used in the estimation of the process standard deviation based on subgroup ranges.

For more information, refer to the American Society for Quality Control (1983), the American Society for Testing and Materials (1976), Montgomery (1996), and Wadsworth, Stephens, and Godfrey (1986).

You can use the constant d_3 to calculate an unbiased estimate ($\hat{\sigma}$) of the standard deviation σ_R of the range of a sample of n normally distributed observations from the sample range of n observations:

$$\hat{\sigma}_R = (\text{sample range})(d_3/d_2)$$

You can use the D2 function to calculate d_2.

Examples

The following statements result in a value of 0.8640819411:

```
data;
   constant=d3(5);
   put constant;
run;
```

EWMAARL Function

computes the average run length for an exponentially weighted moving average.

Syntax

EWMAARL(δ, r, k)

where

δ	is the shift to be detected, expressed as a multiple of the process standard deviation (σ), where $\delta \geq 0$.
r	is the weight factor for the current subgroup mean in the EWMA, where $0 < r \leq 1$. If $r = 1$, the EWMAARL function returns the average run length for a Shewhart chart for means. Refer to Wadsworth, Stephens, and Godfrey (1986). If $r \leq 0.05$, $k \geq 3$, and $\delta < 0.10$, the algorithm used is unstable. However, note that the EWMA behaves like a cusum when $r \rightarrow 0$, and in this case the CUSUMARL function is applicable.
k	is the multiple of σ used to define the control limits, where $k \geq 0$. Typically $k = 3$.

Description

The EWMAARL function computes the average run length for an exponentially weighted moving average (EWMA) scheme using the method of Crowder (1987a, b). The notation used in the preceding list is consistent with that used in the MACONTROL procedure.

For a specified shift δ, you can use the EWMAARL function to design an exponentially weighted moving average scheme by first calculating average run lengths for a range of values of r and k and then choosing the combination of r and k that yields a desired average run length.

Examples

The following statements specify a shift of 1σ, a weight factor of 0.25, and 3σ control limits. The EWMAARL function returns an average run length of 11.154267016.

```
data;
   arl=ewmaarl(1.00,0.25,3.0);
   put arl;
run;
```

PROBACC2 Function

computes the acceptance probability for a double-sampling plan.

Syntax

PROBACC2$(a_1, r_1, a_2, n_1, n_2, D, N)$

PROBACC2$(a_1, r_1, a_2, n_1, n_2, p)$

where

a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 > a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
D	is the number of nonconforming items in the lot, where $0 \leq D \leq N$.
N	is the lot size, where $N \geq 2$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The PROBACC2 function returns the acceptance probability for a double-sampling plan of Type A if you specify the parameters D and N, and it returns the acceptance probability for a double-sampling plan of Type B if you specify the parameter p. For details on Type A and Type B double-sampling plans, see "Types of Sampling Plans" on page 2237.

For either type of sampling plan, the acceptance probability is calculated as

$$P_{a_1} + P_{a_2}$$

where

$$\begin{aligned} P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\ &= \text{probability of acceptance for first sample} \\ P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1)F(a_2 - d|n_2) \\ &= \text{probability of acceptance for second sample} \end{aligned}$$

and

$$
\begin{aligned}
f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\
&= \text{binomial probability that the number of nonconforming items in a sample of size } n \text{ is exactly } d \\
F(a|n) &= \sum_{d=0}^{a} f(d|n) \\
&= \text{probability that the number of nonconforming items is less than or equal to } a
\end{aligned}
$$

These probabilities are determined from either the hypergeometric distribution (Type A sampling) or the binomial distribution (Type B sampling).

Examples

The first set of statements results in a value of 0.2396723824. The second set of statements results in a value of 0.0921738126.

```
data;
   prob=probacc2(1,4,3,50,100,10,200);
   put prob;
run;

data;
   prob=probacc2(0,2,1,13,13,0.18);
   put prob;
run;
```

PROBBNML Function

computes the probability that an observation from a binomial(n, p) distribution will be less than or equal to m.

Syntax

PROBBNML(p, n, m)

where

p	is the probability of success for the binomial distribution, where $0 \leq p \leq 1$. In terms of acceptance sampling, p is the probability of selecting a nonconforming item.
n	is the number of independent Bernoulli trials in the binomial distribution, where $n \geq 1$. In terms of acceptance sampling, n is the number of items in the sample.
m	is the number of successes, where $0 \leq m \leq n$. In terms of acceptance sampling, m is the number of nonconforming items.

Description

The PROBBNML function returns the probability that an observation from a binomial distribution (with parameters n and p) is less than or equal to m. To compute the probability that an observation is equal to a given value m, compute the difference of two values for the cumulative binomial distribution.

In terms of acceptance sampling, the function returns the probability of finding m or fewer nonconforming items in a sample of n items, where the probability of a nonconforming item is p. To find the probability that the sample contains exactly m nonconforming items, compute the difference between PROBBNML(p, n, m) and PROBBNML($p, n, m - 1$).

In addition to using the PROBBNML function to return the probability of acceptance, the function can be used in calculations for average sample number, average outgoing quality, and average total inspection in Type B single-sampling. See "Evaluating Single-Sampling Plans" on page 2237 for details.

The PROBBNML function computes

$$\sum_{j=0}^{m} \binom{n}{j} p^j (1-p)^{n-j}$$

where m, n, and p are defined in the preceding list.

Examples

The following statements compute the probability that an observation from a binomial distribution with p = 0.05 and n = 10 is less than or equal to 4:

```
data;
   prob=probbnml(0.05,10,4);
   put prob;
run;
```

These statements result in the value 0.9999363102. In terms of acceptance sampling, for a sample of size 10 where the probability of a nonconforming item is 0.05, the probability of finding 4 or fewer nonconforming items is 0.9999363102.

The following statements compute the probability that an observation from a binomial distribution with p = 0.05 and n = 10 is exactly 4:

```
data;
   p=probbnml(0.05,10,4) - probbnml(0.05,10,3);
   put p;
run;
```

These statements result in the value 0.0009648081.

For additional information on probability functions, refer to *SAS Functions and CALL Routines: Reference*.

PROBHYPR Function

computes the probability that an observation from a hypergeometric distribution is less than or equal to *x*.

Syntax

PROBHYPR(*N*, *K*, *n*, *x* <, *r* >)

where

N	is the population size for a hypergeometric distribution. In terms of acceptance sampling, *N* is the lot size.
K	is the number of items in the category of interest in the population. In terms of acceptance sampling, *K* is the number of nonconforming items in a lot.
n	is the sample size for a hypergeometric distribution. In terms of acceptance sampling, *n* is the sample size.
x	is the number of items from the category of interest in the sample. In terms of acceptance sampling, *x* is the number of nonconforming items in the sample.
r	is optional and gives the odds ratio for the extended hypergeometric distribution. For the standard hypergeometric distribution, *r* = 1; this value is the default. In acceptance sampling, typically *r* = 1.

Restrictions on items in the syntax are given in the following equations:

$$1 \leq N$$
$$0 \leq K \leq N$$
$$0 \leq n \leq N$$
$$\max(0, K + n - N) \leq x \leq \min(K, n)$$
N, K, n and x are integers

Description

The PROBHYPR function returns the probability that an observation from an extended hypergeometric distribution with parameters *N*, *K* and *n* and an odds ratio of *r* is less than or equal to *x*. The default for *r* is 1 and leads to the usual hypergeometric distribution.

In terms of acceptance sampling, if *r* = 1, the PROBHYPR function gives the probability of *x* or fewer nonconforming items in a sample of size *n* taken from a lot containing *N* items, *K* of which are nonconforming, when sampling is done without replacement. Typically *r* = 1 in acceptance sampling.

For example, suppose an urn contains red and white balls, and you are interested in the probability of selecting a white ball. If *r* = 1, the function returns the probability of selecting *x* white balls when given the population size (number of balls in the urn), sample size (number of balls taken from the urn), and number of white balls in the population (urn).

If, however, the probability of selecting a white ball differs from the probability of selecting a red ball, then $r \neq 1$. Suppose an urn contains one white ball and one red ball, and the probability of choosing the red ball is higher than the probability of choosing the white ball. This might occur if the red ball were larger than the white ball, for example. Given the probabilities of choosing a red ball and a white ball when an urn contains

one of each, you calculate r and use the value in the PROBHYPR function. Returning to the case where an urn contains many balls with $r \neq 1$, the function gives the probability of selecting x white balls when given the number of balls in the urn, the number of balls taken from the urn, the number of white balls in the urn, and the relative probability of selecting a white ball or a red ball.

The PROBHYPR function is used to evaluate Type A single-sampling plans. See "Evaluating Single-Sampling Plans" on page 2237 for details.

If $r = 1$ (the default), the PROBHYPR function calculates probabilities from the usual hypergeometric distribution:

$$\Pr[X \leq x] = \sum_{i=0}^{x} P_i$$

where

$$P_i = \begin{cases} \dfrac{\binom{K}{i}\binom{N-K}{n-i}}{\binom{N}{n}} & \text{if } \max(0, K+n-N) \leq i \leq \min(K, n) \\ 0 & \text{otherwise} \end{cases}$$

The PROBHYPR function accepts values other than 1 for r, and in these cases, it calculates the probability for the extended hypergeometric distribution:

$$\Pr[X_1 \leq x | X_1 + X_2 = n] = \sum_{i=o}^{x} P_i$$

where

$$P_i = \begin{cases} \dfrac{\binom{K}{i}\binom{N-K}{n-i}r^j}{\sum_{j=0}^{n}\binom{K}{j}\binom{N-K}{n-j}r^j} & \text{if } \max(0, K+n-N) \leq i \leq \min(K, n) \\ 0 & \text{otherwise} \end{cases}$$

where

X_1	is binomially distributed with parameters K and p_1.
X_2	is binomially distributed with parameters N-K and p_2.
q_1	$= 1 - p_1$
q_2	$= 1 - p_2$
r	$= (p_1 q_2)/(p_2 q_1)$

For details on the extended hypergeometric distribution, refer to Johnson and Kotz (1969).

Examples

Suppose you take a sample of size 10 (without replacement) from an urn that contains 200 balls, 50 of which are white. The remaining 150 balls are red. The following statements calculate the probability that your sample contains 2 or fewer white balls:

```
data;
   y=probhypr(200,50,10,2);
   put y;
run;
```

These statements result in a value of 0.5236734081. Now, suppose the probability of selecting a red ball does not equal the probability of selecting a white ball. Specifically, suppose the probability of choosing a red ball is $p_2 = 0.4$ and the probability of choosing a white ball is $p_1 = 0.2$. Calculate r as

$$r = \frac{p_1 q_2}{p_2 q_1} = \frac{(0.2)(0.6)}{(0.4)(0.8)} = 0.375$$

With $r = 0.375$, the probability of choosing 2 or fewer white balls from an urn that contains 200 balls, 50 of which are white, is calculated using the following statements:

```
data;
   y=probhypr(200,50,10,2,0.375);
   put y;
run;
```

These statements return a value of 0.9053936127. See "Evaluating Single-Sampling Plans" on page 2237 for another example.

For additional information on probability functions, refer to *SAS Functions and CALL Routines: Reference*.

PROBMED Function

computes cumulative probabilities for the sample median.

Syntax

PROBMED(n, x)

where

n	is the sample size.
x	is the point of interest; that is, the PROBMED function calculates the probability that the median is less than or equal to x.

Description

The PROBMED function computes the probability that the sample median is less than or equal to x for a sample of n independent, standard normal random variables (mean 0, variance 1).

Let n represent the sample size and $X_{(i)}$ represent the ith order statistic. Then, when n is odd, the function calculates

$$\Pr[X_{((n+1)/2)} \leq x] = \mathrm{I}_{\Phi(x)}\left(\frac{n+1}{2}, \frac{n+1}{2}\right)$$

where

$$\mathrm{I}_p(a,b) = \frac{1}{\mathrm{B}(a,b)} \int_0^p t^{a-1}(1-t)^{b-1}\, dt$$

and $\mathrm{B}(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$, where $\Gamma(\cdot)$ is the gamma function. If n is even, the PROBMED function calculates

$$\Pr\left[\frac{X_{(n/2)} + X_{((n/2)+1)}}{2} \leq x\right] =$$

$$\frac{2}{\mathrm{B}(\frac{n}{2}, \frac{n}{2})} \int_{-\infty}^{x} \left\{[1-\Phi(u)]^{n/2} - [1-\Phi(2x-u)]^{n/2}\right\} [\Phi(u)]^{(n/2)-1}\, \phi(u)\, du$$

where $\mathrm{B}(n/2, n/2) = [\Gamma(n/2)]^2/\Gamma(n)$ and $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal cumulative distribution function and density function, respectively.

For more information, refer to David (1981).

Examples

The statements

```
data;
   b=probmed(5,-0.1);
   put b;
run;
```

result in a value of 0.4256380897.

STDMED Function

computes the standard deviation of a sample median.

Syntax

STDMED(n)

where n is the sample size.

Description

The STDMED function gives the standard deviation of the median of a normally distributed sample with a mean of 0 and a variance of 1. This function gives the standard error used to determine the width of the control limits for charts produced by the MCHART and MRCHART statements in PROC SHEWHART.

Let n represent the sample size and $X_{(i)}$ represent the ith order statistic. Then, when n is odd, the STDMED function calculates $\sqrt{\mathrm{Var}(X_{((n+1)/2)})}$, where

$$\mathrm{Var}(X_{((n+1)/2)}) = \frac{1}{\mathrm{B}\left(\frac{n+1}{2}, \frac{n+1}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{(n-1)/2} [1 - \Phi(x)]^{(n-1)/2} \, \phi(x) \, dx$$

where $\mathrm{B}(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ and $\Gamma(\cdot)$ is the gamma function, $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $\phi(\cdot)$ is the corresponding density function.

If n is even, the function calculates the square root of the following:

$$\mathrm{Var}\left[\frac{X_{(n/2)} + X_{((n/2)+1)}}{2}\right] =$$

$$(1/4)\left[\mathrm{E}(X^2_{(n/2)}) + \mathrm{E}(X^2_{((n/2)+1)}) + 2\mathrm{E}(X_{(n/2)} X_{((n/2)+1)})\right]$$

where

$$\mathrm{E}(X^2_{(n/2)}) = \frac{2}{\mathrm{B}\left(\frac{n}{2}, \frac{n}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{(n/2)-1} [1 - \Phi(x)]^{n/2} \, \phi(x) \, dx$$

$$\mathrm{E}(X^2_{((n/2)+1)}) = \frac{2}{\mathrm{B}\left(\frac{n}{2}, \frac{n}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{n/2} [1 - \Phi(x)]^{(n/2)-1} \, \phi(x) \, dx$$

$$\mathrm{E}(X_{(n/2)} X_{((n/2)+1)}) = \frac{n}{\mathrm{B}\left(\frac{n}{2}, \frac{n}{2}\right)} \int_{-\infty}^{\infty} \int_{-\infty}^{y} xy [\Phi(x)]^{(n/2)-1} [1 - \Phi(y)]^{(n/2)-1} \, \phi(x) \, \phi(y) \, dx \, dy$$

For more details, refer to David (1981), Kendall and Stuart 1977, p. 252, and Sarhan and Greenberg (1962).

Examples

These statements use a loop to calculate the standard deviation of the median for sample sizes from 6 to 12:

```
data;
   do n=6 to 12;
      s=stdmed(n);
      put s;
      output;
   end;
run;
```

The statements produce these values:

```
0.4634033519
0.4587448763
0.410098592
0.4075552495
0.3719226208
0.3703544701
0.3428063408
```

Details

Types of Sampling Plans

In single sampling, a random sample of n items is selected from a lot of size N. If the number d of nonconforming (defective) items found in the sample is less than or equal to an acceptance number c, the lot is accepted. Otherwise, the lot is rejected.

In double sampling, a sample of size n_1 is drawn from the lot, and the number d_1 of nonconforming items is counted. If d_1 is less than or equal to an acceptance number a_1, the lot is accepted, and if d_1 is greater than or equal to a rejection number r_1, the lot is rejected. Otherwise, if $a_1 < d_1 < r_1$, a second sample of size n_2 is taken, and the number of nonconforming items d_2 is counted. Then if $d_1 + d_2$ is less than or equal to an acceptance number a_2, the lot is accepted, and if $d_1 + d_2$ is greater than or equal to a rejection number $r_2 = a_2 + 1$, the lot is rejected. This notation follows that of Schilling (1982). Note that some authors, including Montgomery (1996), define the first rejection number using a strict inequality.

In *Type A sampling*, the sample is intended to represent a single, finite-sized lot, and the characteristics of the sampling plan depend on D, the number of nonconforming items in the lot, as well as N, n, and c.

In *Type B sampling*, the sample is intended to represent a series of lots (or the lot size is effectively infinite), and the characteristics of the sampling plan depend on p, the proportion of nonconforming items produced by the process, as well as n and c.

A hypergeometric model is appropriate for Type A sampling, and a binomial model is appropriate for Type B sampling.

Evaluating Single-Sampling Plans

You can use the Base SAS functions PROBBNML and PROBHYPR to evaluate single-sampling plans. Measures of the performance of single-sampling plans include

- the probability of acceptance P_a
- the average sample number ASN
- the average outgoing quality AOQ
- the average total inspection ATI

Probability of Acceptance

Since P_a is the probability of finding c or fewer defectives in the sample, you can calculate the acceptance probability using the function PROBHYPR(N, D, n, c) for Type A sampling and the function PROBBNML(p, n, c) for Type B sampling.

For example, the following statements calculate P_a for the plan n = 20, c = 1 when sampling from a single lot of size N = 120 that contains D = 22 nonconforming items, resulting in a value of 0.0762970752:

```
data;
   prob=probhypr(120,22,20,1);
   put prob;
run;
```

Similarly, the following statements calculate P_a for the plan $n = 20$, $c = 1$ when sampling from a series of lots for which the proportion of nonconforming items is $p = 0.18$, resulting in a value of 0.1018322793:

```
data;
   prob=probbnml(0.18,20,1);
   put prob;
run;
```

Other Measures of Performance

The measures ASN, AOQ, and ATI are meaningful only for Type B sampling and can be calculated using the PROBBNML function. For reference, the following equations are provided.

Average sample number: Following the notation of Schilling (1982), let $F(c|n)$ denote the probability of finding c or fewer nonconforming items in a sample of size n. Note that $F(c|n)$ is equivalent to PROBBNML(p, n, c). Then, depending on the mode of inspection, the average sample number can be expressed as shown in the following table:

Mode of Inspection	ASN
Full	n
Semicurtailed	$nF(c\|n) + \dfrac{(c+1)(1-F(c+1\|n+1))}{p}$
Fully curtailed	$\dfrac{(n-c)F(c\|n+1)}{1-p} + \dfrac{(c+1)(1-F(c+1\|n+1))}{p}$

Average outgoing quality can be expressed as

$$\text{AOQ} = \frac{p(N-n)F(c|n)}{N}$$

if the nonconforming items found are replaced with conforming items, and as

$$\text{AOQ} = \frac{p(N-n)F(c|n)}{N-np}$$

if the nonconforming items found are not replaced.

Average total inspection can be expressed as

$$\text{ATI} = n + (1 - F(c|n))(N-n)$$

Evaluating Double-Sampling Plans

The following list gives some measures for double-sampling plans. The formula for each measure is given in the section describing the corresponding function.

- the probability of acceptance, P_a, calculated with the PROBACC2 function
- the average sample number, ASN, calculated with the ASN2 function
- the average outgoing quality, AOQ, calculated with the AOQ2 function
- the average total inspection, ATI, calculated with the ATI2 function

Deriving Control Chart Constants

You can use the functions D2, D3, and C4 to calculate standard control chart constants that are derived from d_2, d_3 and c_4. For reference, the following equations for some of these constants are provided:

$$
\begin{aligned}
A_2 &= k/(d_2\sqrt{n}) \\
A_3 &= k/(c_4\sqrt{n}) \\
B_3 &= \max(0, 1-(k/c_4)\sqrt{1-c_4^2}\,) \\
B_4 &= 1+(k/c_4)\sqrt{1-c_4^2} \\
B_5 &= \max(0, c_4 - k\sqrt{1-c_4^2}) \\
B_6 &= c_4 + k\sqrt{1-c_4^2} \\
c_5 &= \sqrt{1-c_4^2} \\
D_1 &= \max(0, d_2 - kd_3) \\
D_2 &= d_2 + kd_3 \\
D_3 &= \max(0, 1-kd_3/d_2) \\
D_4 &= 1+kd_3/d_2 \\
E_2 &= k/d_2 \\
E_3 &= k/c_4
\end{aligned}
$$

In the preceding equations, k is the multiple of standard error (k = 3 in the case of 3σ limits), and n is the subgroup sample size. The use of these control chart constants is discussed in the American Society for Quality Control (1983), the American Society for Testing and Materials (1976), Montgomery (1996), and Wadsworth, Stephens, and Godfrey (1986).

Although you do not ordinarily need to calculate control chart constants when using the SHEWHART procedure, you may find the D2, D3, and C4 functions useful for creating LIMITS= data sets that contain control limits to be read by the SHEWHART procedure.

References

American Society for Quality Control (1983). *ASQC Glossary and Tables for Statistical Quality Control*. Milwaukee: ASQC.

American Society for Testing and Materials (1976). *ASTM Manual on Presentation of Data and Control Chart Analysis*. Philadelphia: ASTM.

Box, G. E. P., and Meyer, D. R. (1986). "An Analysis for Unreplicated Fractional Factorials." *Technometrics* 28:11–18.

Crowder, S. V. (1987a). "Average Run Lengths of Exponentially Weighted Moving Average Charts." *Journal of Quality Technology* 19:161–164.

Crowder, S. V. (1987b). "A Simple Method for Studying Run-Length Distributions of Exponentially Weighted Moving Average Charts." *Technometrics* 29:401–408.

David, H. A. (1981). *Order Statistics*. 2nd ed. New York: John Wiley & Sons.

Goel, A. L., and Wu, S. M. (1971). "Determination of A.R.L. and a Contour Nomogram for Cusum Charts to Control Normal Mean." *Technometrics* 13:221–230.

Johnson, N. L., and Kotz, S. (1969). *Discrete Distributions*. New York: John Wiley & Sons.

Kendall, M. G., and Stuart, A. (1977). *The Advanced Theory of Statistics.* 4th ed. Vol. 1. New York: Macmillan.

Kume, H. (1985). *Statistical Methods for Quality Improvement*. Tokyo: AOTS Chosakai.

Lucas, J. M., and Crosier, R. B. (1982). "Fast Initial Response for CUSUM Quality Control Schemes: Give Your CUSUM a Head Start." *Technometrics* 24:199–205.

Montgomery, D. C. (1996). *Introduction to Statistical Quality Control*. 3rd ed. New York: John Wiley & Sons.

Sarhan, A. H., and Greenberg, B. G. (1962). *Contributions to Order Statistics*. New York: John Wiley & Sons.

Schilling, E. G. (1982). *Acceptance Sampling in Quality Control*. New York: Marcel Dekker.

Tippett, L. H. C. (1925). "On the Extreme Individuals and the Range of Samples Taken from a Normal Population." *Biometrika* 17:364–387.

Wadsworth, H. M., Stephens, K. S., and Godfrey, A. B. (1986). *Modern Methods for Quality Control and Improvement*. New York: John Wiley & Sons.

Appendix D
Special Fonts in SAS/QC Software

Introduction

Twelve special fonts are provided with SAS/QC software. The SAS/QC procedures use these fonts to display symbols such as $\bar{X}$ and σ, which are commonly encountered in statistical quality improvement applications. The procedure selects the special font that most closely matches the font used to display the other text on the graph. You can also use these fonts to display special symbols in the titles and footnotes of graphs. See the section "Step 1: Preliminary Mean and Standard Deviation Charts" on page 2102 for an example in which a special font is used to create a title for a control chart.

Font Selection

Each of the four special software fonts matches a particular SAS/GRAPH font, as shown in Table D.1. If you are using a software font for the general text on your graph, choose the special software font corresponding to the SAS/GRAPH font in the table that provides the closest match.

Table D.1 Special Software Fonts

Special Software Font	Matching SAS/GRAPH Font
QCFONT1	SIMPLEX
QCFONT2	DUPLEX
QCFONT3	SWISSE
QCFONT4	SWISS

Eight special TrueType fonts are provided to match a variety of general fonts based on three criteria, as summarized in Table D.2.

Table D.2 Special TrueType Fonts

Serifs	Bold	Italic	Special TrueType Font
no	no	no	Arial Symbol
no	no	yes	Arial Symbol/Italic
no	yes	no	Arial Symbol/Bold
no	yes	yes	Arial Symbol/Bold Italic
yes	no	no	Times New Roman Symbol
yes	no	yes	Times New Roman Symbol/Italic
yes	yes	no	Times New Roman Symbol/Bold
yes	yes	yes	Times New Roman Symbol/Bold Italic

The following figures illustrate the four special software fonts. In each of the figures, the symbols are shown in the special font, and the title and the character codes are shown in the matching SAS/GRAPH font.

Figure D.1 QCFONT1 and SIMPLEX Fonts

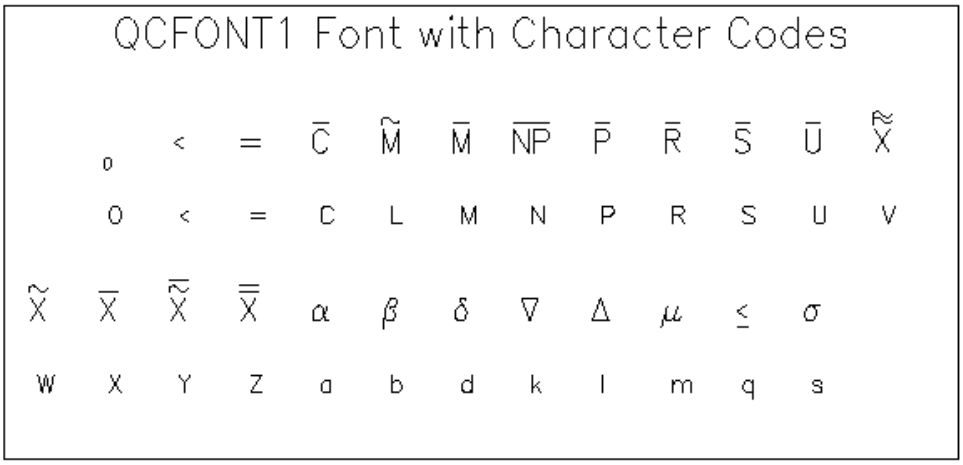


Figure D.2 QCFONT2 and DUPLEX Fonts

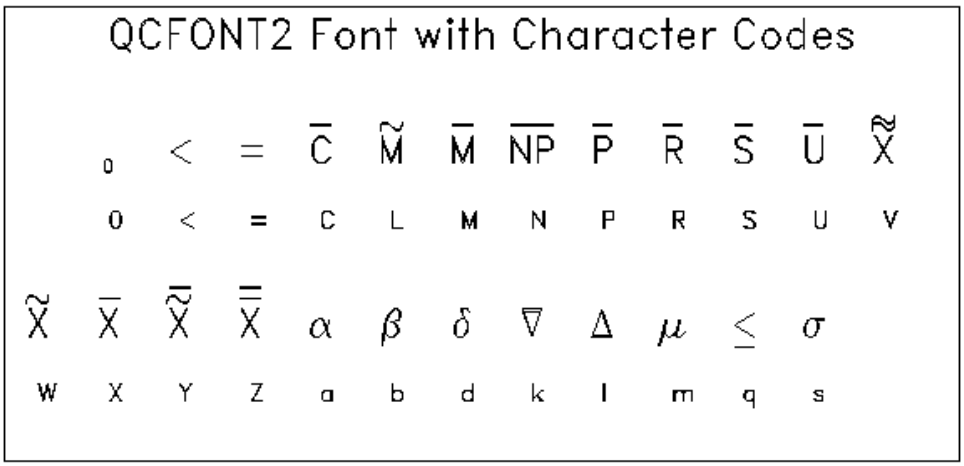


Figure D.3 QCFONT3 and SWISSE Fonts

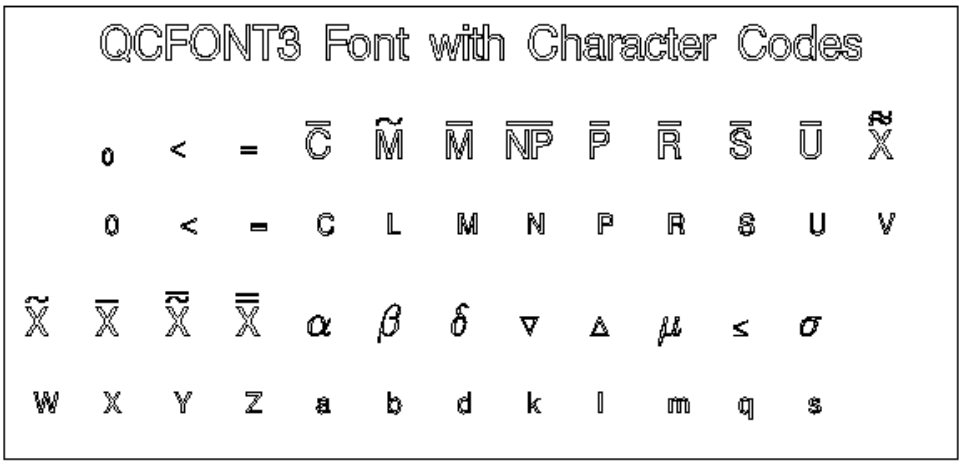


Figure D.4 QCFONT4 and SWISS Fonts

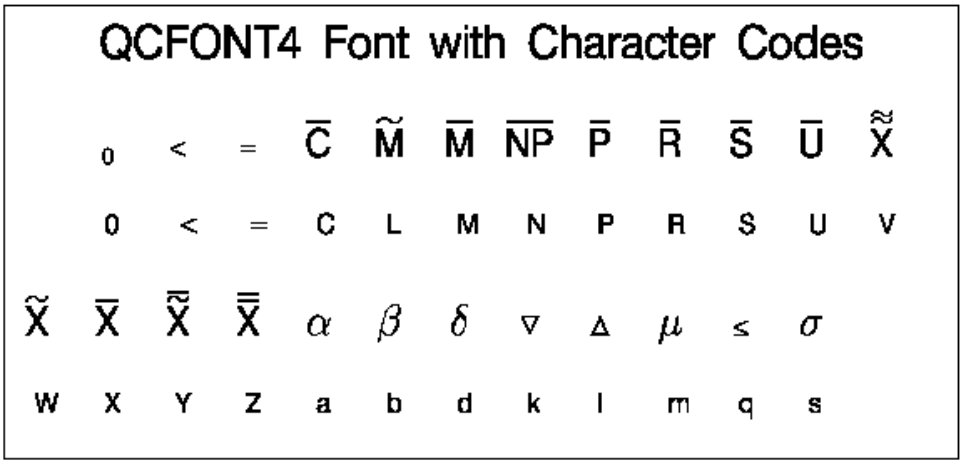


Table D.3 shows the character codes and corresponding symbols available in the special TrueType fonts.

Table D.3 Symbols in Special TrueType Fonts

Character	Hex Code	Symbol
	20	
0	30	0
<	3C	<
=	3D	=
C	43	$\overline{\mathrm{C}}$
D	44	$\triangle$
L	4C	$\widetilde{\mathrm{M}}$
M	4D	$\overline{\mathrm{M}}$
N	4E	$\overline{\mathrm{NP}}$
P	50	$\overline{\mathrm{P}}$
R	52	$\overline{\mathrm{R}}$
S	53	$\overline{\mathrm{S}}$
U	55	$\overline{\mathrm{U}}$
V	56	$\overline{\overline{\mathrm{X}}}$
W	57	$\widetilde{\mathrm{X}}$
X	58	$\overline{\mathrm{X}}$
Y	59	$\overline{\widetilde{\mathrm{X}}}$
Z	5A	$\overline{\overline{\mathrm{X}}}$
a	61	α
b	62	β
c	63	ξ
d	64	δ
e	65	ϵ
f	66	φ
g	67	γ
h	68	η
i	69	ι
j	6A	θ
k	6B	κ
l	6C	λ
m	6D	μ
n	6E	ν
o	6F	o
p	70	π
q	71	$\leq$
r	72	ρ
s	73	σ
t	74	τ
u	75	ϵ
v	76	∇
w	77	ω
x	78	χ
y	79	ψ
z	7A	ζ

Subject Index

A-optimal designs, *see* optimal designs, optimality criteria
aberration of a design, *see* minimum aberration
acceptance probability
  double-sampling plan, 2229, 2230
  PROBACC2 function, 2229
  Type A sampling, 2232–2234, 2237, 2238
  Type B sampling, 2230, 2231, 2237, 2238
acceptance sampling
  average outgoing quality, 2217, 2218, 2237, 2238
  average sample number, 2218, 2219, 2237, 2239
  average total inspection, 2220, 2221, 2237, 2238
  evaluating double-sampling plans, 2239
  evaluating single-sampling plans, 2237, 2238
  probability of choosing nonconforming items, 2230–2234
  types of sampling plans, 2237
alias structure
  breaking links, example, 664, 666
  details, 653
  example, 660, 661, 663, 678–681
  listing with GLM procedure, 1053
  syntax, 632
analysis of variance, 700
Anderson-Darling statistic, 229, 352
Anderson-Darling test, 209
annotating
  example, 891
  Shewhart charts, 2056
ANOM boxcharts
  axis labels, 74
  box-and-whisker plots, description of, 62
  central line, 63
  decision limit equations, 63, 64
  examples, advanced, 75
  examples, introductory, 45
  missing values, 74
  notation, 62
  ODS tables, 68
  options summarized by function, 54, 61
  overview, 44
  reading group summary statistics, 47, 48, 50, 72, 73
  reading preestablished decision limits, 70, 71
  reading raw measurements, 45–47, 70
  reading summary statistics and decision limits, 52, 53, 73, 74
  saving decision limits, 51, 66
  saving group summary statistics, 50, 51, 66, 67
  saving summary statistics and decision limits, 52, 53, 67, 68
  syntax, 53
ANOM charts
  group sample size, 183
  options dictionary, 183
ANOM charts for a Two-Way Layout
  central line, 149
  decision limit equations, 149
  notation, 148
  plotted points, 149
ANOM charts for means
  axis labels, 156
  central line, 147
  decision limit equations, 147, 148, 150
  examples, advanced, 157
  examples, introductory, 130
  missing values, 157
  notation, 146
  ODS tables, 152
  options summarized by function, 138, 146
  overview, 129
  plotted points, 146
  reading group summary statistics, 133–135, 155
  reading preestablished decision limits, 154, 155
  reading raw measurements, 130–132, 153, 154
  reading summary statistics and decision limits, 137, 156
  saving decision limits, 136, 150, 151
  saving group summary statistics, 135, 136, 151
  saving summary statistics and decision limits, 137, 152
  syntax, 137
ANOM charts for proportions
  central line, 95
  decision limit equations, 95
  decision limit parameters, 96
  examples, advanced, 103
  getting started, 78
  labeling axes, 102
  missing values, 103
  notation, 94
  ODS tables, 98
  options summarized by function, 86
  overview, 77
  plotted points, 95
  reading group data, 80–82, 101

reading group data and decision limits, 85, 101, 102
reading preestablished decision limits, 100
reading raw data, 78–80, 99
saving decision limits, 84, 85, 96, 97
saving group data, 83, 97
saving group data and decision limits, 84, 85, 98
syntax, 85
ANOM charts for rates
central line, 119
decision limit equations, 119, 120
decision limit parameters, 120
examples, introductory, 106
getting started, 106
labeling axes, 126
missing values, 127
notation, 119
ODS tables, 123
options summarized by function, 111
overview, 105
plotted points, 119
reading group data and decision limits, 126
reading number of nonconformities, 125
reading preestablished decision limits, 124, 125
reading raw data, 106–108, 123, 124
saving decision limits, 108–110, 121
saving group data and decision limits, 122, 123
saving number of nonconformities, 121, 122
syntax, 110
ANOM charts for Rates from Group Counts
examples, advanced, 127
augment, factorial design
example, 660, 664
autocorrelation in process data, 2144, 2145, 2148, 2150–2152
diagnosing and modeling, 2145, 2148
strategies for handling, 2148, 2150–2152
average and range charts, *see* X and R charts
average and standard deviation charts, *see* X and s charts
average charts, *see* X charts
average outgoing quality
AOQ2 function, 2217
Type B single-sampling, 2237, 2238
average run lengths
cusum schemes, 2224, 2225
EWMA scheme, 2228
average run lengths (cusum charts), *see* cumulative sum control charts
average sample number
ASN2 function, 2218
Type B single-sampling, 2238
average total inspection
ATI2 function, 2220
Type B single-sampling, 2238
axes, Pareto charts, 1096, 1111
axes, Shewhart charts, *see* Shewhart charts, axes

balanced incomplete block design, *see* block designs
balanced lattice, 682
Bayesian optimal designs, 1024, 1052
beta distribution
cdf plots, 263
chi-square goodness-of-fit test, 350
deviation from empirical distribution, 351
EDF goodness-of-fit test, 351
histograms, 314, 336
histograms, example, 363
P-P plots, 447
probability plots, 473
Q-Q plots, 502
block designs
balanced lattice, examples, 682
optimal designs, examples, 1008, 1054
randomized complete, examples, 666
block specification, FACTEX procedure
block pseudofactors, 630
block size restrictions, 632
number of blocks, 630
runs per block, 631
blocking, FACTEX procedure
block pseudofactor, 643
blocking factor, 643
example, 689
incomplete block design, example, 682
randomization, 649
rename block variable, 637
box charts
box appearance, options, 1998, 2001, 2015, 2029, 2057, 2058, 2064
box-and-whisker plots, description of, 1446
box-and-whisker plots, style of, 1998
control limit equations, 1447, 1448
control limits, specifying, 2003
displaying points, 1998
examples, advanced, 1461
examples, introductory, 1418
labeling axes, 1973
missing values, 1975
notation, 1445
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1434, 1445
outlier identification color, 2063
outlier identification symbol, 2064
overview, 1417
percentile computation, 1460, 2021
plotting character, 1433

reading preestablished control limits, 1431, 1432, 1454
reading raw measurements, 1418–1420, 1422, 1453, 1454
reading subgroup summary statistics, 1423, 1424, 1426, 1455, 1456
reading summary statistics and control limits, 1431, 1456, 1457
reading summary statistics and decision limits, 69
saving control limits, 1428, 1429, 1448, 1450
saving group summary statistics, 64
saving subgroup summary statistics, 1426–1428, 1450–1452
saving summary statistics and control limits, 1429–1431, 1452, 1453
schematic box-and-whisker plots, 1465
side-by-side box-and-whisker plots, 1417, 1448, 1465
skeletal box-and-whisker plots, 1464
standard deviation, estimating, 1459
syntax, 1432
tables, creating, 2038
$\bar{X}$ charts
standard deviation, estimating, 1459, 1460
box charts
capability indices, computing, 1449

c charts
central line, 1504
control limit equations, 1504, 1505
control limit parameters, 1505
examples, advanced, 1511
examples, introductory, 1483
getting started, 1483
known number of nonconformities, specifying, 1514, 1516
labeling axes, 1973
missing values, 1975
notation, 1503
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1494
overview, 1482
plotted points, 1503
plotting character, 1493
reading number of nonconformities, 1488–1491, 1509, 1510
reading preestablished control limits, 1487, 1488, 1509
reading raw data, 1483, 1484, 1508, 1509
reading subgroup data and control limits, 1510, 1511
saving control limits, 1484–1486, 1506
saving nonconformities per unit, 1491, 1492
saving number of nonconformities, 1506
saving subgroup data and control limits, 1507, 1508
syntax, 1492
tests for special causes, 1511–1513
candidate data set, OPTEX procedure, *see* optimal designs, candidate data set
capability indices
assumptions, 235
Boyles' index C_{pm}^{+}, 240
computing, 235–239
computing, example, 199
confidence interval, example, 253, 436
confidence limits, 205
$Cp_m(a)$, 210
estimation from Q-Q plots, 515, 521
estimation from Q-Q plots, example, 532
nonstandard indices, computing, 434
P_{pk} versus C_{pk}, 235
specialized, 239
specification limits, example, 199
specification limits, specifying, 216
terminology, 235
tests for normality, 204
the index k, 240
the index C_{jkp}, 240
the index C_{pc}, 245
the index C_{pg}, 244
the index C_{pk}^{W}, 244
the index C_{pm}^{W}, 245
the index C_{pp}, 243
the index $C_{pp}^{''}$, 243
the index C_{pq}, 244
the index C_{p}^{W}, 244
the index S_{jkp}, 243
the indices $C_{p(5.15)}$, 241
the indices $C_{pk(5.15)}$, 242
the indices $C_{pm}(a)$, 241
the indices C_{pmk}, 242
Vännmann's index $C_p(u, v)$, 246
Vännmann's index $C_p(v)$, 246
Wright's index C_s, 242
CAPABILITY procedure
introduction, 193
learning about, 194
plot statements, 194
cdf plots
axes, specifying, 269
beta distribution, 263
creating, 256
defining character features, 207, 264, 269
example, 256
exponential distribution, 264

gamma distribution, 264
generalized Pareto distribution, 267
getting started, 256
Gumbel distribution, 265
inverse Gaussian distribution, 265
legends, 266
lognormal distribution, 266
normal distribution, 267
normal distribution, example, 271
ODS graph name, 270
options summarized by function, 258, 260, 263
overview, 255
power function distribution, 268
Rayleigh distribution, 268
reference lines, example, 273
suppressing empirical cdf, 267
suppressing legend, 267
Weibull distribution, 269
center points, example, 663
chart description, Shewhart charts, 2062
chi-square goodness-of-fit test, 350
compared to EDF test, 372
classification variable, *see* comparative histograms
classification variables, OPTEX procedure, *see* optimal designs, model
classification variables, Pareto charts, 1119, 1129
clipping points, Shewhart charts, *see* Shewhart charts, clipping points
coding designs, *see* optimal designs, coding
coding, FACTEX procedure
block factor, 637
design factor, 637
coefficient of variation
computing, 226
collapsing factors, example, 673
coloring Pareto charts, *see* Pareto charts, coloring
coloring, Shewhart charts, *see* Shewhart charts, coloring
comparative histograms
bar labels, specifying, 284, 314
bar width, specifying, 292
bins, specifying, 289
bins, specifying midpoints of, 289
classification variable, missing values of, 289
classification variable, ordering levels of, 290, 291
classification variable, specifying, 284, 285
color, options, 292
getting started, 275
grids, 287
intervals, information about, 291
kernel density estimation, options, 284, 287
legend, 293
line type, grids, 292
normal distribution, example, 277
normal distribution, options, 289
ODS graph name, 294
one-way with inset statistics, example, 294
one-way, example, 276
options summarized by function, 280, 281, 283
overview, 274
specification limits, 285
specification limits, filled areas, 215–217
suppressing plot features, 289
two-way, example, 296
vertical scale, 291
computational form of the cusum chart, *see* cumulative sum control charts
confidence intervals, *see* intervals, CAPABILITY procedure
confidence levels, 204
confidence limits, 204–206
basic parameters, 205
confidence levels, 204
distribution-free, 206
for percentiles, 231
normally distributed, 206
percentiles, 206
probability of exceeding specifications, 206
process capability indices, 205
quantiles, 206
confidence limits, CAPABILITY procedure
confidence level, 205, 206, 211, 2002
type, 205, 206, 211, 2002
confounding rules
compare with alias structure, 653
design factors, 642
details, 653
example, 678
MaxClear designs, 655
minimum aberration, 654
notation, 653
orthogonally confounded, 644
partial confounding, example, 678
run-indexing factors, 642
searching, 644
split-plot designs, 655
syntax, 632, 633
unconfounded effects, 643
connecting points, Shewhart charts, 2070
constants
using functions to calculate, 2239
constants, control charts
A2, 2239
A3, 2239
B3, 2239
B4, 2239
B5, 2239
B6, 2239

D1, 2239
D2, 2239
D3, 2239
D4, 2239
E2, 2239
E3, 2239
c4, 2223
c5, 2239
d2, 2226
d3, 2227
constrained mixture designs, *see* mixture designs
contamination, variance
BAYESACT call, 2221
contribution plots, 908, 909, 967, 968
control chart functions
expected value of range, 2226
standard deviation of range, 2227
control factor design, 652
control factors, 652
control factors, example, 685
control limits, Shewhart charts, *see* Shewhart charts, control limits
correlated runs, designs with, *see* optimal designs, optimal blocking
covariance, optimal designs with, *see* optimal designs, optimal blocking
covariates, optimal designs with, *see* optimal designs, optimal blocking
Cramér-von Mises statistic, 229
Cramér-von Mises test, 209
Cramer-von Mises statistic, 352
cross validation
MVPMODEL procedure, 945
cumulative distribution, *see* cdf plots
cumulative percentage curve, *see* Pareto charts, cumulative percentage curve
cumulative sum control charts
annotating, 550
average run length approach, 590–592
central reference value, 591
color, options, 584
compared with Shewhart charts, 593
computational form, 561, 562, 564, 565
cusum schemes, specifying, 582
decision interval, defining, 588
designing a cusum scheme, 590–592
detecting shifts, 580, 582
economic design, 591
error probability approach, 591
examples, advanced, 602
examples, introductory, 555
FIR (fast initial response) feature, 586
graphics catalog, specifying, 551
headstart values, 580, 586
interpreting one-sided charts, 588
interpreting two-sided charts, 557, 590
introduction, 548
learning about, 549
line printer features, 551
line types, options, 584
line widths, options, 584
lineprinter plots, using, 552
lower cumulative sum, 586
missing values, 602
monitoring variability, example, 602–604
negative shifts, 586
nonstandardized data, 580
notation, 585
ODS tables, 599
one-sided (decision interval) schemes, 561, 562, 564, 565, 585
options summarized by function, 570, 579
origin, specifying, 581
overview, 554
plotting character, 570
positive shifts, 585
process mean, specifying, 581
process standard deviation, specifying, 583
reading cusum scheme parameters, 552, 567, 569, 600, 601
reading raw measurements, 550, 555–557, 600
reading subgroup summary statistics, 552, 558, 559, 601, 602
reference values, specifying, 580
saving cusum scheme parameters, 565, 566, 596, 597
saving subgroup summary statistics, 560, 561, 597
saving summary statistics and cusum parameters, 598
Shewhart charts, combined with, 607, 608
standard deviation, estimating, 583, 594–596
suppressing average run length calculation, 581
suppressing display of V-mask, 581
syntax, 550, 569
two-sided (V-mask) schemes, 587, 588
two-sided (V-mask) schemes, examples, 555–559
Type 1 error probabilities, 579, 583
Type 2 error probabilities, 579
upper and lower cumulative sum charts, combining, 605, 607
upper cumulative sum, 585
V-mask, defining, 588, 590
curvature, check for, example, 663
cusum charts, *see* cumulative sum control charts
cusum schemes
designing with CUSUMARL function, 2224, 2225

D-optimal designs, *see* optimal designs, optimality criteria
density estimation, *see* kernel density estimation
derived factors, FACTEX procedure
creating, 637
example, 672
descriptive statistics
computing, 224, 226
printing, example, 197
using PROC CAPABILITY, 197
design augmentation, 1007, 1021, 1049
design characteristics, FACTEX procedure
alias structure, 653
confounding rules, 653
design listing, 633
design size specification, FACTEX procedure
fraction, 640
minimum runs, 640
number of runs, 639
run indexing factors, 640
syntax, 639
design size specification, OPTEX procedure, 1020
design, factorial, *see* factorial designs
DETMAX algorithm, *see* optimal designs, search algorithms
distance from a point to a set, 1036
distance-based designs, *see* optimal designs, space-filling designs
double-sampling plans, *see* acceptance sampling

EDF, *see* empirical distribution function, *see* empirical distribution function
effect length, FACTEX procedure
limit, 630
effect length, OPTEX procedure
limit, 1013
empirical distribution function
definition of, 228, 351
EDF test compared to chi-square goodness-of-fit test, 372
EDF test statistics, 228, 351
EDF test statistics, Anderson-Darling, 229, 352
EDF test statistics, Cramér-von Mises, 229
EDF test statistics, Cramer-von Mises, 352
EDF test statistics, Kolmogorov-Smirnov, 228, 352
EDF test, probability values, 353
EWMA charts
asymptotic control limits, displaying, 817
asymptotic control limits, example, 837
average run lengths, computing, 846
axis labels, 834
central line, 821
control limit equations, 821
control limits, computing, 817, 821
displaying subgroup means, example, 844
examples, advanced, 835
examples, introductory, 796
missing values, 834
notation, 820
ODS tables, 828
options summarized by function, 808
overview, 795
plotted points, 820
plotting character, 808
plotting subgroup means, 818
probability limits, 817
process mean, specifying, 818
process standard deviation, specifying, 819
reading preestablished control limit parameters, 805–807, 829, 830
reading probability limits, 819
reading raw measurements, 796–798, 829
reading subgroup summary statistics, 799, 801, 802, 830, 831
reading summary statistics and control limits, 805, 831, 832
saving control limit parameters, 803, 825, 826
saving subgroup summary statistics, 802, 803, 826, 827
saving summary statistics and control limits, 804, 805, 827, 828
specifying parameters for, 835, 837
standard deviation, estimating, 832, 833
syntax, 807
varying subgroup sample sizes, 838
weight parameter, choosing, 822
weight parameter, specifying, 820
examine design, FACTEX procedure, *see* design characteristics, FACTEX procedure
examples, FACTEX procedure
advanced, 659
alias links breaking, 660
center points, 663
collapsing factors, 673
completely randomized, 659
derived factors, 672
design replication, 667, 670
fold-over design, 664
full factorial, 620
full factorial in blocks, 622
getting started, 620
half-fraction factorial, 624
hyper-Graeco-Latin square, 674
incomplete block design, 682
minimum aberration, 676
mixed-level, 670, 672
partial confounding, 678

point replication, 667, 670
pseudofactors, 672
randomized complete block design, 666
RCBD, 666
replication, 667, 670
resolution 3 design, 664
resolution 4, 676
resolution 4, augmented, 660
resolution III design, 664
resolution IV, 676
resolution IV, augmented, 660
sequential construction, 678
exchange algorithm, *see* optimal designs, search algorithms
expected value
for range of iid normal variables, 2226
for standard deviation of iid normal sample, 2223, 2224
exponential distribution
cdf plots, 264
chi-square goodness-of-fit test, 350
deviation from empirical distribution, 351
EDF goodness-of-fit test, 351
Goodness-of-fit tests, 1184
histograms, 317, 337
P-P plots, 448
probability plots, 475
Q-Q plots, 503, 504
exponentially weighted moving average charts, *see* EWMA charts
extreme observations, 910, 941, 965
extreme vertex designs, *see* mixture designs

FACTEX procedure
block specification, 630
block specification options, summary, 627
design factor levels, 634
design size options, summary, 627
design size specification, 639
design specification options, summary, 627
examining design characteristics, 632
factor specification options, summary, 627
features, 619
getting started examples, 620
invoking, 629
listing design factors, 634
model specification, 634
model specification options, summary, 627
output, 636
overview, 618
randomization, 639
replication, 638
resolution, 635
split-plot designs, 655
summary of functions, 627
syntax, 627
unit-effect specification, 640
units specification, 631
using interactively, 626
factorial designs, *see* examples, FACTEX procedure
balanced lattice, 682, 683
efficiency, 636
fractional factorial, MaxClear designs, 655
fractional factorial, minimum aberration, 654
fractional factorial, theory, 641
mixed-level, 637
orthogonal, 670
replicate, 638
resolution, 635
split-plot designs, 655
factors, FACTEX procedure
block factor, 643, 646
block pseudofactor, 643, 647, 653
derived factor, 646
design factor, 646
design factor coding, 637
design factor levels, 634
design factor names, 634
pseudofactor, 646
run-indexing factor, 642, 647, 653
types, 646
Fedorov algorithm, *see* optimal designs, search algorithms
filling area underneath density
histograms, 317
FIR (fast initial response) feature, *see* cumulative sum control charts
fold-over design, example, 664
folded normal distribution, histograms
example, 379
fonts, customizing, 2241–2243
fonts, hardware, 1722
fonts, Shewhart charts, 2062
fonts, TrueType, 1722
frequency data, Pareto charts, 1073, 1074
frequency tables, 208
full inspection and ASN2 function, 2218
functions
AOQ2, 2217, 2218, 2239
ASN2, 2218, 2219, 2239
ATI2, 2220, 2221, 2239
BAYESACT call, 2221–2223
C4, 2223, 2224, 2239
CUSUMARL, 2224, 2225
D2, 2226, 2239
D3, 2227, 2239
EWMAARL, 2228
for acceptance sampling, 2215

for control chart analysis, 2215
for sampling plans, 2216
PROBACC2, 2229, 2230, 2239
PROBBNML, 2230, 2231, 2237
PROBHYPR, 2232–2234, 2237
PROBMED, 2234, 2235
STDMED, 2235, 2236
summary of, 2215

G-optimal designs, *see* optimal designs, optimality criteria
gamma distribution
cdf plots, 264
chi-square goodness-of-fit test, 350
deviation from empirical distribution, 351
EDF goodness-of-fit test, 351
histograms, 318, 338
P-P plots, 449
probability plots, 475, 476
Q-Q plots, 504, 505
Generalized Pareto distribution
histograms, 341
generalized Pareto distribution
cdf plots, 267
P-P plots, 452
probability plots, 480
Q-Q plots, 508
geometric moving average charts, *see* EWMA charts
getting started, ANOM procedure
adding insets to plots, 168
getting started, CAPABILITY procedure
adding insets to plots, 386
creating histograms, 300
cumulative distribution plot, 256
distribution of variable across classes, 275
prediction, confidence, and tolerance intervals, 413
probability plot, 462
probability-probability plot, 440
quantile-quantile plot, 494
saving summary statistics, 424
summary statistics for process capability, 197
getting started, CUSUM procedure
adding insets to plots, 610
getting started, MACONTROL procedure
adding insets to plots, 893
getting started, SHEWHART procedure
adding insets to plots, 1976
Gini's mean difference, 209
GLM procedure, 700, 701
goodness-of-fit test, *see* empirical distribution function, *see* chi-square goodness-of-fit test, *see* empirical distribution function
Goodness-of-fit tests
Anderson-Darling statistic, 1185
Cramer-von Mises statistic, 1186
Kolmogorov-Smirnov statistic, 1185
probability values, 1186
Graeco-Latin square, 675
graphical output, Pareto charts, 1080
graphics
descriptions, 540
naming, 541
graphics catalog, specifying
CAPABILITY procedure, 208
grid options, Shewhart charts, 2005, 2062, 2064, 2065, 2069
Gumbel distribution
cdf plots, 265
histograms, 319, 339
P-P plots, 449
probability plots, 477
Q-Q plots, 505

hanging histograms, 320
HBAR charts
options summarized by function, 1081
syntax, 1081
headstart values in cusum schemes, 2224
histograms, *see* comparative histograms
adding summary statistics, 304
axis scaling, 333
bar width, 324
bar width, specifying, 334
bars, suppressing, 326
beta distribution, 314, 336
beta distribution, example, 363
capability indices, based on fitted distribution, 321
capability indices, based on fitted distribution, computing, 354
capability indices, based on fitted distribution, example, 374, 375
changing midpoints, example, 304
chi-square goodness-of-fit for fitted distribution, 350
color, options, 334
endpoints of intervals, 330
exponential distribution, 317, 337
filling area underneath density, 317
folded normal distribution, annotating, 379
gamma distribution, 318, 338
Generalized Pareto distribution, 341
getting started, 300
graphical enhancements, 360
grids, 319
Gumbel distribution, 319, 339
interval midpoints, 355

Inverse Gaussian distribution, 339
inverse Gaussian distribution, 320
Johnson S_B distribution, 330, 344
Johnson S_L distribution, 322
Johnson S_N distribution, 326
Johnson S_U distribution, 332, 345
kernel density estimation, 347
kernel density estimation, example, 375
kernel density estimation, options, 315, 321, 323, 333
legend, options, 327, 335
legends, suppressing, 326
line type, grids, 335
lognormal distribution, 322, 340
midpoints, 323, 324
multiple distributions, example, 367
normal distribution, 326, 340
normal distribution, example, 301
ODS tables, 359
options summarized by function, 306, 308, 311
output data sets, 328, 356, 358, 359
overview, 299
Pareto distribution, 328
percentile axis, 328
percentiles, 356
plots, suppressing, 326
Power Function distribution, 342
power function distribution, 329
printed output, 348, 350–352, 354, 355
printed output, capability indices based on fitted distribution, 354, 355
printed output, intervals, 355
printed output, suppressing, 325, 326
quantiles, 327, 356
Rayleigh distribution, 329, 343
saving curve parameters, 356
saving goodness-of-fit results, 356
S_B distribution, 330, 344
S_L distribution, 322
S_N distribution, 326
specification limits, color, 215, 216
specification limits, example, 300
specification limits, filled areas, 217
S_U distribution, 332, 345
symbols for curves, 336
three-parameter lognormal distribution, example, 377
three-parameter Weibull distribution, example, 379
Weibull distribution, 333, 346
hyper-Graeco-Latin square, example, 674

incomplete block design, *see* block designs
independent estimate of error, examples, 663, 667
individual measurement and moving range charts
central line, 1543
control limit equations, 1543
examples, advanced, 1551
examples, introductory, 1519
interpreting, 1550
labeling axes, 1973
missing values, 1975
moving range calculation, controlling, 1528
notation, 1542
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1530
overview, 1518
plotted points, 1543
plotting character, 1530
reading measurements, 1519–1521, 1547
reading measurements and ranges, 1522, 1523, 1548, 1549
reading measurements, ranges, and control limits, 1526, 1549, 1550
reading preestablished control limits, 1526, 1527, 1547, 1548
saving control limits, 1523, 1544, 1545
saving measurements and ranges, 1521, 1522, 1545
saving measurements, ranges, and control limits, 1524, 1525, 1546, 1547
standard deviation, estimating, 1550
standard values, specifying, 1554, 1556
syntax, 1529
tests for special causes, 1551, 1553, 1554
univariate plots, displaying, 1556–1558
individual measurement and moving range charts
capability indices, computing, 1545
information matrix, 1020
initialization for design search, *see* optimal designs, initialization
inner array, 652, 685
input data sets, Shewhart charts, *see* Shewhart charts, input data sets
insets
background color, 178, 403, 1089, 1987
background color of header, 178, 404, 1089, 1988
displaying summary statistics, example, 168, 386, 1077, 1976
drop shadow color, 178, 404, 1089, 1988
formatting values, example, 170, 387, 1157, 1978
frame color, 178, 404, 1089, 1988
getting started, 168, 386, 610, 893, 1976
goodness-of-fit statistics, example, 410
header text color, 178, 404, 1089, 1988
header text, specifying, 172, 177, 389, 403, 1088, 1159, 1980, 1986

labels, example, 170, 387, 1157, 1978
legend, example, 411
overview, 168, 385, 610, 892, 1975
positioning, details, 179–181, 183, 405–409, 1121, 1123, 1125, 1126, 1988–1992
positioning, example, 172, 389, 1159, 1980
positioning, options, 176, 177, 403, 1088, 1090, 1986, 1987
statistics associated with distributions, 396, 397, 399, 400
summary statistics grouped by function, 174, 392, 396, 1086, 1983
suppressing frame, 177, 403, 1088, 1987
text color, 178, 404, 1090, 1988
interaction, FACTEX procedure
alias structure, 653
between control and noise factors, 688
confounding, 642
examples, 678, 699, 700
generalized, 642, 644, 670
minimum aberration, 654
minimum aberration, example, 676
nonnegligible, 642
resolution, 648
specify terms, 634, 647
interquartile range, 209
intervals
ODS tables, 423
intervals, CAPABILITY procedure
computing for process capability analysis, 417
computing intervals, example, 413
confidence levels, specifying, 418
confidence, for mean, 418, 422
confidence, for standard deviation, 418, 422
intervals, CAPABILITY procedure, 418, 419
list of options, 417
notation used in computing, 420
number of future observations, 418
one-sided limits, example, 416
prediction, for future observations, 418, 420
prediction, for mean, 418, 421
prediction, for standard deviation, 418, 422
saving information, output data set, 419, 423
specifying method used, 418
specifying type of, 419
suppressing output tables, 419
tolerance, 421
tolerance, for proportion of population, 418
tolerance, specifying proportion of population, 419
Inverse Gaussian distribution
histograms, 339
inverse Gaussian distribution
cdf plots, 265
histograms, 320
P-P plots, 450
Ishikawa diagrams
adding arrows, 720–724
aligning arrows, 741–748
arrow colors, 764, 766–772
arrow heads, 774
arrow line style, 764, 766–772
arrow line width, 764, 766–772
balancing arrows, 741–748
box color, modifying, 764
box shadow, 774
clipboard graphics, 761, 762
color, arrow, 764, 766–772
color, box, 764
color, palette, 764, 766–772
color, text, 772
context-sensitive operations, 706, 717
data collection, 748, 749
data presentation, 748, 749
deleting arrows, 733–736
detail, decreasing, 750–752
detail, increasing, 750–752
Edit menu, 719
editing existing diagrams, 777, 778
editing labels, 724–727
examples, 784
examples, Integrated Circuit Failures, 785
examples, Photo Development Process, 786
examples, Quality of Air Travel Service, 784
exporting diagrams, 761, 762
File menu, 718
fonts, modifying, 763
Help menu, 720
highlighting arrows, 764, 766–772
history, 704
hotspots, 706, 717
isolating arrows, 754, 755
labeling arrows, 724–727
line palette, 764, 766–772
managing complexity, 750–758
merging diagrams, 755–758
mouse sensitivity, 774
moving arrows, 727–733, 738–748
multiple diagrams, displaying, 755–758, 778, 779
notepads, 748, 749
output, bitmaps, 761, 762
output, graphics, 758–760
output, SAS data set, 776, 782, 783
overview, 704
palettes, colors, 764, 766–772
palettes, fonts, 763
palettes, lines, 764, 766–772
printing, bitmaps, 761, 762

printing, SAS/GRAPH output, 758–760
resizing arrows, 736–738
SAS data set, input, 777, 778, 782, 783
SAS data set, output, 776, 782, 783
saving, bitmaps, 761, 762
saving, clipboard graphics, 761, 762
saving, graphics, 758–760
saving, SAS data set, 776
subsetting arrows, 736–738, 764, 766–772
summary of operations, 717–720
swapping arrows, 738–741
syntax, 716
tagging arrows, 736–738, 764, 766–772
terminology, 705
text entry, 724–727
tutorial, 707, 708, 710–715
undo, 733–736
View menu, 719
zooming arrows, 752–754, 774

Johnson S_B distribution
histograms, 330, 344
Johnson S_L distribution
histograms, 322
Johnson S_N distribution
histograms, 326
Johnson S_U distribution
histograms, 332, 345

k-exchange algorithm, *see* optimal designs, search algorithms
kernel, *see* kernel density estimation
kernel density estimation, 347
adding density curve to histogram, 321
area underneath density curve, 287, 317
bandwidth parameter, specifying, 284, 315
example, 375
filling area under density curve, 287, 317
kernel function, specifying type of, 287, 321
line type for density curve, 541
lower bound, specifying, 323
options used with, 288, 322
upper bound, specifying, 333
kernel function, *see* kernel density estimation
Kolmogorov-Smirnov statistic, 228, 352
Kolmogorov-Smirnov test, 209
kurtosis
computing, 226
saving in output data set, 427

labeling central line, Shewhart charts, *see* Shewhart charts, labeling central line
labeling Shewhart charts, *see* Shewhart charts, labeling
line types, Shewhart charts, *see* Shewhart charts, line types
location parameter
probability plots, 487
Q-Q plots, 519
lognormal distribution
cdf plots, 266
chi-square goodness-of-fit test, 350
deviation from empirical distribution, 351
EDF goodness-of-fit test, 351
histograms, 322, 340, 377
P-P plots, 450, 451
probability plots, 477
Q-Q plots, 505, 506

main effect, 642, 643, 647, 648
main effect, examples, 678–681, 699, 700
MaxClear designs, 655
maximum value
saving in output data set, 427
mean
saving in output data set, 427
mean and range charts, *see* X and R charts
mean and standard deviation charts, *see* X and s charts
mean charts, *see* X charts
measures of location
mode, 234
median
probability function for, 2234
saving in output data set, 427
standard deviation of, 2235
median absolute deviation about the median, 209
median and *R* charts
axis labels, 1643
central line, 1629
control limit equations, 1630
examples, advanced, 1638
examples, introductory, 1604
labeling axes, 1973
missing values, 1975
notation, 1628
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1617
overview, 1603
plotted points, 1629
plotting character, 1617
reading preestablished control limits, 1613, 1614, 1634, 1635
reading raw measurements, 1604, 1605, 1634
reading subgroup summary statistics, 1606–1609, 1635, 1636
reading summary statistics and control limits, 1613, 1636, 1637
saving control limits, 1610, 1611, 1631, 1632

saving subgroup summary statistics, 1609, 1610, 1632
saving summary statistics and control limits, 1611, 1613, 1633, 1634
standard deviation, estimating, 1637, 1638
syntax, 1615
median and range charts, *see* median and R charts
median charts
central line, 1586
control limit equations, 1586
controlling value of central line, 1595
examples, advanced, 1595
examples, introductory, 1560
labeling axes, 1973
missing values, 1975
notation, 1585
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1574
overview, 1559
plotted points, 1586
plotting character, 1574
reading preestablished control limits, 1571, 1572, 1591, 1592
reading raw measurements, 1560–1562, 1591
reading subgroup summary statistics, 1563, 1565, 1566, 1592, 1593
reading summary statistics and control limits, 1570, 1571, 1593, 1594
saving control limits, 1569, 1587–1589
saving subgroup summary statistics, 1566–1568, 1589
saving summary statistics and control limits, 1569–1571, 1589, 1590
standard deviation, estimating, 1594
syntax, 1573
median charts
capability indices, computing, 1588
minimum aberration
aberration vector, 654
blocked design, 655
example, 676
limitation, 678
minimum aberration, 654
minimum value
saving in output data set, 427
missing values
CAPABILITY procedure, 246
CUSUM procedure, 602
MACONTROL procedure, 834
MVPMODEL procedure, 946
output data set, 427
SHEWHART procedure, 1975
mixed-level, factorial design
construction, examples, 670–674
derived factors, 637
mixture designs
examples, 1009, 1062
plotting, 1063, 1065
mixture-process designs, *see* mixture designs
mode
saving in output data set, 427
model specification, FACTEX procedure
directly, 634
estimated effects, 634
indirectly, 634
maximum clarity, 635
minimum aberration, 636
nonnegligible effects, 634
resolution, 635
resolution, maximum, 635
specifying effects, 647
modes, 208
modified Fedorov algorithm, *see* optimal designs, search algorithms
moving average control charts, *see* EWMA charts, *see* uniformly weighted moving average charts
adding features to, 791
average run lengths, displaying, 891
graphics catalog, specifying, 792
introduction, 788
learning about, 790
line printer features, 791, 792
lineprinter charts, creating, 793
reading control limit parameters, 793
reading raw measurements, 791
reading subgroup summary statistics, 792, 793
syntax, 791
moving range charts, *see* individual measurement and moving range charts
multi-vari charts
examples using the SHEWHART procedure, 1477
multivariate control charts, 2177–2182
chart statistic, calculating, 2177
principal component contributions, 2181
mutually orthogonal Latin square, 675, 682
MVPDIAGNOSE procedure
examples, 917
extreme observations, 910
missing values, 906
ODS graph names, 917
MVPMODEL procedure
centering, 946
concepts, 942
cross validation, 945
examples, 949
extreme observations, 941
missing values, 937

ODS graph names, 948
ODS table names, 948
output data sets, 947
scaling, 946
specifying analysis variables, 941
test set validation, 945
MVPMONITOR procedure
examples, 980
extreme observations, 965
missing values, 963
ODS graph names, 980

neighbor-balanced designs, 1061
Newton-Raphson approximation
gamma shape parameter, 534
Weibull shape parameter, 534
noise factors, 652, 685
nonconforming items
probability of choosing, 2230–2234
nonnormal process data, 2171, 2173–2176
calculating probability limits, 2174
preliminary chart, 2173
normal distribution
cdf plots, 267
cdf plots, example, 271
chi-square goodness-of-fit test, 350
comparative histograms, 289
comparative histograms, example, 277
deviation from empirical distribution, 228, 351
EDF goodness-of-fit test, 228, 351
histograms, 325, 326, 340
histograms, example, 301
P-P plots, 451
P-P plots, example, 440
probability plots, 478
Q-Q plots, 507
normal random variables
expected value of standard deviation, 2224
standard deviation of range, 2226
normality tests, 209, 227
Anderson-Darling test, 209
changes made to, 227
Cramér-von Mises test, 209
Kolmogorov-Smirnov test, 209
Shapiro-Wilk test, 209
np charts
central line, 1668
control limit equations, 1669
control limit parameters, 1669
control limits, specifying, 1684–1686
examples, advanced, 1676
getting started, 1647
labeling axes, 1973
missing values, 1975
notation, 1667
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1658
overview, 1646
plotted points, 1668
plotting character, 1658
reading preestablished control limits, 1655, 1656, 1673, 1684–1686
reading raw data, 1647–1649, 1672, 1673
reading subgroup data, 1649–1651, 1674
reading subgroup data and control limits, 1653, 1654, 1675
saving control limits, 1652, 1653, 1669, 1670
saving subgroup data, 1651, 1652, 1670, 1671
saving subgroup data and control limits, 1653, 1671, 1672
standard average proportion, specifying, 1678, 1680
syntax, 1656
tests for special causes, 1676, 1677
unequal subgroup sample sizes, 1680–1683
null hypothesis
location parameter, 208

observation exclusion, 207
OC Curve, 1727, 1879
ODS (Output Delivery System)
MVPMODEL procedure table names, 948
RAREEVENTS procedure table names, 1190
ODS tables
CAPABILITY procedure, 247
FACTEX procedure, 659
OPTEX procedure, 1042
RELIABILITY procedure, 1388, 1389
one-way comparative Pareto charts, *see* Pareto charts, comparative
Operating Characteristic Curve, 1727, 1879
optimal blocking, *see* optimal designs, optimal blocking
optimal designs
A-efficiency, 1031
Bayesian optimal designs, 1024, 1052
covariate designs, 1010, 1024
customizing design search, 1020
D-efficiency, 1031
data set roles, 1026, 1027
design augmentation, 1007, 1021, 1049
design augmentation data set, 1026, 1027
design listing, 1019
design search defaults, 1020
efficiency measures, 1031
efficiency measures, comparing, 1042, 1043, 1045
efficiency measures, interpreting, 1032

epsilon value, 1013
evaluating an existing design, 1022, 1038, 1040, 1059
examining, 1019, 1020
G-efficiency, 1031
getting started examples, 1001
including identification variables, 1024, 1027, 1028
information matrix, 1020
input data sets, 1026
interactively, 1020, 1043
invoking, 1012
learning about the OPTEX procedure, 1000
memory usage, 1036
mixture designs, 1062
number of design points, 1020, 1023
number of search tries, 1020, 1022
number of tries to keep, 1023
OPTEX procedure features, 999
OPTEX procedure overview, 999
optimal blocking, 1039
output, 1041
output data set, 1028
prior precision values, 1024, 1053
random number seed, 1013
resolution 4 designs, 1052
run-time considerations, 1036
saturated design, 1007, 1023
search methods, 1037
search strategies, 1040
statement descriptions, 1012
status of search, 1014
summary of functions, 1010
syntax, 1010
treatment candidate points, 1059
variance matrix, 1020
optimal designs, candidate data set
creating with DATA step, 1008, 1009, 1042
creating with FACTEX procedure, 1007, 1008
creating with PLAN procedure, 1001, 1002, 1046
discussion, 1026, 1027
examples of creating, advanced, 1042
examples of creating, introductory, 1001
recommendations, 1040, 1050
specifying, 1013
optimal designs, coding
default coding, 1032
discussion, 1032
examples, 1033
no coding, 1034
orthogonal coding, 1033, 1057–1059
recommendations, 1033
specifying, 1013
static coding, 1032
optimal designs, examples
advanced, 1042
Bayesian optimal designs, 1052
block design, 1008, 1054
design augmentation, 1007, 1049
designs with correlated runs, 1060
designs with covariates, 1057
handling many variables, 1008
initialization, 1047
introductory, 1001
mixture design, 1009, 1062
nonstandard modeling, 1042
reducing candidate set, 1050
resolution 4 design, 1052
saturated second-order design, 1007
using different search methods, 1045
optimal designs, initialization
defaults, 1020–1022
example, 1047
initial design data set, 1022, 1026, 1027, 1048
optimal blocking, 1015
partially random, 1022
random, 1022
recommendations, 1040
sequential, 1021
specifying, 1021
optimal designs, model
abbreviation operators, 1030
classification variables, 1015, 1029
crossed effects, 1030
discussion, 1029
examples, 1031
factorial model, 1031
interactions, 1030
main effects, 1029
main effects model, 1031
no-intercept model, 1024
nonstandard, 1042
polynomial effects, 1029
quadratic model, 1031
regressor effects, 1029
specifying, 1024
types of effects, 1024, 1029
types of variables, 1029
optimal designs, optimal blocking
A-efficiency, 1032
block specification, 1014
classification variables, 1015
covariance specification, 1014
covariate designs, 1057
D-efficiency, 1032
data sets, 1028
discussion, 1039
evaluating an existing design, 1040

examples, 1054, 1057, 1060
initialization, 1015
number of search tries, 1015
specifying, 1011, 1014
suppressing exchange step, 1015
treatment candidate points, 1014, 1059
tries to keep, 1015
optimal designs, optimality criteria
A-optimality, 1021, 1035, 1045
computational limitations, 1036
D-optimality, 1021, 1034
default, 1020
definitions, 1034–1036
discussion, 1034
distance-based, 1034, 1036
examples, 1043, 1062
G-optimality, 1025, 1035
I-optimality, 1035
information-based, 1034
S-optimality, 1021, 1036
specifying, 1021
types, 1034
U-optimality, 1021, 1036, 1062
optimal designs, output
block variable name, 1025
design number, 1025
options, 1025
output data set, 1025, 1028
selecting design by efficiency, 1025, 1035
transfer variables, 1024
optimal designs, search algorithms
comparing different algorithms, 1045, 1047
default, 1020
DETMAX, 1023, 1039, 1045
discussion, 1037
example, 1045, 1047
exchange, 1023, 1038
excursion level for DETMAX, 1023
FEDOROV, 1047
Fedorov, 1023, 1039
k-exchange, 1023
modified Fedorov, 1023, 1039
rank-one updates, 1037
sequential, 1023, 1038, 1045, 1047
specifying, 1023
speed, 1023, 1037, 1045, 1046
optimal designs, space-filling designs
coding for, 1034
criteria, 1034
definitions, 1036
distance from a point to a set, 1036
efficiency measures, 1032
examples, 1062
S-optimality, 1036
specifying, 1021
U-optimality, 1036
options summary
ESTIMATE statement, 1282
options, ANOM charts
dictionary, 183
options, Shewhart charts
dictionary, 1993
orthogonal confounding, 646, 647
orthogonal design
theory, 641
outer array, 652, 685
outgoing quality, *see* AOQ2 function
output data set, Pareto charts, 1126, 1127
output data sets, CAPABILITY procedure
creating, 433
getting started, 424
naming, 427
percentile variable names, 432
percentiles, 433
saving summary statistics, 427
output data sets, Shewhart charts, *see* Shewhart charts, output data sets
output, FACTEX procedure
code design factor levels, 637
decode block factor levels, 637
decode design factor levels, 637
details, 658
options, 637
output data set, 636, 658
rename block variable, 637

p charts
central line, 1709
control limit equations, 1710
control limit parameters, 1710
control limits, revising, 1724, 1726, 1727
examples, advanced, 1716
getting started, 1687
labeling axes, 1973
missing values, 1975
notation, 1708
OC curves, 1727–1729
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1699
overview, 1686
plotted points, 1709
plotting character, 1699
reading preestablished control limits, 1696, 1697, 1714
reading raw data, 1687, 1688, 1690, 1713
reading subgroup data, 1690–1692, 1715

reading subgroup data and control limits, 1695, 1715, 1716
saving control limits, 1693, 1694, 1696, 1711
saving subgroup data, 1693, 1711, 1712
saving subgroup data and control limits, 1694, 1695, 1712, 1713
standard average proportion, specifying, 1719, 1721
syntax, 1697
tests for special causes, 1717, 1718
unequal subgroup sample sizes, 1721, 1723
P-P plots
beta distribution, 447
compared to Q-Q plots, 458
distribution options, 443, 445, 459
distribution reference line, 442, 444
exponential distribution, 448
gamma distribution, 449
generalized Pareto distribution, 452
getting started, 440
graphics, options, 460
Gumbel distribution, 449
interpreting, 455
inverse Gaussian distribution, 450
line printer, options, 453
line width, distribution reference line, 460
lognormal distribution, 450, 451
normal distribution, 451
normal distribution, example, 440
options summarized by function, 443, 445
overview, 439
power function distribution, 452
Rayleigh distribution, 453
Weibull distribution, 454
Pareto charts
"trivial many", 1068, 1141
"useful many", 1068, 1141
"vital few", 1068, 1141
avoiding clutter, 1128
axes, 1096, 1102, 1111
before-and-after, 1130–1132, 1134
classification variables, 1119, 1129
examples, advanced, 1130
examples, introductory, 1069
graphics catalog, 1080
grids, 1102
highlighting, 1141–1145
labeling chart features, 1120
large data sets, 1129
levels, 1117
many categories, 1160
merging columns, example, 1150
missing values, 1104, 1105, 1129
options summarized by function, 1079
output data set, 1126, 1127
overview, 1068
Pareto curve, 1071
Pareto, Vilfredo, 1068
process variables, 1070, 1117, 1129
reading frequency data, 1073, 1074
reading raw data, 1069, 1070, 1072
reference lines, 1113
restricting number of categories, 1074, 1077
saving information, 1126, 1127
scaling bars, 1109, 1128
seven basic QC tools, 1068
side-by-side, 1068
stacked, 1068
syntax, 1078
tied categories, 1074, 1077
using raw data, example, 1069, 1070, 1072
vertical axis, 1118
visual clarity, 1128
Pareto charts, alternative
example, 1154
Pareto charts, categories, 1071, 1117
legend, 1072
maximum number of, 1129
restricting number of, 1074, 1077, 1103, 1104
ties, 1074, 1077
unbalanced, 1119
Pareto charts, classification variables
examples, 1130, 1134
Pareto charts, coloring
axes, 1111
bar outlines, 1112
bars, 1112
cumulative percentage axis, 1111
cumulative percentage curve, 1112
grid lines, 1113
highest bars, 1099
labels, 1112
lowest bars, 1100
recommendations, 1128
reference lines, 1112, 1113
tick marks, 1111
tiles, 1113
Pareto charts, comparative, 1068, 1119
cells, 1119
classification variables, 1132
classification variables, examples, 1130, 1134
creating, 1099
frequency proportion bars, 1101
key cell, 1100, 1119, 1133, 1140
merging columns, 1150
one-way, 1119
one-way, example, 1138
ordering values, 1108

rows and columns, ordering, 1108
tiles, 1119, 1144
two-way, 1119
two-way, examples, 1134, 1139, 1141, 1142, 1144, 1147, 1150
unbalanced categories, 1108, 1119
weighted charts, 1152
Pareto charts, cumulative percentage curve, 1071, 1106, 1118
anchoring, 1136, 1137
coloring, 1112
enhancing, 1091
scaling, 1120
suppressing, 1128, 1138, 1139
Pareto charts, grid lines
width, 1117
Pareto charts, legends
bar legends, 1097
category legend labels, 1098
highest and lowest bars legend labels, 1102
sample size legends, 1098, 1105
tile legends, 1116
Pareto charts, other category, 1108, 1109
coloring, 1113
labeling, 1103
pattern, 1116
Pareto charts, *other* category, 1074, 1077
Pareto charts, restricted, 1074, 1077, 1104, 1118, 1129
large data sets, 1129
Pareto charts, weighted, 1118
example, 1152
Pareto curve, 1071
Pareto distribution
histograms, 328
Pareto principle, 1068
Pareto, Vilfredo, 1068
partial confounding, example, 678
pattern tests, *see* Shewhart charts, tests for special causes
percent plots, *see* P-P plots
percentiles
axes, Q-Q plots, 508, 510, 521
confidence limits, 231
defining, 209, 230
empirical distribution function, 230
saving in output data set, 433
visual estimates, Q-Q plots, 521
weighted, 230
weighted average, 230
PLAN procedure, 684
plot statements, CAPABILITY procedure, 194
plots
axis color, 539
color, options, 535, 539, 540
comparative, 536
line type, 541
reference lines, options, 535, 538–542
tick marks on horizontal axis, 541
Power Function distribution
histograms, 342
power function distribution
cdf plots, 268
histograms, 329
P-P plots, 452
probability plots, 480
Q-Q plots, 511
prediction intervals, *see* intervals, CAPABILITY procedure
prediction, k-values for
prediction, *k*-values for, 418
probability functions
binomial, 2230, 2231
for median, 2234, 2235
hypergeometric, 2232–2234
probability limits, Shewhart charts, 1995, 2023, 2030
probability of exceeding specifications, 206
probability plots
axes, rotating, 481
beta distribution, 473, 474
distribution reference lines, 482, 488
distribution reference lines, examples, 490–492
distributions, 486
exponential distribution, 475
gamma distribution, 475
generalized Pareto distribution, 479, 480
getting started, 462
graphics, options, 488
Gumbel distribution, 476, 477
legends, 485
legends, suppressing, 478
line printer, options, 486
location parameter, 487
lognormal distribution, 477
lognormal distribution, example, 464
normal distribution, 468, 478
normal distribution, example, 463
options summarized by function, 469–471
overview, 461
percentile axis, 480
power function distribution, 480
Rayleigh distribution, 481
reference lines, 485
scale parameter, 487
shape parameter, 487
syntax, 468
threshold parameter, 483, 487
Weibull distribution, 483–485
probability-probability plots, *see* P-P plots

PROC CAPABILITY statement, 195
process capability indices
  confidence limits, 205
process distribution, *see* empirical distribution function
process potential
  P_{pk} versus C_{pk}, 235
process variables, Pareto charts, 1070, 1117, 1129
pseudofactors, example, 672

Q-Q plots
  axes, percentile scale, 508, 510, 521
  axes, rotating, 512
  beta distribution, 499, 502
  capability indices, 507, 515, 521, 532
  creating, 516
  diagnostics, 517
  distribution reference lines, 495, 520
  distributions, 498, 518
  estimating C_{pk}, 532
  exponential distribution, 499, 503, 504
  gamma distribution, 499, 504
  generalized Pareto distribution, 508, 511
  getting started, 494
  graphics, options, 521
  Gumbel distribution, 505
  interpretation, 517
  legends, 505
  legends, suppressing, 496, 507, 508, 515
  line printer, options, 516
  line width, 521
  location parameter, 519
  lognormal distribution, 499, 505, 506
  lognormal distribution, example, 525
  nonnormal data, example, 523
  normal distribution, 499, 507
  normal distribution, example, 494, 532
  options summarized by function, 498–500, 502
  overview, 493
  percentiles, estimates, 521
  power function distribution, 511
  Rayleigh distribution, 511, 512
  reference lines, 499, 508, 521
  sample estimates, 507
  scale parameter, 519
  syntax, 497
  threshold parameter, 519
  Weibull distribution, 499, 513–515
  Weibull distribution, example, 530
quantile-quantile plots, *see* Q-Q plots
quantiles
  defining, 230
  empirical distribution function, 230
  weighted average, 230

R charts
  capability indices, computing, 1755
  central line, 1753
  control limit equations, 1753, 1754
  control limits, specifying, 1763–1765
  examples, advanced, 1761
  examples, introductory, 1730
  labeling axes, 1973
  missing values, 1975
  notation, 1752
  ODS graph names, 1968
  ODS tables, 1966
  options summarized by function, 1743
  overview, 1729
  plotted points, 1753
  plotting character, 1743
  probability limits, 1761, 1762
  reading preestablished control limits, 1740, 1741, 1758
  reading raw measurements, 1731–1733, 1757
  reading subgroup summary statistics, 1733, 1734, 1736, 1758, 1759
  reading summary statistics and control limits, 1739, 1759, 1760
  saving control limits, 1737, 1738, 1754, 1755
  saving subgroup summary statistics, 1736, 1737, 1755, 1756
  saving summary statistics and control limits, 1738, 1739, 1756, 1757
  standard deviation, estimating, 1760, 1761
  syntax, 1742
randomization, FACTEX procedure
  blocking, 649
  details, 649
  example, 659, 666
  prevent, 639, 650
  seed, 639, 666
randomized complete block, example, 666
randomized treatments, example, 666
range
  saving in output data set, 427
range charts, *see R* charts
RAREEVENTS procedure
  constructing charts, 1183
  examples, 1191
  input data sets, 1187
  ODS graph names, 1191
  ODS table names, 1190
  output data sets, 1189
Rayleigh distribution
  cdf plots, 268
  histograms, 329, 343
  P-P plots, 453
  probability plots, 481
  Q-Q plots, 512

reference lines, Shewhart charts, *see* Shewhart charts, reference lines
reliability analysis
analyzing accelerated life test data, 1214–1219
analyzing arbitrarily censored data, 1224
analyzing binomial data, 1260, 1261
analyzing combined failure modes, 1241
analyzing groups of data, 1210, 1211, 1214
analyzing interval-censored data, 1219, 1221, 1222, 1227
analyzing regression models, 1229, 1231, 1232, 1234
analyzing repair data, 1245, 1247, 1250
analyzing right-censored data, 1206, 1207, 1210
analyzing two groups of repair data, 1250, 1252, 1253, 1257
arbitrarily censored data, 1348
binomial parameter estimation, 1365, 1366
classification variables, 1279
comparison of groups of recurrence data, 1381
confidence intervals for parameters, 1360, 1361
covariance matrix of parameters, 1359
creating life-stress relation plots, 1325, 1326, 1329, 1332, 1335, 1337, 1338
creating output data sets, 1291, 1389
creating probability plots, 1311, 1312, 1317, 1319, 1321, 1324
details, 1340
Duane plots, 1387
estimating distribution parameters, 1273, 1275, 1278
examples, 1206
failure modes, 1241, 1283, 1369
fitting regression models, 1302, 1304, 1308
frequency variables, 1284
insets, 1284, 1285, 1287
least squares estimation, 1368
log-scale regression model parameters, 1358
maximum likelihood estimation, 1354
mean cumulative function plots, 1291, 1293, 1296, 1298, 1302, 1339
observation-wise percentiles, 1371, 1372
observation-wise predicted values, 1370
observation-wise predicted values, recurrent events, 1375
observation-wise reliability function estimates, 1373
observation-wise statistics, 1370–1374
observation-wise statistics, recurrent events, 1375
ODS Graphics, 1390
optimization options recurrent events, 1310
optimization options three-parameter Weibull, 1310
overview, 1204, 1205
parameter estimation, 1354, 1357–1369
parametric model for recurrent events data, 1266, 1268
parametric model for two groups of repair data, 1257
percentile estimation, 1362, 1363
Poisson parameter estimation, 1367, 1368
predicted values for recurrent events data, 1257
probability distributions, 1340–1342
probability plots, 1343–1345, 1347
readout data, 1310
recurrence data, 1376, 1377, 1381
recurrent events data, 1266, 1268
regression model parameters, 1357, 1358
reliability function estimation, 1364, 1365
residuals, 1373, 1374
specifying failure modes, 1283
specifying probability distributions, 1280
syntax, 1271
three-parameter Weibull, 1263
Turnbull algorithm, 1348
types of lifetime data, 1340
Weibayes estimation, 1368
replication, FACTEX procedure
data set, 638, 639
design point, 639
design replication, 650–652
details, 650
entire design, 638
example, 667, 670
fixed number of times, 650
inner array, 652
number of times, 638, 639
outer array, 652
point replication, 650–652
resolution, FACTEX procedure
comparison, 648
definition, 648
example, 624, 660, 676
MaxClear designs, 655
minimum aberration, 654
number, 648
numbering scheme, 649
syntax, 635
response, factorial design, 646, 700
restricted Pareto charts, *see* Pareto charts, restricted
robust estimators
location, 232
scale, 207, 232
trimmed means, 232
Winsorized means, 232
robust measures of scale, 209
Q_n, 209
S_n, 209

rounding, 209
rules for lack of control, *see* Shewhart charts, tests for special causes
runs rules, *see* Shewhart charts, tests for special causes
runs tests, *see* Shewhart charts, tests for special causes

s charts
  central line, 1789
  control limit equations, 1790
  examples, advanced, 1798
  examples, introductory, 1768
  notation, 1789
  ODS graph names, 1968
  ODS tables, 1966
  options summarized by function, 1779
  overview, 1767
  plotted points, 1789
  plotting character, 1779
  reading preestablished control limits, 1776–1778, 1794, 1795
  reading raw measurements, 1768–1770, 1794
  reading subgroup summary statistics, 1771, 1773, 1795, 1796
  reading summary statistics and control limits, 1776, 1796, 1797
  saving control limits, 1774, 1775, 1790, 1792
  saving subgroup summary statistics, 1773, 1774, 1792
  saving summary statistics and control limits, 1775, 1776, 1793
  standard deviation, estimating, 1797, 1798
  standard deviation, specifying, 1798, 1799
  syntax, 1778
s charts
  capability indices, computing, 1792
  labeling axes, 1973
  missing values, 1975
S-optimal designs, *see* optimal designs, space-filling designs
sampling plans, *see also* acceptance sampling
  double, 2239
  single, 2237, 2238
  types of, 2237
saturated designs, analysis of, 2222
saturated designs, OPTEX procedure, 1007, 1023
S_B distribution
  histograms, 330, 344
scale parameter
  probability plots, 487
  Q-Q plots, 519
score plots, 910, 911
search design, FACTEX procedure
  confounding rules, 644
  limit, 630
  maximum time, 630
  speeding, 645
semicurtailed inspection and ASN2 function, 2218
sequential algorithm, *see* optimal designs, search algorithms
seven basic QC tools, 1068
shape parameter
  probability plots, 487
  Q-Q plots, 519
Shapiro-Wilk test, 209
Shewhart charts
  subgroup-variables, 1970, 1971
  annotating, 2056
  average run lengths, example, 1800
  between-subgroup variance, 2157
  capability indices, computing, 1971, 1973
  challenging assumptions of, 2143
  chart description, 2062
  chart naming, 2066
  computing capability indices, 2011, 2038, 2045
  connecting points, 2012, 2070
  control chart statistics, 2011
  details, 1966
  displaying points, 1995
  estimating μ, 2011
  estimating σ, 2031, 2036
  exceptions charts, 2005, 2037
  fonts, 2062
  fonts, hardware, 1722
  fonts, TrueType, 1722
  grids, 2005, 2062, 2064, 2065, 2069
  horizontal axes, 2016
  identifying unequal subgroup sample sizes, 2013
  intervals between subgroups, 2008
  missing values, 1975
  options dictionary, 1993
  plot margins, 2011, 2029
  probability limits, 1995, 2023, 2030
  separating, 2029
  separating subgroups, 2021
  subgroup sample size, 2037
  subgroups, 2012
  vertical axes, 2048
Shewhart charts, axes
  appearance, 2056, 2069
  coloring, 2057
  for multiple pages, 2028
  horizontal, 2005, 2046
  labeling, 1643, 1645, 2109–2113
  offset length, 2006
  scaling on *p* charts, 2050
  scaling primary and secondary charts, 2050
  suppressing labels, 2014, 2067
  tick mark labels, 2030, 2068

tick marks, 2046, 2063, 2069
vertical axis truncation, 2016
Shewhart charts, box charts, *see* box charts
Shewhart charts, clipping points, 2002, 2010, 2058, 2059, 2070
examples, 2105–2108
Shewhart charts, coloring
axes, 2057
axis labels, 2062
connecting lines, 2003, 2058, 2059
control limits, 2059
frames, 2059
HREF= lines, 2059
inside control limits, 2002
inside stars, 2060
label frames, 2001
outside control limits, 2018, 2060
phase labels, 2060
star outlines, 2004, 2061
STARCIRCLES= circles, 2060
TESTS= option, 2061, 2062
tick marks, 2062
VREF= lines, 2062
Shewhart charts, control limits
appearance, 2069
computing, 1995, 2015, 2030
for autocorrelated data, 2144, 2145, 2148, 2150–2152
for data with multiple components of variation, 2152, 2153, 2155–2159
for nonnormal processes, 2171, 2174–2176
for short-run processes, 2160–2162, 2164–2168, 2170
labeling, 2009, 2045
line type, 2065
multiple sets, 2081, 2083–2085, 2087, 2088, 2090
observations used in computation, 2116
sample size, 2009, 2010
Shewhart charts, fonts
customizing, 2241–2243
Shewhart charts, for autocorrelated data, *see* autocorrelation in process data
Shewhart charts, for data with multiple components of variation, *see* variation, multiple components of
Shewhart charts, for multivariate data, *see* multivariate control charts
Shewhart charts, for nonnormal process data, *see* nonnormal process data
Shewhart charts, for short-run processes, *see* short run process control
Shewhart charts, input data sets
control limits, 2023, 2024
probability limits, 2023
specifying blocks, 2024
Shewhart charts, labeling
angles for, 2064
axes, 1643, 1645, 2109–2113
control limits, 2009, 2012, 2045
fonts for, 2064, 2068
height for, 2062, 2064, 2068
horizontal axis, 2110, 2112, 2113
points, 1994, 2052, 2068
points outside control limits, 2018
reference lines, 2007, 2047
splitting labels, 2031
stars, 2033
tests for special causes, 2040
tick marks, 2030, 2068
vertical axis, 2067, 2111–2113
zone lines, 2050, 2051
Shewhart charts, labeling central line
c chart, 2004
m chart, 2049
p chart, 2023
r chart, 2028
s chart, 2031
u chart, 2045
x chart, 2049
decimal digits, number of, 2012
np chart, 2017
Shewhart charts, line types
reference lines, 2066
star outlines, 2066
STARCIRCLES= circles, 2065
TESTS= option, 2066
Shewhart charts, markers
displaying selected points, 2051
labeling selected points, 2052
suppressing missing groups, 2052
Shewhart charts, nonnormal process data
example, 1962–1966
Shewhart charts, output data sets
chart information, 2019
control limits, 2018, 2019
indicating parameters as estimates or standard values, 2044
subgroup summary statistics, 2018, 2019
Shewhart charts, pages
maximum, 2011
numbering, 2020
splitting, 2057
Shewhart charts, phase variables
control limits, 2021
delineating, 2021
labels, 2021
legends, 2021
Shewhart charts, reference lines

applying to all BY groups, 2013
horizontal axis, 2006, 2007
label position, 2007, 2047
labels, 2007, 2047
line type, 2065, 2066
symbol, 2070, 2071
vertical axis, 2046, 2047
Shewhart charts, specifying parameters
μ_0, 2012
p_0, 2020
σ_0, 2029
u_0, 2045
Shewhart charts, star charts, 2090–2099
contrasted with multivariate control charts, 2092
Shewhart charts, stars
circle outline width, 2069
creating, 2036
inner radius, 2033
labeling, 2033
legends, 2034
outer radius, 2032, 2034
process variables, 2091
reference circles, 2032, 2093, 2094
standardizing, 2035, 2098, 2099
star outline width, 2069
style, 2036, 2095–2097
vertex angle, 2036
vertex variables, 2091, 2092
Shewhart charts, stratification of data, 2071–2080
by *block-variables*, 2074–2078
by a *_PHASE_ variable*, 2079, 2080
by a *symbol-variable*, 2051, 2052, 2073, 2074
by a _PHASE_ *variable*, 2079
Shewhart charts, subgroup selection
using switch variables, 2117, 2118
using WHERE statement, 2113, 2115–2117
Shewhart charts, suppressing features of
central lines, 2014
connecting line segments, 2013
control limit frames, 2066
control limit legends, 2014
control limits, 2014
entire chart, 2013
frames, 2066
horizontal axis labels, 2014
labels, 2014
legends, 2014
line segments, 2016
lower control limits, 2014
phase legend frames, 2067
upper control limits, 2014, 2017
vertical axis labels, 2067
Shewhart charts, symbols
displaying selected points, 2051
labeling selected points, 2052
suppressing missing groups, 2052
Shewhart charts, tables, 2037
adding central line values, 2038
adding control limit exceedances, 2038
adding ID variables, 2038
adding legends, 2038
adding TESTS= results, 2038
box charts, 2038
Shewhart charts, tests for special causes, 2016, 2039–2043, 2068, 2071
across phases, 2016, 2039
customizing tests, 2141, 2142
definitions, 2119, 2121
generalized patterns, 2136–2138, 2140
label angles, 2064
label fonts, 2064, 2068
label height, 2064, 2068
labeling signaled points, 2128, 2134
labels, 2040
line segment character, 2070
M-patterns, 2136–2138, 2140
multiple phases, 2129
multiple sets of control limits, 2131, 2132, 2134
nonstandard tests, 2134–2138, 2140–2142
overlapping points, 2041
range and standard deviation charts, 2134, 2135
reset, 2039, 2041
run lengths, 2039
S-patterns, 2136–2138, 2140
standard tests, 2119, 2121–2125, 2127–2129, 2131, 2132, 2134
standard tests, interpreting, 2124
standard tests, modifying, 2124
standard tests, requesting, 2122, 2123
suppressing 3-sigma check, 2013
T-patterns, 2136–2138, 2140
varying subgroup sample sizes, 2040, 2125, 2127
zone line labels, 2050, 2051
zone lines, 2071
zones, 2050
Shewhart charts, trends
displaying, 2070, 2100, 2102–2104
modeling, 2103, 2104
recognizing, 2102
trend variables, 2044
Shewhart charts, warning limits
vertical axis, 2046
Shewhart charts, Westgard rules, 2048
short run process control, 2160–2162, 2164–2168, 2170
difference from nominal approach, 2161, 2162, 2164–2168
standardization approach, 2170

testing for constant variances, 2168
side-by-side Pareto charts, 1068
sign test, 208
signal-to-noise ratio, 685
signed rank statistic, computing, 227
signed rank test, 208
single-sampling plans, *see* acceptance sampling
size specification, *see* design size specification, FACTEX procedure
skewness
saving in output data set, 427
S_L distribution
histograms, 322
smoothing data distribution, *see* kernel density estimation
S_N distribution
histograms, 326
space-filling designs, *see* optimal designs, space-filling designs
specialized capability indices, 210
specification limits, 210
capability indices, confidence interval, 253
comparative histograms, 285
computing capability indices, example, 199
examples, 248
histograms, example, 300
identifying, 220
lower limit, specification of, 216
reading from data set, example, 248
reference lines, color of, 215, 216
reference lines, example, 251
reference lines, filled areas, 217
reference lines, line type, 216
reference lines, width of, 217
summary information, 199
suppressing legend for, 267, 327
target line, color of, 216
target line, line type, 217
target value, specification of, 216
upper limit, specification of, 216
split-plot designs, 655, 689
stacked Pareto charts, 1068
standard deviation
boxcharts, 2023
CAPABILITY procedure, 211
for median of standard normal, 2235, 2236
range of iid normal variables, 2227
saving in output data set, 427
specifying, 267
standard deviation charts, *see s* charts
star charts, *see* Shewhart charts, star charts
S_U distribution
histograms, 332, 345
subgroup variables
dates or *times*, 1970
indices, 1970
character, 1971
numeric, 1971
sum
saving in output data set, 427
sum of weights
saving in output data set, 427
summary statistics, 204
printing, example, 197
saving, 209, 938
tables, 204
supplementary rules, *see* Shewhart charts, tests for special causes
suppressing features of Shewhart charts, *see* Shewhart charts, suppressing features of
suspended histograms, 320

tables
modes, 208
sign test, 208
signed rank test, 208
trimmed means, 211
Winsorized means, 211
tables, CAPABILITY procedure
summary statistics, 204
tables, Shewhart charts, *see* Shewhart charts, tables
template
macro variables, 1968
test set validation
MVPMODEL procedure, 945
tests for normality, 204
tests for special causes, Shewhart charts, *see* Shewhart charts, tests for special causes, *see* Shewhart charts, tests for special causes
tests of location
location parameter, 208
threshold parameter
probability plots, 483, 487
Q-Q plots, 513, 519
tolerance intervals, *see* intervals, CAPABILITY procedure
tolerance, p-values for
tolerance, *p*-values for, 419
trimmed means, 211, 232
two-way comparative Pareto charts, *see* Pareto charts, comparative
Type A sampling, 2237
Type B sampling, 2237
Type I sum of squares, 700

u charts
central line, 1824
compared with *c* charts, 1824

control limit equations, 1824, 1825
control limit parameters, 1825
examples, advanced, 1831
examples, introductory, 1802
getting started, 1802
known number of nonconformities, specifying, 1833, 1835
labeling axes, 1973
missing values, 1975
notation, 1823
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1814
overview, 1801
plotted points, 1823
plotting character, 1813
reading number of nonconformities, 1808–1811, 1829, 1830
reading preestablished control limits, 1807, 1808, 1829
reading raw data, 1802–1804, 1828
reading subgroup data and control limits, 1830, 1831
saving control limits, 1805, 1806, 1825, 1826
saving nonconformities per unit, 1811, 1812
saving number of nonconformities, 1826, 1827
saving subgroup data and control limits, 1827, 1828
syntax, 1812
tests for special causes, 1831, 1832
unequal subgroup sample sizes, 1835–1838
U-optimal designs, *see* optimal designs, space-filling designs
uniformly weighted moving average charts
adding features to, 891
annotating charts, 891
asymptotic control limits, displaying, 871
axis labels, 888
central line, 874
control limit equations, 874–876
control limits, computing, 871
examples, advanced, 889
examples, introductory, 849
missing values, 888
notation, 874
ODS tables, 882
options summarized by function, 862
overview, 848
plotted points, 874
plotting character, 862
plotting subgroup means, 872
probability limits, 871
process mean, specifying, 872
process standard deviation, specifying, 873
reading preestablished control limit parameters, 859, 860, 884
reading probability limits, 873
reading raw measurements, 849, 851, 852, 883, 884
reading subgroup summary statistics, 853–855, 885
reading summary statistics and control limits, 859, 886
saving control limit parameters, 856, 857, 879, 880
saving subgroup summary statistics, 855, 856, 880, 881
saving summary statistics and control limits, 857, 858, 881, 882
span of moving average, choosing, 876
span parameter, specifying, 874
specifying parameters for, 889, 891
standard deviation, estimating, 886–888
syntax, 861

V-mask charts, *see* cumulative sum control charts
variance
divisors for, 211
saving in output data set, 427
variance of median, *see* STDMED function
variation, multiple components of, 2152, 2153, 2155–2159
determining components, 2156–2159
preliminary examination, 2153, 2155, 2156
VBAR charts
options summarized by function, 1091
syntax, 1090

Weibull distribution
cdf plots, 269
chi-square goodness-of-fit test, 350
deviation from empirical distribution, 351
EDF goodness-of-fit test, 351
Goodness-of-fit tests, 1184
histograms, 333, 346, 379
P-P plots, 454
probability plots, 483, 484
Q-Q plots, 513–515
weighted Pareto charts, 1118
Western Electric rules, *see* Shewhart charts, tests for special causes
Wilcoxon signed rank test, 227
Winsorized means, 211, 232

$\bar{X}$ and R charts
capability indices, computing, 1910, 1971–1973
capability indices, saving, 1890
central line, 1908
control limit equations, 1908

examples, advanced, 1916
examples, introductory, 1882
labeling axes, 1973
missing values, 1975
notation, 1907
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1896
overview, 1881
plotted points, 1907
plotting character, 1895
reading preestablished control limits, 1892, 1912, 1913
reading raw measurements, 1882, 1912
reading subgroup summary statistics, 1885, 1913, 1914
reading summary statistics and control limits, 1892, 1914, 1915
saving control limits, 1889, 1909, 1910
saving subgroup summary statistics, 1888, 1910
saving summary statistics and control limits, 1890, 1892, 1911, 1912
specifying parameters for, 1919, 1921
standard deviation, estimating, 1915, 1916
syntax, 1894
tests for special causes, 1916–1918

$\bar{X}$ and *s* charts
ODS tables, 1966

$\bar{X}$ and *s* charts
capability indices, computing, 1952
central line, 1950
control limit equations, 1950
examples, advanced, 1959
examples, introductory, 1926
labeling axes, 1973
missing values, 1975
notation, 1949
ODS graph names, 1968
options summarized by function, 1938
overview, 1925
plotted points, 1950
reading preestablished control limits, 1935, 1955
reading raw measurements, 1926–1928, 1954
reading subgroup summary statistics, 1929, 1930, 1932, 1955, 1956
reading summary statistics and control limits, 1935, 1956, 1957
saving control limits, 1933, 1951, 1952
saving subgroup summary statistics, 1932, 1952, 1953
saving summary statistics and control limits, 1933, 1935, 1953, 1954
specifying parameters for, 1951
standard deviation, estimating, 1957–1959
syntax, 1936

$\bar{X}$ and *s* charts
plotting character, 1938

$\bar{X}$ charts
capability indices, computing, 1865
central line, 1863
control limit equations, 1863, 1864
examples, advanced, 1873
examples, introductory, 1839
labeling axes, 1973
missing values, 1975
notation, 1863
OC curves, 1879
ODS graph names, 1968
ODS tables, 1966
options summarized by function, 1852
overview, 1838
plotted points, 1863
plotting character, 1852
reading preestablished control limits, 1849, 1850, 1868, 1869
reading raw measurements, 1839, 1841, 1842, 1868
reading subgroup summary statistics, 1842, 1844, 1845, 1869, 1870
reading summary statistics and control limits, 1847, 1848, 1870, 1871
saving control limits, 1846, 1864–1866, 1880
saving subgroup summary statistics, 1845, 1846, 1866
saving summary statistics and control limits, 1847, 1848, 1867
standard deviation, estimating, 1871–1873, 1875, 1877, 1878
syntax, 1851
tests for special causes, 1873, 1874

Syntax Index

ACTUALALPHA option
SHEWHART procedure, 1994
ALLLABEL2= option
CUSUM procedure, 1994
MACONTROL procedure, 1994
SHEWHART procedure, 1994
ALLLABEL= option
CUSUM procedure, 1994
MACONTROL procedure, 1994
SHEWHART procedure, 1994
ALLN option
CUSUM procedure, 1995
MACONTROL procedure, 1995
SHEWHART procedure, 1995, 2123
ALPHA= option
ANOM procedure, 183
chart statement, 969
CUSUM procedure, 579
MACONTROL procedure, 817
SCOREMATRIX statement, 911
SCOREPLOT statement, 912
SHEWHART procedure, 1995
ALPHADELTA= option
CAPABILITY procedure, 534
ALPHAINITIAL= option
CAPABILITY procedure, 534
ALPHALPL= option
CHART statement, 1178
ALPHAUPL= option
CHART statement, 1178
ANCHOR= option
PARETO procedure, 1096
ANGLE= option
PARETO procedure, 1110
ANNOKEY option
CAPABILITY procedure, 539
PARETO procedure, 1110
ANNOTATE2= data set
PARETO procedure, 1079
ANNOTATE2= option
CUSUM procedure, 2056
MACONTROL procedure, 2056
SHEWHART procedure, 2056
ANNOTATE= data set
PARETO procedure, 1079
ANNOTATE= option
CAPABILITY procedure, 539
CUSUM procedure, 2056
MACONTROL procedure, 2056
SHEWHART procedure, 2056
ANOM procedure, 41
syntax, 41
ANOM procedure, all chart statements
ALPHA= option, 183
CINFILL= option, 183
CLIMITS= option, 183
DFE= option, 183
LDLLABEL= option, 184
LIMITK= option, 184
LIMITN= option, 184
LIMLABSUBCHAR= option, 184
LLIMITS= option, 184
MEAN= option, 184
MSE= option, 184
NDECIMAL= option, 184
NOCTL option, 184
NOLDL option, 184
NOLIMIT0 option, 184
NOLIMIT1 option, 184
NOLIMITLABEL option, 184
NOLIMITS option, 184
NOLIMITSFRAME option, 185
NOLIMITSLEGEND option, 185
NONEEDLES option, 185
NOREADLIMITS option, 185
NOUDL option, 185
OUTSUMMARY= option, 185
P= option, 185
PSYMBOL= option, 185
READINDEXES= option, 185
TYPE= option, 185
U= option, 185
UDLLABEL= option, 185
USYMBOL= option, 185
WLIMITS= option, 186
XSYMBOL= option, 186
ANOM procedure, BOXCHART statement, *see also* ANOM procedure, all chart statements
ALPHA= option, 64
BOX= data set, 69
DATA= data set, 70
LIMITN= option, 64
LIMITS= data set, 70, 71
MEAN= option, 64
missing values, 74
MSE= option, 64

NOCHART option, 50, 51
OUTBOX= data set, 64
OUTLIMITS= data set, 51, 52, 66
OUTSUMMARY= data set, 50, 51, 66, 67
OUTTABLE= data set, 52, 53, 68
SUMMARY= data set, 47, 48, 50, 72, 73
TABLE= data set, 53, 73, 74
ANOM procedure, BY statement, 41
ANOM procedure, INSET statement
CFILL= option, 178
CFILLH= option, 178
CFRAME= option, 178
CHEADER= option, 178
CSHADOW= option, 178
CTEXT= option, 178
DATA option, 176
FONT= option, 177
FORMAT= option, 177
HEADER= option, 177
NOFRAME option, 177
POSITION= option, 177, 179, 180
REFPOINT= option, 177
ANOM procedure, PCHART statement, *see also* ANOM procedure, all chart statements
ALPHA= option, 96
DATA= data set, 99
GROUPN= option, 79
LIMITN= option, 96
LIMITS= data set, 100
missing values, 103
OUTLIMITS= data set, 84, 96, 97
OUTSUMMARY= data set, 83, 97
OUTTABLE= data set, 84, 85, 98
P= option, 96
SUMMARY= data set, 81, 82, 101
TABLE= data set, 85, 101, 102
ANOM procedure, UCHART statement, *see also* ANOM procedure, all chart statements
ALPHA= option, 120
DATA= data set, 123, 124
GROUPN= option, 107
LIMITN= option, 120
LIMITS= data set, 124, 125
missing values, 127
NOCHART option, 108, 109
OUTLIMITS= data set, 108, 121
OUTSUMMARY= data set, 121, 122
OUTTABLE= data set, 109, 110, 122, 123
SUMMARY= data set, 125
TABLE= data set, 110, 126
U= option, 120
ANOM procedure, XCHART statement, *see also* ANOM procedure, all chart statements
ALPHA= option, 148, 150
DATA= data set, 153, 154
LIMITN= option, 148, 150
LIMITS= data set, 154, 155
MEAN= option, 148, 150
missing values, 157
MSE= option, 148, 150
NOCHART option, 135
OUTLIMITS= data set, 136, 150, 151
OUTSUMMARY= data set, 135, 136, 151
OUTTABLE= data set, 137, 152
SUMMARY= data set, 133–135, 155
TABLE= data set, 137, 156
AOQ2 function, 2217, 2218, 2239
ASN2 function, 2218, 2219, 2239
ASYMPTOTIC option
MACONTROL procedure, 817, 871
ATI2 function, 2220, 2221, 2239
AXISFACTOR option
PARETO procedure, 1096

BARLABEL= option
PARETO procedure, 1096
BARLABPOS= option
PARETO procedure, 1111
BARLEGEND= option
PARETO procedure, 1097
BARLEGLABEL= option
PARETO procedure, 1097
BARS= option
PARETO procedure, 1098
BARWIDTH= option
PARETO procedure, 1111
BAYESACT call, 2221–2223
BETA= option
CUSUM procedure, 579
BILEVEL option
CUSUM procedure, 2057
MACONTROL procedure, 2057
SHEWHART procedure, 2057
block-variables, ANOM procedure
BOXCHART statement, 54
PCHART statement, 86
UCHART statement, 111
XCHART statement, 138
block-variables, CUSUM procedure
XCHART statement, 570
block-variables, MACONTROL procedure
EWMACHART statement, 808
MACHART statement, 861
block-variables, SHEWHART procedure
BOXCHART statement, 1433
CCHART statement, 1493
displaying values, 1997
IRCHART statement, 1529

labels, 1995, 1996
legends, 1996, 2057
MCHART statement, 1573
MRCHART statement, 1616
NPCHART statement, 1657
PCHART statement, 1698
RCHART statement, 1742
SCHART statement, 1779
UCHART statement, 1813
XCHART statement, 1851
XRCHART statement, 1895
XSCHART statement, 1937
BLOCKLABELPOS= option
CUSUM procedure, 1995
MACONTROL procedure, 1995
SHEWHART procedure, 1995, 2078, 2168
BLOCKLABTYPE= option
CUSUM procedure, 1996
MACONTROL procedure, 1996
SHEWHART procedure, 1996, 2168
BLOCKPOS= option
CUSUM procedure, 1996
MACONTROL procedure, 1996
SHEWHART procedure, 1996, 2076–2078
BLOCKREFTRANSPARENCY= option
ANOM procedure, 2051
CUSUM procedure, 2051
MACONTROL procedure, 2051
SHEWHART procedure, 2051
BLOCKREP option
CUSUM procedure, 1997
MACONTROL procedure, 1997
SHEWHART procedure, 1997
BLOCKS statement, FACTEX procedure, *see* FACTEX procedure, BLOCKS statement
syntax, 630
BLOCKS statement, OPTEX procedure, *see* OPTEX procedure, BLOCKS statement
syntax, 1014
BLOCKVAR= option
ANOM procedure, 1997
CUSUM procedure, 1997
MACONTROL procedure, 1997
SHEWHART procedure, 1997
BOXCHART statement, *see also* SHEWHART procedure, BOXCHART statement
examples, advanced, 1461
examples, introductory, 1418
options summarized by function, 1434, 1445
overview, 1417
syntax, 1432
BOXCHART statement, ANOM procedure, *see also* ANOM procedure, BOXCHART statement
examples, advanced, 75
examples, introductory, 45
options summarized by function, 54, 61
overview, 44
syntax, 53
BOXCONNECT option
SHEWHART procedure, 1998
BOXES= option
ANOM procedure, 1998
SHEWHART procedure, 1998
BOXFILL= option
ANOM procedure, 1998
SHEWHART procedure, 1998
BOXSTYLE= option
SHEWHART procedure, 1477, 1998
BOXSTYLE= option, SHEWHART procedure, 1481
BOXTRANSPARENCY= option
ANOM procedure, 2051
SHEWHART procedure, 2051
BOXWIDTH= option
SHEWHART procedure, 2001
BOXWIDTHSCALE= option
SHEWHART procedure, 2001
BY statement
ANOM procedure, 41
CAPABILITY procedure, 212
CUSUM procedure, 552
MACONTROL procedure, 793
MVPDIAGNOSE procedure, 907
MVPMODEL procedure, 941
MVPMONITOR procedure, 964
PARETO procedure, 1080
RAREEVENTS procedure, 1176
RELIABILITY procedure, 1279
SHEWHART procedure, 1410

C4 function, 2223, 2224, 2239
CAPABILITY procedure, 201
and PROC SHEWHART, 2173, 2174
introduction, 193
syntax, 201
CAPABILITY procedure, BY statement, 212
CAPABILITY procedure, CDFPLOT statement
ALPHA= beta-option, 263
ALPHA= gamma-option, 263
BETA beta-option, 263
BETA= option, 264
C= option, 264
CDFSYMBOL= option, 264
EXPONENTIAL option, 264
GAMMA option, 264
GUMBEL option, 265
IGAUSS option, 265
LAMBDA= iGauss-option, 266
LEGEND= option, 266

LOGNORMAL option, 266
MU= option, 266
NOCDFLEGEND option, 267
NOECDF option, 267
NOLEGEND option, 267
NORMAL option, 267
NOSPECLEGEND option, 267
PARETO option, 267
POWER option, 268
RAYLEIGH option, 268
SIGMA= option, 269
SYMBOL= option, 269
THETA= option, 269
THRESHOLD= option, 269
VSCALE= option, 269
WEIBULL Weibull-option, 269
ZETA= option, 270
CAPABILITY procedure, CLASS statement
KEYLEVEL= option, 213
MISSING option, 212
NOKEYMOVE option, 214
ORDER= option, 212
CAPABILITY procedure, COMPHISTOGRAM statement
BARLABEL= option, 284
BARWIDTH= option, 292
C= option, 284
CBARLINE= option, 292
CFILL= option, 292
CFRAMENLEG= option, 292
CGRID= option, 292
CLASS= option, 279
CLASSKEY= option, 285
CLASSSPEC= option, 285
CLIPSPEC= option, 292
ENDPOINTS= option, 286, 316
FILL option, 287
FRONTREF option, 292
GRID option, 287
HOFFSET= option, 292
INTERTILE= option, 287
K= option, 287
KERNEL kernel-option, 284, 287
LGRID= option, 292
LOWER= option, 288
MAXNBIN= option, 288
MAXSIGMAS= option, 288
MIDPOINTS= option, 288
MISSING1 option, 289
MISSING2 option, 289
MU= option, 289
NLEGEND option, 292, 293
NLEGENDPOS option, 293
NOBARS option, 289
NOCHART option, 289
NOKEYMOVE option, 289
NOPLOT option, 289
NORMAL normal-option, 289
ORDER1= option, 290
ORDER2= option, 291
OUTHISTOGRAM= option, 291
PFILL= option, 293
RTINCLUDE option, 291
SIGMA= option, 291
TILELEGLABEL= option, 293
UPPER= option, 291
VOFFSET= option, 293
VSCALE= option, 291
WBARLINE= option, 293
WGRID= option, 293
CAPABILITY procedure, HISTOGRAM statement
ALPHA= option, 314, 337
BARLABEL= option, 314
BARWIDTH= option, 334
BETA beta-option, 314, 336
BETA= option, 315, 337
BMCFILL= option, 334
BMCFRAME= option, 334
BMCOLOR= option, 334
BMMARGIN= option, 334
BMPLOT= option, 315
C= option, 315, 347
CBARLINE= option, 334
CFILL= option, 334
CGRID= option, 334
CLIPREF option, 334
CLIPSPEC= option, 334
CURVELEGEND= option, 335
DELTA= option, 316, 344, 345
EDFNSAMPLES= option, 316
EDFSEED= option, 316
EXPONENTIAL exponential-option, 317, 337
FILL option, 317, 318
FITINTERVAL= option, 318
FITMETHOD= option, 318
FITTOLERANCE= option, 318
FRONTREF option, 335
GAMMA gamma-option, 318, 338
GAMMA= option, 319, 344, 345
GRID option, 319
GUMBEL Gumbel-option, 339
GUMBEL option, 319
HANGING option, 320
HOFFSET= option, 335
IGAUSS iGauss-option, 339
IGAUSS option, 320
INDICES option, 321, 354, 355
INTERBAR= option, 335

K= option, 321, 347
KERNEL option, 321, 347
LAMBDA= iGauss-option, 322
LEGEND= option, 335
LGRID= option, 335
LOGNORMAL lognormal-option, 322, 340
MAXNBIN= option, 323
MAXSIGMAS= option, 323
MIDPERCENTS option, 323, 355
MIDPOINTS= option, 324
MIDPTAXIS= option, 325
MU= option, 325, 340
NENDPOINTS= option, 325
NMIDPOINTS= option, 325
NOBARS option, 326
NOCURVELEGEND option, 326
NOLEGEND option, 326
NOPLOT option, 326
NOPRINT option, 326
NORMAL normal-option, 326, 340
NOSPECLEGEND option, 327
NOTABCONTENTS option, 327
OUTFIT= option, 327, 356
OUTHISTOGRAM= option, 327, 356, 358
OUTKERNEL= option, 328, 356, 359
PARETO option, 328
PARETO Pareto-option, 341
PCTAXIS= option, 328
PERCENTS= option, 328, 355
PFILL= option, 335
POWER option, 329
POWER power-option, 342
RAYLEIGH option, 329
RAYLEIGH Rayleigh-option, 343
RTINCLUDE option, 330
SB option, 330, 344
SCALE= option, 338, 346
SHAPE= option, 340, 346
SIGMA= option, 331, 337, 340, 344, 345
SPECLEGEND= option, 335
SU option, 332, 345
SYMBOL= option, 336
THETA= option, 332, 337
THRESHOLD= option, 332, 338, 340, 344–346
VOFFSET= option, 335
VSCALE= option, 333
WBARLINE= option, 335
WEIBULL option, 333, 346
WGRID= option, 335
ZETA= option, 334
CAPABILITY procedure, INSET statement
CFILL= option, 403
CFILLH= option, 404
CFRAME= option, 404
CHEADER= option, 404
CSHADOW= option, 404
CTEXT= option, 404
DATA option, 403
displaying C_{pk}, 533
FONT= option, 404
FORMAT= option, 403
GUTTER= option, 403
HEADER= option, 403
HEIGHT= option, 404
NCOLS= option, 403
NOFRAME option, 403
POSITION= option, 403, 405–407
REFPOINT= option, 404
CAPABILITY procedure, INTERVALS statement
ALPHA= option, 418
K= option, 418
METHODS= option, 418, 420–422
NOPRINT option, 419
OUTINTERVALS= option, 419, 423
P= option, 419
TYPE= option, 419
CAPABILITY procedure, OUTPUT statement
OUT= option, 427, 433
PCTLGROUP= option, 431
PCTLNAME= option, 432
PCTLNDEC= option, 432
PCTLPRE= option, 432
PCTLPTS= option, 433
CAPABILITY procedure, plot statements
ALPHADELTA= gamma-option, 534
ALPHAINITIAL= gamma-option, 534
ANNOKEY option, 539
ANNOTATE= option, 539
CAXIS= option, 539
CDELTA= option, 534
CFRAME= option, 539
CFRAMESIDE= option, 539
CFRAMETOP= option, 539
CHREF= option, 539
CINITIAL= option, 535
COLOR= option, 540
CONTENTS= option, 535
CPROP= option, 535
CSTATREF= option, 540
CTEXT= option, 540
CTEXTSIDE= option, 540
CTEXTTOP= option, 540
CVREF= option, 540
DESCRIPTION= option, 540
FONT= option, 540
HAXIS= option, 535
HEIGHT= option, 540
HMINOR= option, 541

HREF= option, 535
HREFCHAR= option, 542
HREFLABELS= option, 535
HREFLABPOS= option, 535
INFONT= option, 541
INHEIGHT= option, 541
INTERTILE= option, 535
L= option, 541
LHREF= option, 541
LSTATREF= option, 541
LVREF= option, 541
MAXITER= option, 535
NAME= option, 541
NCOLS= option, 536
NOFRAME option, 541
NOHLABEL option, 536
NOVLABEL option, 536
NOVTICK option, 536
NROWS= option, 536
ODSFOOTNOTE2= option, 536
ODSFOOTNOTE= option, 536
ODSTITLE2= option, 537
ODSTITLE= option, 537
OVERLAY option, 537
SCALE= option, 537
SHAPE= option, 537
STATREF= option, 538
STATREFLABELS= option, 538
STATREFSUBCHAR= option, 538
TURNVLABELS option, 542
VAXIS= option, 538
VAXISLABEL= option, 538
VMINOR= option, 542
VREF= option, 538
VREFCHAR= option, 542
VREFLABELS= option, 539
VREFLABPOS= option, 539
W= option, 542
WAXIS= option, 542
CAPABILITY procedure, PPPLOT statement
ALPHA= option, 447, 449
BETA option, 444, 447
BETA= option, 448
C= option, 448, 455
COLOR= option, 442
EXPONENTIAL option, 444, 448
GAMMA option, 444, 449
GUMBEL option, 449
IGAUSS option, 450
LAMBDA= option, 450
LOGNORMAL option, 444, 450
MU= option, 444, 451, 452
NOLINE option, 451
NOOBSLEGEND option, 451
NORMAL option, 444, 451
PARETO option, 452
POWER option, 452
PPSYMBOL= option, 453
RAYLEIGH option, 453
SCALE= option, 449, 451
SHAPE= option, 449, 451
SIGMA= option, 444, 449, 451, 452, 454, 455
SQUARE option, 442, 454
SYMBOL= option, 454
THETA= option, 449, 451, 454, 455
THRESHOLD= option, 449, 451, 454
VAXIS= option, 456
WEIBULL option, 444, 454
ZETA= option, 451, 455
CAPABILITY procedure, PROBPLOT statement
ALPHA= option, 473
BETA option, 470, 473
BETA= option, 474
C= option, 474, 483, 485
CGRID= option, 485
EXPONENTIAL option, 470, 475
GAMMA option, 470, 475
GRID option, 476, 505
GRIDCHAR= option, 485
GUMBEL option, 476, 505
HREF= option, 492
HREFLABELS= option, 492
LEGEND= option, 485
LGRID= option, 485
LOGNORMAL option, 470, 477
MU= option, 478, 479
NADJ= option, 478, 484
NOLEGEND option, 478
NOLINELEGEND option, 478
NOOBSLEGEND option, 485
NORMAL option, 470, 478
NOSPECLEGEND option, 479
PARETO option, 479, 508
PCTLMINOR option, 485, 492
PCTLORDER= option, 480
POWER option, 480, 511
PROBSYMBOL option, 486
RANKADJ= option, 481, 484
RAYLEIGH option, 481, 511
ROTATE option, 481
SCALE= option, 475, 476, 484
SHAPE= option, 483
SIGMA= option, 474, 479, 481, 485
SLOPE= option, 482
SQUARE option, 483, 492
SYMBOL= option, 486
THETA= option, 474, 478, 483
THRESHOLD= option, 475, 476, 483, 484

VAXIS= option, 491
WEIBULL option, 470, 483
WEIBULL2 option, 470, 484
WGRID= option, 485
ZETA= option, 478, 485
CAPABILITY procedure, PROC CAPABILITY statement
ALL option, 204
ALPHA= option, 204–206, 211, 253, 2002
ANNOTATE= option, 204, 221
CHECKINDICES option, 204
CIBASIC= option, 205
CIINDICES= option, 205
CIPCTLDF= option, 206
CIPCTLNORMAL= option, 206
CIPROBEX option, 206
CIQUANTDF= option, 206
CIQUANTNORMAL= option, 206
CPMA= option, 206, 210
DATA= option, 207, 219
DEF= option, 207, 209
EXCLNPWGT option, 207
FORCEQN option, 207
FORCESN option, 207
FORMCHAR= option, 207
FREQ option, 208
GOUT= option, 208
LINEPRINTER option, 208
LOCATION= option, 208
LOCCOUNT option, 208
missing values, 246
MODE option, 208
MODES option, 208, 234
MUO= option, 208
NEXTROBS= option, 209
NEXTRVAL= option, 209
NOBYSPECS option, 209
NOPRINT option, 209
NORMALTEST option, 209, 227
ODS tables, 247
OUTTABLE= option, 209, 222
PCTLDEF= option, 207, 209, 230
ROBUSTSCALE option, 209, 232
ROUND= option, 209
SPEC= option, 210, 220
SPECIALINDICES option, 210
TRIM option, 211
TRIMMED option, 211
TRIMMED= option, 232
TYPE= option, 205, 206, 211, 2002
VARDEF= option, 211
WINSOR option, 211
WINSORIZED option, 211
WINSORIZED= option, 232
CAPABILITY procedure, QQPLOT statement
ALPHA= option, 502, 504
BETA option, 498, 499, 502
BETA= option, 503
C= option, 503, 513, 515
CGRID= option, 515
COLOR= option, 496, 498
CPKREF option, 507, 515, 533
CPKSCALE option, 503, 507, 533
EXPONENTIAL option, 498, 499, 503
GAMMA option, 498, 499, 504
GRID option, 508
GRIDCHAR= option, 508
L= option, 496
LABEL= option, 508
LEGEND= option, 505
LGRID= option, 508, 515
LOGNORMAL option, 498, 499, 505
MU= option, 496, 498, 506, 507
NADJ= option, 506, 517
NOLEGEND option, 507
NOLINELEGEND option, 507
NOOBSLEGEND option, 515
NORMAL option, 498, 499, 507, 533
NOSPECLEGEND option, 496, 508
PCTLAXIS option, 508, 521
PCTLMINOR option, 515
PCTLSCALE option, 510, 521
QQSYMBOL= option, 516
RANKADJ= option, 511, 517
ROTATE option, 512
SCALE= option, 503–506, 514
SHAPE= option, 504, 505, 513
SIGMA= option, 496, 498, 503–505, 507, 512, 514, 515
SLOPE= option, 506, 512, 515
SQUARE option, 496, 513
SYMBOL= option, 516
THETA= option, 503–506, 513, 514
THRESHOLD= option, 503–506, 513, 514
WEIBULL option, 498, 499, 513
WEIBULL2 option, 498, 499, 514
WGRID= option, 515
ZETA= option, 506, 515
CAPABILITY procedure, SPEC statement
CLEFT= option, 215
CLSL= option, 216
CRIGHT= option, 216
CTARGET= option, 216
CUSL= option, 216
LLSL= option, 216
LSL= option, 216
LSLSYMBOL= option, 217
LTARGET= option, 217

LUSL= option, 217
PLEFT= option, 217
PRIGHT= option, 217
TARGET= option, 216
TARGETSYMBOL= option, 217
USL= option, 216
USLSYMBOL= option, 218
WLSL= option, 217
WTARGET= option, 217
WUSL= option, 217
CATLEGEND= option
PARETO procedure, 1098
CATLEGLABEL= option
PARETO procedure, 1098
CATOFFSET= option
PARETO procedure, 1098
CATREF= option
PARETO procedure, 1098
CATREFLABELS= option
PARETO procedure, 1098
CAXIS2= option
PARETO procedure, 1111
CAXIS= option
CAPABILITY procedure, 539
CUSUM procedure, 2057
MACONTROL procedure, 2057
PARETO procedure, 1111
SHEWHART procedure, 2057
CBARLINE= option
PARETO procedure, 1112
CBARS= option
PARETO procedure, 1112
CBLOCKLAB= option
CUSUM procedure, 2057
MACONTROL procedure, 2057
SHEWHART procedure, 2057
CBLOCKVAR= option
CUSUM procedure, 2057
MACONTROL procedure, 2057
SHEWHART procedure, 2057, 2077, 2078
CBOXES= option
SHEWHART procedure, 2057
CBOXFILL= option
SHEWHART procedure, 2058
CCHART statement, SHEWHART procedure, *see* SHEWHART procedure, CCHART statement
examples, advanced, 1511
examples, introductory, 1483
options summarized by function, 1494
overview, 1482
syntax, 1492
CCLIP= option
MACONTROL procedure, 2058
SHEWHART procedure, 2058
CCONNECT= option
CUSUM procedure, 2058
MACONTROL procedure, 2058
PARETO procedure, 1112
SHEWHART procedure, 2058
CCOVERLAY2= option
SHEWHART procedure, 2058
CCOVERLAY= option
SHEWHART procedure, 2058
CCUMREF= option
PARETO procedure, 1112
CDFPLOT statement, *see* CAPABILITY procedure, CDFPLOT statement
examples, 271, 273
getting started, 256
options summarized by function, 258, 260, 263
overview, 255
syntax, 257
CFRAME= option
CAPABILITY procedure, 539
CUSUM procedure, 2059
MACONTROL procedure, 2059
PARETO procedure, 1112
SHEWHART procedure, 2059, 2079
CFRAMELAB= option
CUSUM procedure, 2001
MACONTROL procedure, 2001
SHEWHART procedure, 2001
CFRAMENLEG= option
PARETO procedure, 1098
CFRAMESIDE= option
CAPABILITY procedure, 539
PARETO procedure, 1112
CFRAMETOP= option
CAPABILITY procedure, 539
PARETO procedure, 1113
CFREQREF= option
PARETO procedure, 1113
CGRID2= option
PARETO procedure, 1113
CGRID= option
CUSUM procedure, 2059
MACONTROL procedure, 2059
PARETO procedure, 1113
SHEWHART procedure, 2059
character subgroup variables
SHEWHART procedure, 2012
CHART statement
RAREEVENTS procedure, 1177
CHART statement options
RAREEVENTS procedure, 1177, 1180
chart statement options
MVPMONITOR procedure, 969

CHARTTYPE= option
PARETO procedure, 1099
CHIGH(*n*)= option
PARETO procedure, 1099
CHREF= option
CAPABILITY procedure, 539
CUSUM procedure, 2059
MACONTROL procedure, 2059
SHEWHART procedure, 2059
CINFILL= option
ANOM procedure, 183
CUSUM procedure, 584
MACONTROL procedure, 2002
SHEWHART procedure, 2002
CLABEL= option
CUSUM procedure, 2059
MACONTROL procedure, 2059
SHEWHART procedure, 2059
CLASS statement
syntax, 212
CLASS statement, OPTEX procedure, *see* OPTEX procedure, CLASS statement
syntax, 1015
CLASS= option
PARETO procedure, 1099
CLASSKEY= option
PARETO procedure, 1100
CLIMITS= option
ANOM procedure, 183
CUSUM procedure, 584
MACONTROL procedure, 2059
SHEWHART procedure, 2059
CLIPCHAR= option
MACONTROL procedure, 2070
SHEWHART procedure, 2070
CLIPFACTOR= option
MACONTROL procedure, 2002
SHEWHART procedure, 2002, 2106–2108
CLIPLEGEND= option
MACONTROL procedure, 2002
SHEWHART procedure, 2002, 2108
CLIPLEGPOS= option
MACONTROL procedure, 2059
SHEWHART procedure, 2059, 2108
CLIPREF option
PARETO procedure, 1113
CLIPSUBCHAR= option
MACONTROL procedure, 2002
SHEWHART procedure, 2002, 2108
CLIPSYMBOL= option
MACONTROL procedure, 2059
SHEWHART procedure, 2059, 2108
CLIPSYMBOLHT= option
SHEWHART procedure, 2059
CLOW(*n*)= option
PARETO procedure, 1100
CMASK= option
CUSUM procedure, 584
CMEANSYMBOL= option
MACONTROL procedure, 818, 872
CMPCTLABEL option
PARETO procedure, 1101
CNEEDLES= option
CUSUM procedure, 2059
MACONTROL procedure, 2059
SHEWHART procedure, 2059, 2104
COMP= option
SCORECHART statement, 966
COMPARE statement
RAREEVENTS procedure, 1179
COMPARE statement options
RAREEVENTS procedure, 1180
COMPHISTOGRAM statement, *see* CAPABILITY procedure, COMPHISTOGRAM statement
examples, 276, 277
getting started, 275
options summarized by function, 280, 281, 283
overview, 274
syntax, 278
CONNECTCHAR= option
CUSUM procedure, 2070
MACONTROL procedure, 2070
PARETO procedure, 1117
SHEWHART procedure, 2070
CONTENTS= option
CAPABILITY procedure, 535
CONTRIBUTIONPANEL statement
MVPDIAGNOSE procedure, 908
CONTRIBUTIONPLOT statement
MVPDIAGNOSE procedure, 909
CONTRIBUTIONS option
chart statement, 969
CONTROLSTAT= option
SHEWHART procedure, 2003
COTHER= option
PARETO procedure, 1113
COUT= option
CUSUM procedure, 2003
MACONTROL procedure, 2003
SHEWHART procedure, 2003
COUTFILL= option
CUSUM procedure, 2060
MACONTROL procedure, 2060
SHEWHART procedure, 2060
COV option
PROC MVPMODEL statement, 935
COVERLAY2= option
SHEWHART procedure, 2060

COVERLAY= option
SHEWHART procedure, 2060
COVERLAYCLIP= option
SHEWHART procedure, 2060
CPHASEBOX= option
SHEWHART procedure, 1477, 2003
CPHASEBOXCONNECT= option
SHEWHART procedure, 2003
CPHASEBOXFILL= option
SHEWHART procedure, 1477, 2003
CPHASELEG= option
CUSUM procedure, 2060
MACONTROL procedure, 2060
SHEWHART procedure, 2060, 2079
CPHASEMEANCONNECT= option
SHEWHART procedure, 1477, 2004
CPMA= option
CAPABILITY procedure, 210
CPROP= option
CAPABILITY procedure, 535
PARETO procedure, 1101
CSTARCIRCLES= option
CUSUM procedure, 2060
MACONTROL procedure, 2060
SHEWHART procedure, 2060
CSTARFILL= option
CUSUM procedure, 2060
MACONTROL procedure, 2060
SHEWHART procedure, 2060
CSTAROUT= option
CUSUM procedure, 2004
MACONTROL procedure, 2004
SHEWHART procedure, 2004
CSTARS= option
CUSUM procedure, 2061
MACONTROL procedure, 2061
SHEWHART procedure, 2061
CSTATREF= option
CAPABILITY procedure, 540
CSYMBOL= option
SHEWHART procedure, 2004
CTESTLABBOX= option
SHEWHART procedure, 2061
CTESTS= option
SHEWHART procedure, 2061, 2134
CTESTSYMBOL= option
SHEWHART procedure, 2062
CTEXT= option
CAPABILITY procedure, 540
CUSUM procedure, 2062
MACONTROL procedure, 2062
PARETO procedure, 1113
SHEWHART procedure, 2062
CTEXTSIDE= option
CAPABILITY procedure, 540
PARETO procedure, 1113
CTEXTTOP= option
CAPABILITY procedure, 540
PARETO procedure, 1113
CTILES= option
PARETO procedure, 1113
CUMAXIS= option
PARETO procedure, 1101
CUMAXISLABEL= option
PARETO procedure, 1101
CUMREF= option
PARETO procedure, 1101
CUMREFLABELS= option
PARETO procedure, 1101
CUSUM procedure, 550
ANNOTATE2= option, 550
ANNOTATE= option, 550
DATA= data set, 550
FORMCHAR= option, 551
GOUT= option, 551
GRAPHICS option, 566
HISTORY= data set, 552
introduction, 548
LIMITS= data set, 552
LINEPRINTER option, 552
overview, 549
syntax, 550
CUSUM procedure, BY statement, 552
CUSUM procedure, INSET statement, *see* INSET and INSET2 statements
getting started, 610
overview, 610
syntax, 612
CUSUM procedure, XCHART statement
ALLN option, 587
ALPHA= option, 556, 569, 579, 590
BETA= option, 579, 590
CINFILL= option, 584
CLIMITS= option, 584
CMASK= option, 584
DATA= data set, 555–557, 600
DATAUNITS option, 580, 586
DELTA= option, 556, 569, 580, 585
H= option, 562, 564, 569, 580, 590
HEADSTART= option, 580, 586
HISTORY= data set, 558, 559, 601, 602
INTERVAL= option, 588
K= option, 562, 564, 580, 590
LIMITN= option, 581, 585, 587
LIMITS= data set, 567, 569, 600, 601
LLIMITS= option, 584
LMASK= option, 584
missing values, 602

MU0= option, 556, 569, 581, 585
NOARL option, 581
NOMASK option, 581
NOREADLIMITS option, 581
ORIGIN= option, 581
OUTHISTORY= data set, 560, 561, 597
OUTLIMITS= data set, 565, 566, 596, 597
OUTTABLE= data set, 598, 602, 605
READINDEX= option, 582
READLIMITS option, 582
READSIGMAS option, 582
SCHEME= option, 562, 564, 569, 582
SHIFT= option, 582, 585
SIGMA0= option, 556, 583
SIGMAS= option, 583
SMETHOD= option, 583, 594–596
TABLEALL option, 562, 564, 583
TABLECHART option, 583
TABLECOMP option, 583
TABLEID option, 583
TABLEOUT option, 584
TABLESUMMARY option, 584
TYPE= option, 584, 585
VAXIS= option, 556
WLIMITS= option, 584
WMASK= option, 584
CUSUMARL function, 2224, 2225
CV= option
PROC MVPMODEL statement, 935
CVREF= option
CAPABILITY procedure, 540
CUSUM procedure, 2062
MACONTROL procedure, 2062
SHEWHART procedure, 2062
CZONES= option
SHEWHART procedure, 2062, 2134

D2 function, 2226, 2239
D3 function, 2227, 2239
DATA= data set
PARETO procedure, 1080
DATA= option
PROC MVPDIAGNOSE statement, 906, 915
PROC MVPMODEL statement, 937, 946
PROC MVPMONITOR statement, 963, 975
PROC RAREEVENTS statement, 1176, 1187
DATAUNIT= option
SHEWHART procedure, 2004
DATAUNITS option
CUSUM procedure, 580
DELTA= option
CUSUM procedure, 580
DESCENDING option
CLASS statement (OPTEX), 1016
DESCRIPTION2= option
SHEWHART procedure, 2062
DESCRIPTION= option
CAPABILITY procedure, 540
CUSUM procedure, 2062
MACONTROL procedure, 2062
PARETO procedure, 1114
SHEWHART procedure, 2062
DFE= option
ANOM procedure, 183
DISCRETE option
CUSUM procedure, 2005
MACONTROL procedure, 2005
SHEWHART procedure, 2005
DIST= option
CHART statement, 1180
COMPARE statement, 1180

EFFECTPLOT statement
RELIABILITY procedure, 1281
ELLIPSE option
SCOREMATRIX statement, 911
SCOREPLOT statement, 912
ENDGRID option
CUSUM procedure, 2062
MACONTROL procedure, 2062
SHEWHART procedure, 2062
ESTIMATE statement
RELIABILITY procedure, 1282
EWMAARL function, 2228
EWMACHART statement, *see also* MACONTROL procedure, EWMACHART statement
examples, advanced, 835
examples, introductory, 796
overview, 795
syntax, 807
EXAMINE statement, FACTEX procedure, *see* FACTEX procedure, EXAMINE statement
syntax, 632
EXAMINE statement, OPTEX procedure, *see* OPTEX procedure, EXAMINE statement
syntax, 1019
EXCHART option
CHART statement, 1178
chart statement, 969
CUSUM procedure, 2005
MACONTROL procedure, 2005
SHEWHART procedure, 2005

FACTEX procedure, 627
getting started, 620
overview, 618
summary of functions, 627
syntax, 627

FACTEX procedure, BLOCKS statement
NBLKFACS= option, 630
NBLKFACS=MAXIMUM option, 631
NBLOCKS= option, 630
NBLOCKS= option, examples, 622, 678
NBLOCKS=MAXIMUM option, 631
SIZE= option, 631
SIZE=MINIMUM option, 631
UNITS= option, 631
FACTEX procedure, EXAMINE statement
ALIASING option, 632
ALIASING option, example, 625
CONFOUNDING option, 632, 633
DESIGN option, 633
DESIGN option, example, 620
SUMMARY option, 633
FACTEX procedure, FACTORS statement
example, 620
NLEV= option, 634
FACTEX procedure, MODEL statement
ESTIMATE= option, 634
ESTIMATE= option, examples, 662, 679
MAXCLEAR option, 635
MINABS option, 636, 654
MINABS option, example, 677
MINABS option, limitation, 678
NONNEGLIGIBLE= option, 634
RESOLUTION= option, 635
RESOLUTION= option, examples, 624, 660, 664
RESOLUTION=MAX option, 635
RESOLUTION=MAX option, examples, 622, 668, 669
FACTEX procedure, OUTPUT statement
CVALS= option, 637, 638, 648
CVALS= option, example, 666
decode design factors, 637
derived factors, 637
derived factors, examples, 672, 674
DESIGNREP= option, 638
DESIGNREP= option, examples, 667–672
NOVALRAN option, 639
NVALS= option, 637, 638, 648
NVALS= option, example, 666
OUT= option, 637
OUT= option, example, 666
POINTREP= option, 639
POINTREP= option, examples, 667–672
RANDOMIZE= option, 639
RANDOMIZE= option, examples, 659, 666
RANDOMIZE= option, NOVALRAN option, 639
RANDOMIZE= option, seed, 639
recode block factor, 637
recode block factor levels, examples, 623, 666
recode design factor levels, examples, 621, 624, 666
FACTEX procedure, PROC FACTEX statement
example, 620
NAMELEN option, 630
NOCHECK option, 630, 645, 678
ODS tables, 659
SECONDS= option, 630
TIME= option, 630, 678
FACTEX procedure, SIZE statement
DESIGN= option, 639
DESIGN= option, examples, 624, 660
DESIGN=MINIMUM option, 640
FRACTION= option, 640
FRACTION=MAXIMUM option, 640
NRUNFACS= option, 640
NRUNFACS=MINIMUM option, 640
FACTEX procedure, UNITEFFECT statement
syntax, 640
FACTORS statement, FACTEX procedure, *see* FACTEX procedure, FACTORS statement
syntax, 634
FITINTERVAL= option
CAPABILITY procedure, 318
FITMETHOD= option
CAPABILITY procedure, 318
FITTOLERANCE= option
CAPABILITY procedure, 318
FONT= option
CAPABILITY procedure, 540
CUSUM procedure, 2062
MACONTROL procedure, 2062
PARETO procedure, 1114
SHEWHART procedure, 1722, 2062
FORMCHAR= option
PARETO procedure, 1080
FREQ= option
PARETO procedure, 1101
FREQAXIS= option
PARETO procedure, 1102
FREQAXISLABEL= option
PARETO procedure, 1102
FREQOFFSET= option
PARETO procedure, 1102
FREQREF= option
PARETO procedure, 1102
FREQREFLABELS= option
PARETO procedure, 1102
FRONTREF option
ANOM procedure, 2005
SHEWHART procedure, 2005

GENERATE statement, OPTEX procedure, *see* OPTEX procedure, GENERATE statement

default options, 1020
syntax, 1020
GOUT= option
PARETO procedure, 1080
GRID option
CUSUM procedure, 2005
MACONTROL procedure, 2005
PARETO procedure, 1102
SHEWHART procedure, 2005
GRID2 option
PARETO procedure, 1102
group-variable, ANOM procedure
BOXCHART statement, 54
PCHART statement, 86
UCHART statement, 111
XCHART statement, 138
GROUP= option
SCOREMATRIX statement, 911
SCOREPLOT statement, 912
GROUPN= option
ANOM procedure, 183

H= option
CUSUM procedure, 580
HAXIS= option
CAPABILITY procedure, 535
CUSUM procedure, 2005
MACONTROL procedure, 2005
SHEWHART procedure, 2005
HEADSTART= option
CUSUM procedure, 580
HEIGHT= option
CAPABILITY procedure, 540
CUSUM procedure, 2062
MACONTROL procedure, 2062
PARETO procedure, 1114
SHEWHART procedure, 2062
HISTOGRAM statement, *see* CAPABILITY procedure, HISTOGRAM statement
getting started, 300
options summarized by function, 306, 308, 311
overview, 299
syntax, 305
HISTORY= option
PROC MVPDIAGNOSE statement, 906, 915
PROC MVPMONITOR statement, 963, 975
HLLEGLABEL= option
PARETO procedure, 1102
HMINOR= option
CAPABILITY procedure, 541
CUSUM procedure, 2063
MACONTROL procedure, 2063
SHEWHART procedure, 2063
HOFFSET= option
CUSUM procedure, 2006
MACONTROL procedure, 2006
SHEWHART procedure, 2006
HREF2= option
CUSUM procedure, 2006
MACONTROL procedure, 2006
SHEWHART procedure, 2006
HREF2DATA= option
CUSUM procedure, 2006
MACONTROL procedure, 2006
SHEWHART procedure, 2006
HREF2LABELS= option
CUSUM procedure, 2007
MACONTROL procedure, 2007
SHEWHART procedure, 2007
HREF= option
CAPABILITY procedure, 535
CUSUM procedure, 2006
MACONTROL procedure, 2006
SHEWHART procedure, 2006
HREFCHAR= option
CUSUM procedure, 2070
MACONTROL procedure, 2070
PARETO procedure, 1117
SHEWHART procedure, 2070
HREFDATA= option
CUSUM procedure, 2007
MACONTROL procedure, 2007
SHEWHART procedure, 2007
HREFLABELS= option
CAPABILITY procedure, 535
CUSUM procedure, 2007
MACONTROL procedure, 2007
SHEWHART procedure, 2007
HREFLABPOS= option
CAPABILITY procedure, 535
CUSUM procedure, 2007
MACONTROL procedure, 2007
PARETO procedure, 1102
SHEWHART procedure, 2007
HTML2= option
SHEWHART procedure, 2063
HTML= option
CUSUM procedure, 2063
MACONTROL procedure, 2063
PARETO procedure, 1114
SHEWHART procedure, 2063
HTML_LEGEND= option
CUSUM procedure, 2063
MACONTROL procedure, 2063
SHEWHART procedure, 2063

ID statement
MVPDIAGNOSE procedure, 910

MVPMODEL procedure, 941
MVPMONITOR procedure, 965
RAREEVENTS procedure, 1177
ID statement, OPTEX procedure, *see* OPTEX procedure, ID statement
syntax, 1024
IDCOLOR= option
SHEWHART procedure, 2063
IDCTEXT= option
SHEWHART procedure, 2063
IDFONT= option
SHEWHART procedure, 2063
IDHEIGHT= option
SHEWHART procedure, 2063
IDSYMBOL= option
SHEWHART procedure, 2064
IDSYMBOLHEIGHT= option
SHEWHART procedure, 2064
INDEPENDENTZONES option
SHEWHART procedure, 2008
INFILLTRANSPARENCY= option
ANOM procedure, 2051
CUSUM procedure, 2051
MACONTROL procedure, 2051
SHEWHART procedure, 2051
INFONT= option
CAPABILITY procedure, 541
PARETO procedure, 1114
INHEIGHT= option
CAPABILITY procedure, 541
PARETO procedure, 1114
INSET and INSET2 statements
list of options, 1985
overview, 1975
syntax, 1981
INSET statement, *see* ANOM procedure, INSET statement, *see* CAPABILITY procedure, INSET statement
getting started, 168, 386, 1976
keywords summarized by function, 174, 392, 396, 399, 400, 1086, 1983
list of options, 175, 402, 1087
overview, 168, 385
syntax, 173, 390, 1085
INTERBAR= option
CAPABILITY procedure, 335
PARETO procedure, 1114
INTERTILE= option
CAPABILITY procedure, 535
PARETO procedure, 1103
INTERVAL= option
CUSUM procedure, 2008
MACONTROL procedure, 2008
SHEWHART procedure, 2008
INTERVALS statement, *see* CAPABILITY procedure, INTERVALS statement
getting started, 413
list of options, 417
overview, 413
syntax, 417
INTSTART= option
CUSUM procedure, 2009
MACONTROL procedure, 2009
SHEWHART procedure, 2009
IRCHART statement, *see also* SHEWHART procedure, IRCHART statement
examples, advanced, 1551
examples, introductory, 1519
options summarized by function, 1530
overview, 1518
syntax, 1529
Ishikawa diagrams
adding arrows, 720–724
aligning arrows, 741–748
balancing arrows, 741–748
data collection, 748, 749
data presentation, 748, 749
deleting arrows, 733–736
detail, decreasing, 750–752
detail, increasing, 750–752
editing existing diagrams, 777, 778
editing labels, 724–727
exporting diagrams, 761, 762
fonts, modifying, 763
highlighting arrows, 764, 766–772
isolating arrows, 753–755
labeling arrows, 724–727
managing complexity, 750–758
merging diagrams, 755–758
moving arrows, 727–733, 738–748
notepads, 748, 749
output, bitmaps, 761, 762
output, graphics, 758–760
output, SAS data set, 776, 782, 783
overview, 704
printing, bitmaps, 761, 762
printing, SAS/GRAPH output, 758–760
resizing arrows, 736–738
SAS data set, input, 777, 778, 782, 783
SAS data set, output, 776, 782, 783
saving, bitmaps, 761, 762
saving, clipboard graphics, 761, 762
saving, graphics, 758–760
saving, SAS data set, 776
subsetting arrows, 736–738, 764, 766–772
summary of operations, 717–720
swapping arrows, 738–741
tagging arrows, 736–738, 764, 766–772

terminology, 705
text entry, 724–727
undo, 733–736
zooming arrows, 752, 753, 774
ISHIKAWA procedure, 716
syntax, 716

K= option
CUSUM procedure, 580

L= option
CAPABILITY procedure, 541
LABELANGLE= option
MACONTROL procedure, 2064
SHEWHART procedure, 2064
LABELFONT= option
MACONTROL procedure, 2064
SHEWHART procedure, 2064, 2099
LABELHEIGHT= option
MACONTROL procedure, 2064
SHEWHART procedure, 2064
LABELS= option
SCOREMATRIX statement, 911
SCOREPLOT statement, 912
LABOTHER= option
PARETO procedure, 1103
LAST= option
PARETO procedure, 1103
LBOXES= option
SHEWHART procedure, 2064
LCATREF= option
PARETO procedure, 1115
LCLLABEL2= option
SHEWHART procedure, 2009
LCLLABEL= option
MACONTROL procedure, 2009
SHEWHART procedure, 2009
LDLLABEL= option
ANOM procedure, 184
LENDGRID= option
CUSUM procedure, 2064
MACONTROL procedure, 2064
SHEWHART procedure, 2064
LFREQREF= option
PARETO procedure, 1115
LGRID2= option
PARETO procedure, 1115
LGRID= option
CUSUM procedure, 2065
MACONTROL procedure, 2065
PARETO procedure, 1115
SHEWHART procedure, 2065
LHREF= option
CAPABILITY procedure, 541
CUSUM procedure, 2065
MACONTROL procedure, 2065
SHEWHART procedure, 2065
LIMITK= option
ANOM procedure, 184
LIMITN= option
ANOM procedure, 184
CUSUM procedure, 581
MACONTROL procedure, 818, 872
SHEWHART procedure, 2009, 2123
LIMITS= option
PROC RAREEVENTS statement, 1176, 1187
LIMLABSUBCHAR= option
ANOM procedure, 184
SHEWHART procedure, 2010
LINEPRINTER option
PARETO procedure, 1080
LLIMITS= option
ANOM procedure, 184
CUSUM procedure, 584
MACONTROL procedure, 2065
SHEWHART procedure, 2065
LMASK= option
CUSUM procedure, 584
LOADINGS= option
PROC MVPDIAGNOSE statement, 906, 916
PROC MVPMONITOR statement, 963, 976
LOTHER= option
PARETO procedure, 1103
LOVERLAY2= option
SHEWHART procedure, 2065
LOVERLAY= option
SHEWHART procedure, 2065
LSL= option
SHEWHART procedure, 2011
LSMEANS statement
RELIABILITY procedure, 1288
LSMESTIMATE statement
RELIABILITY procedure, 1290
LSTARCIRCLES= option
CUSUM procedure, 2065
MACONTROL procedure, 2065
SHEWHART procedure, 2065, 2094, 2099
LSTARS= option
CUSUM procedure, 2066
MACONTROL procedure, 2066
SHEWHART procedure, 2066
LSTATREF= option
CAPABILITY procedure, 541
LTESTS= option
SHEWHART procedure, 2066, 2134
LTMARGIN= option
SHEWHART procedure, 2011, 2078
LTMPLOT= option

SHEWHART procedure, 2011
LVREF= option
CAPABILITY procedure, 541
CUSUM procedure, 2066
MACONTROL procedure, 2066
SHEWHART procedure, 2066
LZONES= option
CUSUM procedure, 2066
MACONTROL procedure, 2066
SHEWHART procedure, 2066

MACHART statement, *see also* MACONTROL procedure, MACHART statement
examples, advanced, 889
examples, introductory, 849
overview, 848
syntax, 861
macontrol, 788
MACONTROL procedure, 791
ANNOTATE2= option, 791
ANNOTATE= option, 791
DATA= data set, 791
FORMCHAR= option, 791, 792
GOUT= option, 792
HISTORY= data set, 792, 793
INSET statement, 891
introduction, 788
LIMITS= data set, 793
LINEPRINTER option, 793
overview, 790
syntax, 791
TABLE= data set, 793
MACONTROL procedure, BY statement, 793
MACONTROL procedure, EWMACHART statement
ALLN option, 842
ALPHA= option, 817
ASYMPTOTIC option, 817, 837
CMEANSYMBOL= option, 818
DATA= data set, 829
HISTORY= data set, 799, 801, 802, 830, 831
LIMITN= option, 818, 841
LIMITS= data set, 805–807, 829, 830, 836
MEANCHAR= option, 818
MEANSYMBOL= option, 818, 846
missing values, 834
MU0= option, 818, 835, 837
NMARKERS option, 842
NOREADLIMITS option, 818
OUTHISTORY= data set, 802, 803, 826, 827
OUTLIMITS= data set, 803, 825, 826
OUTTABLE= data set, 804, 805, 827, 828
READALPHA option, 819
READINDEX= option, 819
READLIMITS option, 819
RESET option, 819
SIGMA0= option, 819, 835, 837
SIGMAS= option, 820
SMETHOD= option, 833, 843
TABLE= data set, 805, 831, 832
VREF= option, 846
WEIGHT= option, 797, 807, 820
XSYMBOL= option, 835
MACONTROL procedure, INSET statement, *see* INSET and INSET2 statements
getting started, 893
overview, 892
syntax, 894
MACONTROL procedure, MACHART statement
ALPHA= option, 871
ASYMPTOTIC option, 871
CMEANSYMBOL= option, 872
DATA= data set, 883, 884
HISTORY= data set, 853–855, 885
LIMITN= option, 872
LIMITS= data set, 837, 859, 860, 884, 890, 891
MEANCHAR= option, 872
MEANSYMBOL= option, 872
missing values, 888
MU0= option, 872, 889, 891
NOREADLIMITS option, 872
OUTHISTORY= data set, 855, 856, 880, 881
OUTLIMITS= data set, 856, 857, 879, 880
OUTTABLE= data set, 857, 858, 881, 882
READALPHA option, 873
READINDEX= option, 873
READLIMITS option, 873
SIGMA0= option, 873, 889, 891
SIGMAS= option, 873
SMETHOD= option, 887
SPAN= option, 851, 861, 874
TABLE= data set, 859, 886
XSYMBOL= option, 889
MARKERDISPLAY2= option
ANOM procedure, 2051
CUSUM procedure, 2051
MACONTROL procedure, 2051
SHEWHART procedure, 2051
MARKERDISPLAY= option
ANOM procedure, 2051
CUSUM procedure, 2051
MACONTROL procedure, 2051
SHEWHART procedure, 2051
MARKERLABEL2= option
ANOM procedure, 2052
CUSUM procedure, 2052
MACONTROL procedure, 2052
SHEWHART procedure, 2052
MARKERLABEL= option

ANOM procedure, 2052
CUSUM procedure, 2052
MACONTROL procedure, 2052
SHEWHART procedure, 2052
MARKERMISSINGGROUP= option
ANOM procedure, 2052
CUSUM procedure, 2052
MACONTROL procedure, 2052
SHEWHART procedure, 2052
MARKERS option
CUSUM procedure, 2053
MACONTROL procedure, 2053
PARETO procedure, 1103
SHEWHART procedure, 2053
MAXCMPCT= option
PARETO procedure, 1103
MAXITER= option
CAPABILITY procedure, 535
MAXNCAT= option
PARETO procedure, 1104
MAXNPLOTS= option
chart statement, 969
CONTRIBUTIONPANEL statement, 908
CONTRIBUTIONPLOT statement, 909
MAXNVARS= option
chart statement, 969
CONTRIBUTIONPANEL statement, 908
CONTRIBUTIONPLOT statement, 910
MAXPANELS= option
CUSUM procedure, 2011
MACONTROL procedure, 2011
SHEWHART procedure, 2011
MCHART statement, *see also* SHEWHART procedure, MCHART statement
examples, advanced, 1595
examples, introductory, 1560
options summarized by function, 1574
overview, 1559
syntax, 1573
MEAN= option
ANOM procedure, 184
MEANCHAR= option
MACONTROL procedure, 818, 872
MEANSYMBOL= option
MACONTROL procedure, 818, 872
MEDCENTRAL= option
SHEWHART procedure, 2011
MINPCT= option
PARETO procedure, 1104
MISSBREAK option
CUSUM procedure, 2012
MACONTROL procedure, 2012
SHEWHART procedure, 2012
MISSING option
PARETO procedure, 1104
missing subgroup variable values
SHEWHART procedure, 2012
MISSING1 option
PARETO procedure, 1105
MISSING2 option
PARETO procedure, 1105
MISSING= option
PROC MVPDIAGNOSE statement, 906
PROC MVPMODEL statement, 937
PROC MVPMONITOR statement, 963
MODEL statement, FACTEX procedure, *see* FACTEX procedure, MODEL statement
syntax, 634
MODEL statement, OPTEX procedure, *see* OPTEX procedure, MODEL statement
syntax, 1024
MRCHART statement, *see also* SHEWHART procedure, MRCHART statement
examples, advanced, 1638
examples, introductory, 1604
options summarized by function, 1617
overview, 1603
syntax, 1615
MRRESTART
SHEWHART procedure, 2012
MSE= option
ANOM procedure, 184
MU0= option
CUSUM procedure, 581
MACONTROL procedure, 818, 872
SHEWHART procedure, 2012, 2123, 2170
MVPDIAGNOSE procedure
syntax, 906
MVPDIAGNOSE procedure, BY statement, 907
MVPDIAGNOSE procedure, CONTRIBUTIONPANEL statement, 908
MAXNPLOTS= option, 908
MAXNVARS= option, 908
NCOLS= option, 909
NROWS= option, 909
TYPE= option, 909
MVPDIAGNOSE procedure, CONTRIBUTIONPLOT statement, 909
MAXNPLOTS= option, 909
MAXNVARS= option, 910
TYPE= option, 910
MVPDIAGNOSE procedure, ID statement, 910
MVPDIAGNOSE procedure, plot statement
ODSFOOTNOTE2= option, 913
ODSFOOTNOTE= option, 913
ODSTITLE2= option, 913
ODSTITLE= option, 913

MVPDIAGNOSE procedure, plot statement options, 913
MVPDIAGNOSE procedure, PROC MVPDIAGNOSE statement, 906
DATA= option, 906, 915
HISTORY= option, 906, 915
LOADINGS= option, 906, 916
MISSING= option, 906
PREFIX= option, 907
RPREFIX= option, 907
MVPDIAGNOSE procedure, SCOREMATRIX statement, 910
ALPHA= option, 911
ELLIPSE option, 911
GROUP= option, 911
LABELS= option, 911
NCOMP= option, 911
MVPDIAGNOSE procedure, SCOREPLOT statement, 911
ALPHA= option, 912
ELLIPSE option, 912
GROUP= option, 912
LABELS= option, 912
XCOMP= option, 912
YCOMP= option, 912
MVPDIAGNOSE procedure, TIME statement, 912
MVPMODEL procedure
syntax, 934
MVPMODEL procedure, BY statement, 941
MVPMODEL procedure, ID statement, 941
MVPMODEL procedure, PROC MVPMODEL statement, 935
COV option, 935
CV= option, 935
DATA= option, 937, 946
MISSING= option, 937
NCOMP= option, 937
NITEROBS= option, 936, 937
NOBS= option, 936
NOCENTER option, 937
NOCVSTDIZE option, 937
NOPRINT option, 937
NOSCALE option, 938
NTESTOBS= option, 937
NTESTVAR= option, 937
NVAR= option, 936
OUT= option, 938, 947
OUTLOADINGS= option, 938, 947
PLOTS= option, 938
PREFIX= option, 940
RPREFIX= option, 940
SEED= option, 937
STDSCORES option, 940
MVPMODEL procedure, VAR statement, 941
MVPMONITOR procedure
syntax, 962
MVPMONITOR procedure, BY statement, 964
MVPMONITOR procedure, chart statement
ALPHA= option, 969
CONTRIBUTIONS option, 969
EXCHART option, 969
MAXNPLOTS= option, 969
MAXNVARS= option, 969
NOHLABEL option, 970
NOVLABEL option, 970
NPANELPOS= option, 971
ODSFOOTNOTE2= option, 971
ODSFOOTNOTE= option, 971
ODSTITLE2= option, 971
ODSTITLE= option, 971
OUTTABLE= option, 972
OVERLAY option, 972
SERIESVALUE= option, 972
TOTPANELS= option, 972
MVPMONITOR procedure, chart statement options, 969
MVPMONITOR procedure, ID statement, 965
MVPMONITOR procedure, PROC MVPMONITOR statement, 963
DATA= option, 963, 975
HISTORY= option, 963, 975
LOADINGS= option, 963, 976
MISSING= option, 963
OUTHISTORY= option, 963, 978
OUTTABLE= option, 979
PREFIX= option, 963
RPREFIX= option, 964
TABLE= option, 964, 977
MVPMONITOR procedure, SCORECHART statement, 965
COMP= option, 966
SIGMAS= option, 966
MVPMONITOR procedure, SERIES statement, 965
MVPMONITOR procedure, SPECHART statement, 967
NOCHART option, 970
MVPMONITOR procedure, TIME statement, 967
MVPMONITOR procedure, TSQUARECHART statement, 968

NAME2= option
SHEWHART procedure, 2066
NAME= option
CAPABILITY procedure, 541
CUSUM procedure, 2066
MACONTROL procedure, 2066
PARETO procedure, 1115
SHEWHART procedure, 2066

NBINS= option
COMPARE statement, 1179
NCOLS= option
CAPABILITY procedure, 536
CONTRIBUTIONPANEL statement, 909
PARETO procedure, 1105
NCOMP= option
PROC MVPMODEL statement, 937
SCOREMATRIX statement, 911
NDECIMAL2= option
SHEWHART procedure, 2012
NDECIMAL= option
ANOM procedure, 184
MACONTROL procedure, 2012
SHEWHART procedure, 2012
NEEDLES option
CUSUM procedure, 2012
MACONTROL procedure, 2012
SHEWHART procedure, 2012
NITEROBS= option
PROC MVPMODEL statement, 936, 937
NLEGEND= option
PARETO procedure, 1105
NMARKERS option
CUSUM procedure, 2013
MACONTROL procedure, 2013
SHEWHART procedure, 2013
NO3SIGMACHECK option
SHEWHART procedure, 2013
NOARL option
CUSUM procedure, 581
NOBLOCKREF option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOBLOCKREFFILL option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOBOXFILLLEGEND option
ANOM procedure, 2053
SHEWHART procedure, 2053
NOBS= option
PROC MVPMODEL statement, 936
NOBYREF option
CUSUM procedure, 2013
MACONTROL procedure, 2013
SHEWHART procedure, 2013
NOCATLABEL option
PARETO procedure, 1106
NOCENTER option
PROC MVPMODEL statement, 937
NOCHART option
CHART statement, 1178
CUSUM procedure, 2013
MACONTROL procedure, 2013
PARETO procedure, 1106
SHEWHART procedure, 2013
SPECHART statement, 970
NOCHART2 option
SHEWHART procedure, 2013
NOCONNECT option
CUSUM procedure, 2013
MACONTROL procedure, 2013
SHEWHART procedure, 2013
NOCTL option
ANOM procedure, 184
MACONTROL procedure, 2014
SHEWHART procedure, 2014
NOCTL2 option
SHEWHART procedure, 2014
NOCUMLABEL option
PARETO procedure, 1106
NOCURVE option
PARETO procedure, 1106
NOCVSTDIZE option
PROC MVPMODEL statement, 937
NOFILLLEGEND option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOFRAME option
CAPABILITY procedure, 541
CUSUM procedure, 2066
MACONTROL procedure, 2066
PARETO procedure, 1115
SHEWHART procedure, 2066
NOFREQLABEL option
PARETO procedure, 1106
NOFREQTICK option
PARETO procedure, 1106
NOHLABEL option
CAPABILITY procedure, 536
CHART statement, 1181
chart statement, 970
COMPARE statement, 1181
CUSUM procedure, 2014
MACONTROL procedure, 2014
SHEWHART procedure, 2014
NOHLLEG option
PARETO procedure, 1106
NOKEYMOVE option
PARETO procedure, 1106
NOLCL option
MACONTROL procedure, 2014

SHEWHART procedure, 2014
NOLCL2 option
SHEWHART procedure, 2014
NOLDL option
ANOM procedure, 184
NOLEGEND option
CUSUM procedure, 2014
MACONTROL procedure, 2014
SHEWHART procedure, 2014, 2076–2079, 2153, 2155
NOLIMIT0 option
ANOM procedure, 184
SHEWHART procedure, 2014
NOLIMIT1 option
ANOM procedure, 184
SHEWHART procedure, 2014
NOLIMITLABEL option
ANOM procedure, 184
MACONTROL procedure, 2014
SHEWHART procedure, 2014
NOLIMITS option
ANOM procedure, 184
MACONTROL procedure, 2014
SHEWHART procedure, 2014, 2153
NOLIMITSFRAME option
ANOM procedure, 185
SHEWHART procedure, 2066
NOLIMITSLEGEND option
ANOM procedure, 185
MACONTROL procedure, 2014
SHEWHART procedure, 2014
NOMASK option
CUSUM procedure, 581
NONEEDLES option
ANOM procedure, 185
NOOVERLAYLEGEND option
SHEWHART procedure, 2015
NOPHASEFRAME option
SHEWHART procedure, 2067
NOPHASEREF option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOPHASEREFFILL option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOPRINT option
PROC MVPMODEL statement, 937
NOREADLIMITS option
ANOM procedure, 185
CUSUM procedure, 581
MACONTROL procedure, 818, 872
SHEWHART procedure, 2015
NOREF option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOREFFILL option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOSCALE option
PROC MVPMODEL statement, 938
NOSTARFILLLEGEND option
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOTCHES option
SHEWHART procedure, 2015
NOTESTACROSS option
SHEWHART procedure, 2016
NOTICKREP option
SHEWHART procedure, 2016
NOTRANSPARENCY option
ANOM procedure, 2053
CUSUM procedure, 2053
MACONTROL procedure, 2053
SHEWHART procedure, 2053
NOTRENDCONNECT option
CUSUM procedure, 2016
MACONTROL procedure, 2016
SHEWHART procedure, 2016
NOTRUNC option
SHEWHART procedure, 2016
NOUCL option
MACONTROL procedure, 2017
SHEWHART procedure, 2017
NOUCL2 option
SHEWHART procedure, 2017
NOUDL option
ANOM procedure, 185
NOV2LABEL option
CUSUM procedure, 2067
MACONTROL procedure, 2067
SHEWHART procedure, 2067
NOVANGLE option
CUSUM procedure, 2067
MACONTROL procedure, 2067
SHEWHART procedure, 2067
NOVLABEL option
CAPABILITY procedure, 536
CHART statement, 1181
chart statement, 970

COMPARE statement, 1181
CUSUM procedure, 2067
MACONTROL procedure, 2067
SHEWHART procedure, 2067
NOVTICK option
CAPABILITY procedure, 536
NOVTICK2 option
PARETO procedure, 1106
NPANELPOS= option
CHART statement, 1178
chart statement, 971
CUSUM procedure, 2017
MACONTROL procedure, 2017
SHEWHART procedure, 2017
NPCHART statement, SHEWHART procedure, *see also* SHEWHART procedure, NPCHART statement
examples, advanced, 1676
examples, introductory, 1647
options summarized by function, 1658
overview, 1646
syntax, 1656
NPSYMBOL= option
SHEWHART procedure, 2017
NROWS= option
CAPABILITY procedure, 536
CONTRIBUTIONPANEL statement, 909
PARETO procedure, 1106
NTESTOBS= option
PROC MVPMODEL statement, 937
NTESTVAR= option
PROC MVPMODEL statement, 937
NVAR= option
PROC MVPMODEL statement, 936

ODSFOOTNOTE option
PARETO procedure, 1106
ODSFOOTNOTE2 option
PARETO procedure, 1107
ODSFOOTNOTE2= option
ANOM procedure, 2054
CHART statement, 1182
chart statement, 971
COMPARE statement, 1182
CUSUM procedure, 2054
MACONTROL procedure, 2054
plot statement, 913
SHEWHART procedure, 2054
ODSFOOTNOTE= option
ANOM procedure, 2054
CHART statement, 1181
chart statement, 913, 971
COMPARE statement, 1181
CUSUM procedure, 2054
MACONTROL procedure, 2054
SHEWHART procedure, 2054
ODSLEGENDEXPAND option
ANOM procedure, 2054
CUSUM procedure, 2054
MACONTROL procedure, 2054
SHEWHART procedure, 2054
ODSTITLE option
PARETO procedure, 1107
ODSTITLE2 option
PARETO procedure, 1107
ODSTITLE2= option
ANOM procedure, 2055
CHART statement, 1182
chart statement, 971
COMPARE statement, 1182
CUSUM procedure, 2055
MACONTROL procedure, 2055
plot statement, 913
SHEWHART procedure, 2055
ODSTITLE= option
ANOM procedure, 2054
CHART statement, 1182
chart statement, 971
COMPARE statement, 1182
CUSUM procedure, 2054
MACONTROL procedure, 2054
plot statement, 913
SHEWHART procedure, 2054
OPTEX procedure, 1010
getting started, 1001
learning about, 1000
order of statements, 1010, 1015, 1024, 1058
overview, 999
summary of functions, 1010
syntax, 1010
OPTEX procedure, BLOCKS statement
COVAR= option, 1014, 1061
DESIGN= option, 1014, 1057
INIT= option, 1015
ITER= option, 1015
KEEP= option, 1015
NOEXCHANGE option, 1015
options summarized by function, 1011
STRUCTURE= option, 1014, 1055
VAR= option, 1061
OPTEX procedure, CLASS statement
DESCENDING option, 1016
example, 1002
ORDER= option, 1016
PARAM= option, 1016
REF= option, 1019
syntax, 1015
TRUNCATE option, 1019

OPTEX procedure, EXAMINE statement
DESIGN option, 1019
INFORMATION option, 1020
NUMBER= option, 1020
VARIANCE option, 1020
OPTEX procedure, GENERATE statement
AUGMENT= option, 1021, 1049, 1050
CRITERION= option, 1021, 1064
INITDESIGN= option, 1021, 1048
ITER= option, 1022
KEEP= option, 1023
METHOD= option, 1023, 1046
N= option, 1007, 1023, 1048
OPTEX procedure, ID statement, 1024
OPTEX procedure, MODEL statement
example, 1002
NOINT option, 1024, 1063
PRIOR= option, 1024, 1053
OPTEX procedure, OUTPUT statement
BLOCKNAME= option, 1025
NUMBER= option, 1025, 1043
OUT= option, 1025
OPTEX procedure, PROC OPTEX statement
CODING= option, 1013, 1057
DATA= option, 1013
EPSILON= option, 1013
example, 1002
NAMELEN option, 1013
NOCODE option, 1013, 1063
NOPRINT option, 1013
ODS tables, 1042
options summarized by function, 1011
SEED= option, 1013
STATUS= option, 1014
ORDER1= option
PARETO procedure, 1108
ORDER2= option
PARETO procedure, 1108
ORDER= option
CLASS statement (OPTEX), 1016
ORIGIN= option
CUSUM procedure, 581
OTHER= option
PARETO procedure, 1108
OTHERCVAL= option
PARETO procedure, 1109
OTHERNVAL= option
PARETO procedure, 1109
OUT= data set
PARETO procedure, 1109
OUT= option
PROC MVPMODEL statement, 938, 947
OUTBOX= option
SHEWHART procedure, 2017
OUTFILL option
ANOM procedure, 2018
CUSUM procedure, 2018
MACONTROL procedure, 2018
SHEWHART procedure, 2018
OUTFILLTRANSPARENCY= option
ANOM procedure, 2055
CUSUM procedure, 2055
MACONTROL procedure, 2055
SHEWHART procedure, 2055
OUTHIGHHTML= option
SHEWHART procedure, 2067
OUTHIGHURL= option
SHEWHART procedure, 2055
OUTHISTORY= option
CUSUM procedure, 2018
MACONTROL procedure, 2018
PROC MVPMONITOR statement, 963, 978
SHEWHART procedure, 2018
OUTINDEX= option
CUSUM procedure, 2018
MACONTROL procedure, 2018
SHEWHART procedure, 2018
OUTLABEL2= option
SHEWHART procedure, 2018
OUTLABEL= option
CUSUM procedure, 2018
MACONTROL procedure, 2018
SHEWHART procedure, 2018
OUTLIMITS= option
CHART statement, 1178
CUSUM procedure, 2019
MACONTROL procedure, 2019
PROC RAREEVENTS statement, 1189
SHEWHART procedure, 2019
OUTLOADINGS= option
PROC MVPMODEL statement, 938, 947
OUTLOWHTML= option
SHEWHART procedure, 2067
OUTLOWURL= option
SHEWHART procedure, 2055
OUTPHASE= option
CUSUM procedure, 2019
MACONTROL procedure, 2019
SHEWHART procedure, 2019
OUTPUT statement, CAPABILITY procedure, *see* CAPABILITY procedure, OUTPUT statement
getting started, 424
keywords summarized by function, 427
overview, 424
syntax, 427
OUTPUT statement, FACTEX procedure, *see* FACTEX procedure, OUTPUT statement

syntax, 636
OUTPUT statement, OPTEX procedure, *see* OPTEX procedure, OUTPUT statement
syntax, 1025
OUTSUMMARY= option
ANOM procedure, 185
OUTTABLE= option
CHART statement, 1178
chart statement, 972
CUSUM procedure, 2019
MACONTROL procedure, 2019
PROC MVPMONITOR statement, 979
PROC RAREEVENTS statement, 1190
SHEWHART procedure, 2019
OVERLAY option
CAPABILITY procedure, 537
chart statement, 972
OVERLAY2= option
SHEWHART procedure, 2019
OVERLAY2HTML= option
SHEWHART procedure, 2067
OVERLAY2ID= option
SHEWHART procedure, 2020
OVERLAY2SYM= option
SHEWHART procedure, 2067
OVERLAY2SYMHT= option
SHEWHART procedure, 2067
OVERLAY2URL= option
SHEWHART procedure, 2055
OVERLAY= option
SHEWHART procedure, 2019
OVERLAYCLIPSYM= option
SHEWHART procedure, 2067
OVERLAYCLIPSYMHT= option
SHEWHART procedure, 2067
OVERLAYHTML= option
SHEWHART procedure, 2067
OVERLAYID= option
SHEWHART procedure, 2020
OVERLAYLEGLAB= option
SHEWHART procedure, 2020
OVERLAYSYM= option
SHEWHART procedure, 2068
OVERLAYSYMHT= option
SHEWHART procedure, 2068
OVERLAYURL= option
SHEWHART procedure, 2055

P0= option
SHEWHART procedure, 2020
P= option
ANOM procedure, 185
PAGENUM= option
CUSUM procedure, 2020
MACONTROL procedure, 2020
SHEWHART procedure, 2020
PAGENUMPOS= option
CUSUM procedure, 2020
MACONTROL procedure, 2020
SHEWHART procedure, 2020
PARAM= option
CLASS statement (OPTEX), 1016
PARETO procedure, 1078
examples, advanced, 1130
examples, introductory, 1069
options summarized by function, 1079
overview, 1068
syntax, 1078
PARETO procedure, BY statement, 1080, 1130, 1131
PARETO procedure, HBAR statement
ANCHOR= option, 1096
ANGLE= option, 1110
ANNOKEY option, 1110
ANNOTATE2= data set, 1110
ANNOTATE= data set, 1110
AXISFACTOR= option, 1096
BARLABEL= option, 1096
BARLABPOS= option, 1111
BARLEGEND= option, 1097
BARLEGLABEL= option, 1097
BARS= option, 1098
BARWIDTH= option, 1111
CATLEGEND= option, 1098
CATLEGLABEL= option, 1098
CATOFFSET= option, 1098
CATREF= option, 1098
CATREFLABELS= option, 1098
CAXIS2= option, 1111
CAXIS= option, 1111
CBARLINE= option, 1112
CBARS= option, 1112
CCATREF= option, 1112
CCONNECT= option, 1112
CCUMREF= option, 1112
CFRAME= option, 1112
CFRAMENLEG= option, 1098
CFRAMESIDE= option, 1112
CFRAMETOP= option, 1113
CFREQREF= option, 1113
CGRID2= option, 1113
CGRID= option, 1113
CHARTTYPE= option, 1099, 1154
CHIGH(*n*)= option, 1099
CLASS= option, 1099
CLASSKEY= option, 1100
CLIPREF option, 1113
CLOW(*n*)= option, 1100
CMPCTLABEL option, 1101

COTHER= option, 1113
CPROP= option, 1101
CTEXT= option, 1113
CTEXTSIDE= option, 1113
CTEXTTOP= option, 1113
CTILES= option, 1113
CUMAXIS= option, 1101
CUMAXISLABEL= option, 1101
CUMREF= option, 1101
CUMREFLABELS= option, 1101
DESCRIPTION= option, 1114
FONT= option, 1114
FREQ= option, 1073, 1074, 1101
FREQAXIS= option, 1102
FREQAXISLABEL= option, 1102
FREQOFFSET= option, 1102
FREQREF= option, 1102
FREQREFLABELS= option, 1102
FRONTREF option, 1114
GRID option, 1102
GRID2 option, 1102
HAXIS2= option, 1101
HAXIS2LABEL= option, 1101
HEIGHT= option, 1114
HLLEGLABEL= option, 1102
HREFLABPOS= option, 1102
HTML= option, 1114
INFONT= option, 1114
INHEIGHT= option, 1114
INTERBAR= option, 1114
INTERTILE= option, 1103
LABOTHER= option, 1103
LAST= option, 1073, 1074, 1103
LCATREF= option, 1115
LCUMREF= option, 1115
LFREQREF= option, 1115
LGRID2= option, 1115
LGRID= option, 1115
LOTHER= option, 1103
MARKERS option, 1073, 1074, 1103
MAXCMPCT= option, 1103
MAXNCAT= option, 1074, 1077, 1104
MINPCT= option, 1104
MISSING option, 1104
MISSING1 option, 1105
MISSING2 option, 1105
NAME= option, 1115
NCOLS= option, 1105
NLEGEND option, 1105
NLEGEND= option, 1073, 1074, 1105
NOCATLABEL option, 1106
NOCHART option, 1106
NOCUMLABEL option, 1106
NOCUMTICK option, 1106
NOCURVE option, 1106
NOFRAME option, 1115
NOFREQLABEL option, 1106
NOFREQTICK option, 1106
NOHLLEG option, 1106
NOKEYMOVE option, 1106
NROWS= option, 1106
ODSFOOTNOTE2= option, 1107
ODSFOOTNOTE= option, 1106
ODSTITLE2= option, 1107
ODSTITLE= option, 1073, 1074, 1107
options summarized by function, 1081
ORDER1= option, 1108
ORDER2= option, 1108
OTHER= option, 1074, 1077, 1103, 1104, 1108
OTHERCVAL= option, 1109
OTHERNVAL= option, 1109
OUT= option, 1109
PBARS= option, 1115
PHIGH(*n*)= option, 1116
PLOW(*n*)= option, 1116
POTHER= option, 1116
SCALE= option, 1073, 1074, 1109
syntax, 1081
TILELEGEND= option, 1116
TILELEGLABEL= option, 1116
URL= option, 1109
WAXIS= option, 1116
WBARLINE= option, 1117
WEIGHT= option, 1110
WGRID2= option, 1117
WGRID= option, 1117
PARETO procedure, INSET statement
CFILL= option, 1089
CFILLH= option, 1089
CFRAME= option, 1089
CHEADER= option, 1089
CSHADOW= option, 1089
CTEXT= option, 1090
DATA option, 1090
FONT= option, 1090
FORMAT= option, 1088
GUTTER= option, 1089
HEADER= option, 1088
HEIGHT= option, 1090
NCOLS= option, 1089
NOFRAME option, 1088
POSITION= option, 1088, 1121, 1123
REFPOINT= option, 1090
PARETO procedure, PROC PARETO statement, 1079
ANNOTATE2= data set, 1079
ANNOTATE= data set, 1079
DATA= data set, 1080
FORMCHAR= option, 1080

GOUT= option, 1080
LINEPRINTER option, 1080
PARETO procedure, VBAR statement
ANCHOR= option, 1096, 1120, 1136, 1137
ANGLE= option, 1110
ANNOKEY option, 1110
ANNOTATE2= data set, 1110
ANNOTATE= data set, 1110
AXISFACTOR= option, 1096, 1120
BARLABEL= option, 1096
BARLABPOS= option, 1111
BARLEGEND= option, 1097, 1143, 1144
BARLEGLABEL= option, 1097
BARS= option, 1098, 1144
BARWIDTH= option, 1111
CATLEGEND= option, 1098
CATLEGLABEL= option, 1098, 1137, 1138
CATOFFSET= option, 1098
CATREF= option, 1098
CATREFLABELS= option, 1098
CAXIS2= option, 1111
CAXIS= option, 1111
CBARLINE= option, 1112
CBARS= option, 1112, 1136, 1143
CCATREF= option, 1112
CCONNECT= option, 1112
CFRAME= option, 1112
CFRAMENLEG option, 1134
CFRAMENLEG= option, 1098
CFRAMESIDE= option, 1112
CFRAMETOP= option, 1113
CFRQREF= option, 1112
CGRID2= option, 1113
CGRID= option, 1113
CHARTTYPE= option, 1099, 1154
CHIGH(*n*) option, 1142
CHIGH(*n*)= option, 1141
CHIGH(*n*)= option, 1099
CLASS= option, 1099, 1129, 1132, 1137–1140
CLASSKEY= option, 1100, 1132
CLIPREF option, 1113
CLOW(*n*)= option, 1141
CLOW(*n*)= option, 1100
CMPCTLABEL option, 1096, 1101
CONNECTCHAR= option, 1117
COTHER= option, 1113
CPROP option, 1134
CPROP= option, 1101
CTEXT= option, 1113
CTEXTSIDE= option, 1113
CTEXTTOP= option, 1113
CTILES= option, 1113, 1144, 1145
CUMAXIS= option, 1101
CUMAXISLABEL= option, 1101
CUMREF= option, 1101
CUMREFLABELS= option, 1101
DESCRIPTION= option, 1114
FONT= option, 1114
FREQ= option, 1101
FREQAXIS= option, 1102
FREQAXISLABEL= option, 1102
FREQOFFSET= option, 1102
FREQREF= option, 1102, 1139
FREQREFLABELS= option, 1102
FRONTREF option, 1114
GRID option, 1102
GRID2 option, 1102
HEIGHT= option, 1114
HLLEGLABEL= option, 1102
HREFCHAR= option, 1117
HREFLABPOS= option, 1102
HTML= option, 1114
INFONT= option, 1114
INHEIGHT= option, 1114
INTERBAR= option, 1114
INTERTILE= option, 1103, 1134
LABOTHER= option, 1103
LAST= option, 1103
LCATREF= option, 1115
LCUMREF= option, 1115
LFREQREF= option, 1115
LGRID2= option, 1115
LGRID= option, 1115
LOTHER= option, 1103
MARKERS option, 1103
MAXCMPCT= option, 1103
MAXNCAT= option, 1104
MINPCT= option, 1104
MISSING option, 1104, 1129
MISSING1 option, 1105, 1129
MISSING2 option, 1105, 1129
NAME= option, 1115
NCOLS= option, 1105, 1119, 1139, 1140
NLEGEND option, 1105, 1136, 1137
NLEGEND= option, 1105, 1134
NOCATLABEL option, 1106, 1137, 1138
NOCHART option, 1106
NOCUMLABEL option, 1106
NOCUMTICK option, 1106
NOCURVE option, 1106, 1128, 1137, 1138
NOFRAME option, 1115
NOFREQLABEL option, 1106
NOFREQTICK option, 1106
NOHLLEG option, 1106
NOKEYMOVE option, 1106
NROWS= option, 1106, 1119, 1138–1140
ODSFOOTNOTE2= option, 1107
ODSFOOTNOTE= option, 1106

ODSTITLE2= option, 1107
ODSTITLE= option, 1107
options summarized by function, 1091
ORDER1= option, 1108, 1129
ORDER2= option, 1108, 1129
OTHER= option, 1103, 1104, 1108
OTHERCVAL= option, 1109, 1127
OTHERNVAL= option, 1109, 1127
OUT= data set, 1126, 1127
OUT= option, 1109
PBARS= option, 1115
PHIGH(*n*) option, 1142
PHIGH(*n*)= option, 1141
PHIGH(*n*)= option, 1116
PLOW(*n*)= option, 1116
POTHER= option, 1116
SCALE= option, 1109, 1128, 1134
SYMBOLCHAR= option, 1117
syntax, 1090
TILELEGEND= option, 1116, 1144, 1145
TILELEGLABEL= option, 1116
TURNVLABEL option, 1116
URL= option, 1109
VAXIS2= option, 1101
VAXIS2LABEL= option, 1101
VREFCHAR= option, 1117
VREFLABPOS= option, 1110
WAXIS= option, 1116
WBARLINE= option, 1117
WEIGHT= option, 1110, 1152, 1153
WGRID2= option, 1117
WGRID= option, 1117
PATTERN statement, 360
PBARS= option
PARETO procedure, 1115
PCHART statement, ANOM procedure, *see also* ANOM procedure, PCHART statement
examples, advanced, 103
examples, introductory, 78
options summarized by function, 86
overview, 77
syntax, 85
PCHART statement, SHEWHART procedure, *see also* SHEWHART procedure, PCHART statement
examples, advanced, 1716
examples, introductory, 1687
options summarized by function, 1699
overview, 1686
syntax, 1697
PCTLDEF= option
SHEWHART procedure, 2021
_PHASE_variables
SHEWHART procedure, 2079
PHASEBOXLABELS option
ANOM procedure, 2055
CUSUM procedure, 2055
MACONTROL procedure, 2055
SHEWHART procedure, 2055
PHASEBREAK option
CUSUM procedure, 2021
MACONTROL procedure, 2021
SHEWHART procedure, 2021, 2131, 2132, 2166
PHASELABTYPE= option
CUSUM procedure, 2021
MACONTROL procedure, 2021
SHEWHART procedure, 2021
PHASELEGEND option
CUSUM procedure, 2021
MACONTROL procedure, 2021
SHEWHART procedure, 2021, 2079, 2083–2085, 2087, 2088
PHASELIMITS option
CUSUM procedure, 2021
MACONTROL procedure, 2021
SHEWHART procedure, 2021
PHASEMEANSYMBOL= option
SHEWHART procedure, 1477, 2021
PHASEPOS= option
CUSUM procedure, 2055
MACONTROL procedure, 2055
SHEWHART procedure, 2055
PHASEREF option
CUSUM procedure, 2021
MACONTROL procedure, 2021
SHEWHART procedure, 2021, 2079, 2083–2085, 2087, 2088
PHASEREFLEVEL= option
ANOM procedure, 2056
CUSUM procedure, 2056
MACONTROL procedure, 2056
SHEWHART procedure, 2056
PHASEREFTRANSPARENCY= option
ANOM procedure, 2051
CUSUM procedure, 2051
MACONTROL procedure, 2051
SHEWHART procedure, 2051
phases of subgroups
SHEWHART procedure, 2003
PHASEVALSEP option
CUSUM procedure, 2022
MACONTROL procedure, 2022
SHEWHART procedure, 2022
PHASEVARLABEL option
CUSUM procedure, 2021
MACONTROL procedure, 2021
SHEWHART procedure, 2021
PHIGH(*n*)= option

PARETO procedure, 1116
plot statement options
MVPDIAGNOSE procedure, 913
PLOTS= option
PROC MVPMODEL statement, 938
PLOW(*n*)= option
PARETO procedure, 1116
POINTSHTML= option
SHEWHART procedure, 2068
POINTSURL= option
SHEWHART procedure, 2056
POTHER= option
PARETO procedure, 1116
PPLOT statement, *see* RELIABILITY procedure, PROBPLOT statement
PPPLOT statement, *see* CAPABILITY procedure, PPPLOT statement
getting started, 440
options dictionary, 447
options summarized by function, 443, 445
overview, 439
syntax, 442
PREFIX= option
PROC MVPDIAGNOSE statement, 907
PROC MVPMODEL statement, 940
PROC MVPMONITOR statement, 963
PROBACC2 function, 2229, 2230, 2239
PROBBNML function, 2230, 2231, 2237, 2238
PROBHYPR function, 2232–2234, 2237
PROBLIMITS= option
SHEWHART procedure, 2022
PROBMED function, 2234, 2235
PROBPLOT statement, *see* CAPABILITY procedure, PROBPLOT statement, *see* RELIABILITY procedure, PROBPLOT statement
getting started, 462
options summarized by function, 469–471
overview, 461
syntax, 468
PROC CAPABILITY statement
examples, 248
getting started, 197
options summarized by function, 202
overview, 195
syntax, 201
CAPABILITY procedure, *see* PROC CAPABILITY statement
PROC FACTEX statement, *see* FACTEX procedure, PROC FACTEX statement
syntax, 629
PROC MVPDIAGNOSE statement, 906, *see* MVPDIAGNOSE procedure
PROC MVPMODEL statement, 935, *see* MVPMODEL procedure
PROC MVPMONITOR statement, 963, *see* MVPMONITOR procedure
PROC OPTEX statement, *see* OPTEX procedure, PROC OPTEX statement
syntax, 1012
PROC PARETO statement, 1079, *see* PARETO procedure
PROC RAREEVENTS statement, 1176, *see* RAREEVENTS procedure
PROC SHEWHART statement
options summarized by function, 1412
PROCESS= option
COMPARE statement, 1180
processes, CUSUM procedure
XCHART statement, 570
processes, MACONTROL procedure
EWMACHART statement, 807
MACHART statement, 861
processes, SHEWHART procedure
BOXCHART statement, 1433
CCHART statement, 1492
IRCHART statement, 1529
MCHART statement, 1573
MRCHART statement, 1616
NPCHART statement, 1657
PCHART statement, 1698
RCHART statement, 1742
SCHART statement, 1778
UCHART statement, 1812
XCHART statement, 1851
XRCHART statement, 1895
XSCHART statement, 1937
PSYMBOL= option
ANOM procedure, 185
SHEWHART procedure, 2023

qc, 7
QQPLOT statement, *see* CAPABILITY procedure, QQPLOT statement
getting started, 494
options summarized by function, 498–500, 502
overview, 493
syntax, 497

RANGES option
SHEWHART procedure, 2023
RAREEVENTS procedure
syntax, 1175
RAREEVENTS procedure, BY statement, 1176
RAREEVENTS procedure, CHART statement, 1177
ALPHALPL= option, 1178
ALPHAUPL= option, 1178
DIST= option, 1180
EXCHART option, 1178

NOCHART option, 1178
NOHLABEL option, 1181
NOVLABEL option, 1181
NPANELPOS= option, 1178
ODSFOOTNOTE2= option, 1182
ODSFOOTNOTE= option, 1181
ODSTITLE2= option, 1182
ODSTITLE= option, 1182
OUTLIMITS= option, 1178
OUTTABLE= option, 1178
TOTPANELS= option, 1179
RAREEVENTS procedure, CHART statement options, 1177, 1180
RAREEVENTS procedure, COMPARE statement, 1179
DIST= option, 1180
NBINS= option, 1179
NOHLABEL option, 1181
NOVLABEL option, 1181
ODSFOOTNOTE2= option, 1182
ODSFOOTNOTE= option, 1181
ODSTITLE2= option, 1182
ODSTITLE= option, 1182
PROCESS= option, 1180
REFERENCE= option, 1180
RAREEVENTS procedure, COMPARE statement options, 1180
RAREEVENTS procedure, ID statement, 1177
RAREEVENTS procedure, PROC RAREEVENTS statement, 1176
DATA= option, 1176, 1187
LIMITS= option, 1176, 1187
OUTLIMITS= option, 1189
OUTTABLE= option, 1190
TABLE= option, 1176, 1189
RCHART statement, *see also* SHEWHART procedure, RCHART statement
examples, advanced, 1761
examples, introductory, 1730
options summarized by function, 1743
overview, 1729
syntax, 1742
READALPHA option
MACONTROL procedure, 819, 873
SHEWHART procedure, 2023
READINDEX= option
CUSUM procedure, 582
MACONTROL procedure, 819, 873
SHEWHART procedure, 2023, 2081, 2083–2085, 2087–2089, 2131–2133
READINDEXES= option
ANOM procedure, 185
READLIMITS option
CUSUM procedure, 582
MACONTROL procedure, 819, 873
SHEWHART procedure, 2024
READPHASES= option
CUSUM procedure, 2024
MACONTROL procedure, 2024
SHEWHART procedure, 2024, 2079–2081, 2083–2085, 2087–2089, 2129, 2131–2133
READSIGMAS option
CUSUM procedure, 582
REF= option
CLASS statement (OPTEX), 1019
REFERENCE= option
COMPARE statement, 1180
REFFILLTRANSPARENCY= option
ANOM procedure, 2051
CUSUM procedure, 2051
MACONTROL procedure, 2051
SHEWHART procedure, 2051
reliability, 1204
RELIABILITY procedure, 1271
details, 1340
examples, 1206
overview, 1204
syntax, 1271
RELIABILITY procedure, ANALYZE statement, 1261, 1273, 1275, 1278
CONVERGE= option, 1355
PPOS= option, 1344, 1345, 1347
PREDICT option, 1261
summary of options, 1275, 1278
TOLERANCE option, 1261
RELIABILITY procedure, BY statement, 1271, 1272, 1279
RELIABILITY procedure, CLASS statement, 1271, 1272, 1279
TRUNCATE option, 1279
RELIABILITY procedure, DISTRIBUTION statement, 1206, 1207, 1216, 1217, 1221, 1222, 1231, 1261, 1271, 1272, 1280
RELIABILITY procedure, EFFECTPLOT statement, 1281
RELIABILITY procedure, ESTIMATE statement, 1282
RELIABILITY procedure, FMODE statement, 1271, 1272, 1283
RELIABILITY procedure, FREQ statement, 1216, 1217, 1221, 1222, 1271, 1272, 1284
RELIABILITY procedure, INSET statement, 1271, 1272, 1284, 1285, 1287
keywords, 1285
summary of options, 1287
RELIABILITY procedure, LSMEANS statement, 1288

RELIABILITY procedure, LSMESTIMATE statement, 1290
RELIABILITY procedure, MAKE statement, 1231, 1271, 1272, 1291
  NOPRINT option, 1291
  table keywords, 1291
RELIABILITY procedure, MCFPLOT statement, 1247, 1250, 1252, 1271, 1291, 1293, 1296, 1298, 1302
  MCFDIFF option, 1252
  summary of options, 1293, 1296, 1298, 1302
RELIABILITY procedure, MODEL statement, 1216, 1217, 1231, 1271, 1302, 1304, 1308
  CONVERGE= option, 1355
  CORRB option, 1231
  COVB option, 1231
  No intercept option, 1357
  OBSTATS option, 1216, 1217, 1231, 1308
  RELATION= option, 1216, 1217
  summary of options, 1304, 1308
RELIABILITY procedure, NENTER statement, 1221, 1222, 1271, 1272, 1310
RELIABILITY procedure, NLOPTIONS statement, 1310
RELIABILITY procedure, ODS
  table keywords, 1389
RELIABILITY procedure, ODS table names, 1389
RELIABILITY procedure, PPLOT statement, *see* RELIABILITY procedure, PROBPLOT statement
RELIABILITY procedure, PROBPLOT statement, 1206, 1207, 1211, 1221, 1222, 1271, 1311, 1312, 1317, 1319, 1321, 1324
  CONVERGE= option, 1355
  COVB option, 1206, 1207
  NOCONF option, 1211, 1221, 1222
  OVERLAY option, 1211
  PCONFPLT option, 1221, 1222
  PPOS= option, 1344, 1345, 1347
  READOUT option, 1221, 1222
  summary of options, 1312, 1317, 1319, 1321, 1324
RELIABILITY procedure, RELATIONPLOT statement, 1216, 1217, 1271, 1325, 1326, 1329, 1332, 1335, 1337, 1338
  CONVERGE= option, 1355
  FIT= option, 1216, 1217
  LUPPER= option, 1216, 1217
  NOCONF option, 1216, 1217
  PLOTDATA option, 1216, 1217
  PLOTFIT option, 1216, 1217
  PPLOT option, 1216, 1217
  PPOS= option, 1344, 1345, 1347
  RELATION= option, 1216, 1217
  SLOWER= option, 1216, 1217
  summary of options, 1326, 1329, 1332, 1335, 1337, 1338
RELIABILITY procedure, RPLOT statement, *see* RELIABILITY procedure, RELATIONPLOT statement
RELIABILITY procedure, SLICE statement, 1338
RELIABILITY procedure, STORE statement, 1339
RELIABILITY procedure, TEST statement, 1339
RELIABILITY procedure, UNITID statement, 1247, 1252, 1271, 1272, 1339
REPEAT option
  CUSUM procedure, 2028
  MACONTROL procedure, 2028
  SHEWHART procedure, 2028
RESET option
  MACONTROL procedure, 819
responses, ANOM procedure
  BOXCHART statement, 53
  PCHART statement, 86
  UCHART statement, 110
  XCHART statement, 138
RPREFIX= option
  PROC MVPDIAGNOSE statement, 907
  PROC MVPMODEL statement, 940
  PROC MVPMONITOR statement, 964
RSYMBOL= option
  SHEWHART procedure, 2028
RTMARGIN= option
  SHEWHART procedure, 2029
RTMPLOT= option
  SHEWHART procedure, 2029, 2173

sas/qc, 7
SCALE= option
  PARETO procedure, 1109
SCHART statement, *see also* SHEWHART procedure, SCHART statement
  examples, advanced, 1798
  examples, introductory, 1768
  options summarized by function, 1779
  overview, 1767
  syntax, 1778
SCHEME= option
  CUSUM procedure, 582
SCORECHART statement
  MVPMONITOR procedure, 965
SCOREMATRIX statement
  MVPDIAGNOSE procedure, 910
SCOREPLOT statement
  MVPDIAGNOSE procedure, 911
SEED= option
  PROC MVPMODEL statement, 937
SEPARATE option

SHEWHART procedure, 2029
SERIES statement
MVPMONITOR procedure, 965
SERIESVALUE= option
chart statement, 972
SERIFS option
SHEWHART procedure, 2029
SHEWHART procedure, 1410
and PROC ARIMA, 2145, 2148–2152
and PROC CAPABILITY, 2173, 2174
and PROC MACONTROL, 2150
and PROC MIXED, 2157, 2158
and PROC PRINCOMP, 2178
syntax, 1410
SHEWHART procedure, all chart statements
ALLLABEL= option, 1994
ALPHA= option, 1995
ANNOTATE= option, 2056
BILEVEL option, 2057
BLOCKLABELPOS= option, 1995
BLOCKLABTYPE= option, 1996
BLOCKPOS= option, 1996
BLOCKREP option, 1997
CAXIS= option, 2057
CBLOCKLAB= option, 2057
CBLOCKVAR= option, 2057
CCONNECT= option, 2058
CCOVERLAY2= option, 2058
CCOVERLAY= option, 2058
CFRAME= option, 2059
CFRAMELAB= option, 2001
CGRID= option, 2059
CHREF= option, 2059
CINFILL= option, 2002
CLABEL= option, 2059
CLIMITS= option, 2059
CONNECTCHAR= option, 2070
COUT= option, 2003
COUTFILL= option, 2060
COVERLAY2= option, 2060
COVERLAY= option, 2060
COVERLAYCLIP= option, 2060
CPHASELEG= option, 2060
CTESTLABBOX= option, 2061
CTESTS= option, 2061
CTESTSYMBOL= option, 2062
CTEXT= option, 2062
CVREF= option, 2062
CZONES= option, 2062
DESCRIPTION= option, 2062
DISCRETE option, 2005
ENDGRID option, 2062
EXCHART option, 2005
FONT= option, 2062
GRID option, 2005
HAXIS= option, 2005
HEIGHT= option, 2062
HMINOR= option, 2063
HOFFSET= option, 2006
HREF2DATA= option, 2006
HREF= option, 2006
HREFCHAR= option, 2070
HREFDATA= option, 2007
HREFLABELS= option, 2007
HREFLABPOS= option, 2007
HTML2= option, 2063
HTML= option, 2063
HTML_LEGEND= option, 2063
INTERVAL= option, 2008
INTSTART= option, 2009
LABELANGLE= option, 2064
LABELFONT= option, 2064
LABELHEIGHT= option, 2064
LCLLABEL= option, 2009
LENDGRID= option, 2064
LGRID= option, 2065
LHREF= option, 2065
LIMITN= option, 2009
LLIMITS= option, 2065
LOVERLAY2= option, 2065
LOVERLAY= option, 2065
LTESTS= option, 2066
LVREF= option, 2066
LZONES= option, 2066
MARKERS option, 2053
MAXPANELS= option, 2011
NAME= option, 2066
NDECIMAL= option, 2012
NO3SIGMACHECK option, 2013
NOBYREF option, 2013
NOCHART option, 2013
NOCONNECT option, 2013
NOCTL option, 2014
NOFRAME option, 2066
NOHLABEL option, 2014
NOLCL option, 2014
NOLEGEND option, 2014
NOLIMITLABEL option, 2014
NOLIMITS option, 2014
NOLIMITSFRAME option, 2066
NOLIMITSLEGEND option, 2014
NOOVERLAYLEGEND option, 2015
NOPHASEFRAME option, 2067
NOREADLIMITS option, 2015
NOTESTACROSS option, 2016
NOUCL option, 2017
NOV2LABEL option, 2067
NOVANGLE option, 2067

NOVLABEL option, 2067
NPANELPOS= option, 2017
OUTHIGHHTML= option, 2067
OUTHIGHURL= option, 2055
OUTHISTORY= option, 2018
OUTINDEX= option, 2018
OUTLABEL= option, 2018
OUTLIMITS= option, 2019
OUTLOWHTML= option, 2067
OUTLOWURL= option, 2055
OUTPHASE= option, 2019
OUTTABLE= option, 2019
OVERLAY2= option, 2019
OVERLAY2HTML= option, 2067
OVERLAY2ID= option, 2020
OVERLAY2SYM= option, 2067
OVERLAY2SYMHT= option, 2067
OVERLAY2URL= option, 2055
OVERLAY= option, 2019
OVERLAYCLIPSYM= option, 2067
OVERLAYCLIPSYMHT= option, 2067
OVERLAYHTML= option, 2067
OVERLAYID= option, 2020
OVERLAYLEGLAB= option, 2020
OVERLAYSYM= option, 2068
OVERLAYSYMHT= option, 2068
OVERLAYURL= option, 2055
PAGENUM= option, 2020
PAGENUMPOS= option, 2020
PHASEBREAK option, 2021
PHASELABTYPE= option, 2021
PHASELEGEND option, 2021
PHASELIMITS option, 2021
PHASEPOS= option, 2055
PHASEREF option, 2021
PHASEVALSEP option, 2022
PHASEVARLABEL option, 2021, 2022
POINTSHTML= option, 2068
POINTSURL= option, 2056
READALPHA option, 2023
READINDEX= option, 2023
READLIMITS option, 2024
READPHASES= option, 2024
REPEAT option, 2028
SIGMAS= option, 2030
SKIPHLABELS= option, 2030
SYMBOLCHARS= option, 2070
SYMBOLLEGEND= option, 2037
SYMBOLORDER= option, 2037
TABLE option, 2037
TABLEALL option, 2037
TABLECENTRAL option, 2038
TABLEID option, 2038
TABLELEGEND option, 2038
TABLEOUTLIM option, 2038
TABLETESTS option, 2038
TEST2RESET= option, 2039
TEST2RUN= option, 2039
TEST3RUN= option, 2039
TESTACROSS option, 2039
TESTCHAR= option, 2070
TESTFONT= option, 2068
TESTHEIGHT= option, 2068
TESTLABBOX option, 2040
TESTLABEL= option, 2040
TESTLABEL*n*= option, 2040
TESTNMETHOD= option, 2040
TESTOVERLAP option, 2041
TESTRESET= option, 2041
TESTS= option, 2041
TOTPANELS= option, 2044
TURNALL option, 2068
TURNHLABELS option, 2068
TYPE= option, 2044
UCLLABEL= option, 2045
URL2= option, 2056
URL= option, 2056
VAXIS= option, 2046
VFORMAT2= option, 2046
VFORMAT= option, 2046
VMINOR= option, 2069
VOFFSET= option, 2046
VREF= option, 2046
VREFCHAR= option, 2071
VREFLABELS= option, 2047
VREFLABPOS= option, 2047
WAXIS= option, 2069
WEBOUT= option, 2069
WESTGARD= option, 2048
WGRID= option, 2069
WLIMITS= option, 2069
WNEEDLES= option, 2069
WOVERLAY2= option, 2069
WOVERLAY= option, 2069
WTESTS= option, 2069
ZEROSTD= option, 2050
ZONECHAR= option, 2071
ZONEVALPOS= option, 2050
SHEWHART procedure, attribute chart statements
ACTUALALPHA, 1994
PROBLIMITS= option, 2022
SHEWHART procedure, BOXCHART statement, *see also* SHEWHART procedure, all chart statements, 1481
ALPHA= option, 1448
BOX= data set, 1457
BOXSTYLE= option, 1464–1467
BOXWIDTHSCALE= option, 1470–1472

CONTROLSTAT= option, 1422, 1447, 1448
DATA= data set, 1453, 1454
HISTORY= data set, 1423, 1424, 1426, 1455, 1456
LBOXES= option, 1472, 1474
LIMITN= option, 1448
LIMITS= data set, 1431, 1432, 1454
LSL= option, 1449
MEDCENTRAL= option, 1448
missing values, 1975
MU0= option, 1448
NOCHART option, 1426
NOLEGEND option, 2153
NOLIMITS option, 2153
NOTCHES option, 1469, 1470
OUTBOX= data set, 1450
OUTBOX= option, 2017
OUTHISTORY= data set, 1426–1428, 1451, 1452
OUTLIMITS= data set, 1428, 1429, 1448, 1450
OUTTABLE= data set, 1429–1431, 1452, 1453
RANGES option, 1427, 2023
SERIFS option, 1464, 1465
SIGMA0= option, 1448
SIGMAS= option, 1448
SMETHOD= option, 1459, 1460
STDDEVIATIONS option, 2153
TABLE= data set, 1431, 1456, 1457
TARGET= option, 1449
TESTS= option, 2119
USL= option, 1449
SHEWHART procedure, BY statement, 1410
SHEWHART procedure, CCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1505
CSYMBOL= option, 1514, 1516
DATA= data set, 1508, 1509
HISTORY= data set, 1488–1491, 1509, 1510
LIMITN= option, 1505
LIMITS= data set, 1487, 1488, 1509
LTESTS= option, 1512, 1513
NOCHART option, 1484, 1485
NOLEGEND option, 1514, 1516
OUTHISTORY= data set, 1491, 1492, 1506, 1507
OUTLIMITS= data set, 1484, 1485, 1506
OUTTABLE= data set, 1485, 1486, 1507, 1508
SIGMAS= option, 1505
SUBGROUPN= option, 1491, 1492
TABLE= data set, 1486, 1510, 1511
TABLELEGEND option, 1512, 1513
TABLETESTS option, 1512, 1513
TESTS= option, 1512, 1513, 2119
U0= option, 1505, 1514, 1516
ZONELABELS option, 1512, 1513
SHEWHART procedure, INSET and INSET2 statements, *see* INSET and INSET2 statements
SHEWHART procedure, INSET statement
CFILL= option, 1987
CFILLH= option, 1988
CFRAME= option, 1988
CHEADER= option, 1988
CSHADOW= option, 1988
CTEXT= option, 1988
DATA option, 1986
FONT= option, 1988
FORMAT= option, 1986
HEADER= option, 1986
HEIGHT= option, 1986
HTRANSPARENCY= option, 1987
NOFRAME option, 1987
POSITION= option, 1987–1990
REFPOINT= option, 1987
TRANSPARENCY= option, 1987
SHEWHART procedure, IRCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1543
DATA= data set, 1547
HISTORY= data set, 1522, 1523, 1548, 1549
LIMITN= option, 1528, 1543
LIMITS= data set, 1526, 1527, 1547, 1548
LSL= option, 1545
LTMARGIN= option, 1558
LTMPLOT= option, 1557, 1558
MU0= option, 1543, 1554, 2170
NOCHART option, 1521
OUTHISTORY= data set, 1521, 1522, 1545
OUTLIMITS= data set, 1523, 1544, 1545
OUTTABLE= data set, 1524, 1525, 1546, 1547
PHASEBREAK option, 2166
RTMPLOT= option, 1556–1558, 2173
SIGMA0= option, 1543, 1554, 2170
SIGMAS= option, 1543
TABLE= data set, 1526, 1549, 1550
TABLETESTS option, 1551, 1553
TARGET= option, 1545
TEST2RUN= option, 1551, 1553, 1554
TESTS= option, 1551, 1553, 1554, 2119
USL= option, 1545
XSYMBOL= option, 1554
ZONELABELS option, 1551, 1553, 1554
SHEWHART procedure, MCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1587
DATA= data set, 1591

HISTORY= data set, 1563, 1565, 1566, 1592, 1593
LIMITN= option, 1587
LIMITS= data set, 1571, 1572, 1591, 1592
LSL= option, 1588
MEDCENTRAL= option, 1587, 1596–1598
MU0= option, 1587, 1598, 1599
NDECIMAL= option, 1595
NOCHART option, 1566, 1567
OUTHISTORY= data set, 1566–1568, 1589
OUTLIMITS= data set, 1569, 1587–1589
OUTTABLE= data set, 1569–1571, 1589, 1590
SIGMA0= option, 1587
SIGMAS= option, 1587
SMETHOD= option, 1600–1602
STDDEVIATIONS option, 1568, 1601, 1602
TABLE= data set, 1570, 1571, 1593, 1594
TARGET= option, 1588
TESTS= option, 2119
USL= option, 1588
XSYMBOL= option, 1598, 1599
SHEWHART procedure, MRCHART statement, *see also* SHEWHART procedure, all chart statements
ALLN option, 1641, 1642
ALPHA= option, 1630
DATA= data set, 1634
HISTORY= data set, 1606–1609, 1635, 1636
LIMITN= option, 1630, 1640–1642
LIMITS= data set, 1613, 1614, 1634, 1635
MEDCENTRAL= option, 1630
MU0= option, 1630
NMARKERS option, 1641, 1642
NOCHART option, 1609, 1610
OUTHISTORY= data set, 1609, 1610, 1632
OUTLIMITS= data set, 1610, 1611, 1631, 1632
OUTTABLE= data set, 1611, 1613, 1633, 1634
SIGMA0= option, 1630
SIGMAS= option, 1630
SMETHOD= option, 1637, 1638, 1642, 1643
TABLE= data set, 1613, 1636, 1637
TESTS2= option, 2134
TESTS= option, 2119
SHEWHART procedure, NPCHART statement, *see also* SHEWHART procedure, all chart statements
ALLN option, 1682, 1683
ALPHA= option, 1669
DATA= data set, 1672, 1673
DATAUNIT= option, 1649, 1650
HISTORY= data set, 1650, 1651, 1674
LIMITN= option, 1669, 1682, 1683
LIMITS= data set, 1655, 1673, 1678, 1680, 1684–1686
LTESTS= option, 1676, 1677
NEEDLES option, 1678, 1680
NOLEGEND option, 1678, 1680
NPSYMBOL= option, 1678, 1680
OUTHISTORY= data set, 1652, 1670, 1671
OUTLIMITS= data set, 1652, 1653, 1669, 1670, 1680–1683
OUTTABLE= data set, 1653, 1671, 1672
P0= option, 1669, 1678, 1680
SIGMAS= option, 1669
SUBGROUPN= option, 1648, 1680–1683
TABLE= data set, 1653, 1654, 1675
TABLELEGEND option, 1676, 1677
TABLETESTS option, 1676, 1677
TESTS= option, 1676, 1677, 2119
ZONELABELS option, 1676, 1677
SHEWHART procedure, PCHART statement, *see also* SHEWHART procedure, all chart statements
ALLN option, 1723
ALPHA= option, 1710
DATA= data set, 1713
DATAUNIT= option, 1691
FONT= option, 1722
HISTORY= data set, 1691, 1692, 1715
LIMITN= option, 1710, 1723
LIMITS= data set, 1696, 1714, 1719, 1721
LTESTS= option, 1717, 1718
NEEDLES option, 1719, 1721
NOLEGEND option, 1719, 1721
OUTHISTORY= data set, 1693, 1711, 1712
OUTLIMITS= data set, 1694, 1711, 1721, 1723
OUTTABLE= data set, 1694, 1695, 1712, 1713
P0= option, 1710, 1719, 1721
PSYMBOL= option, 1719, 1721
READINDEX= option, 1726, 1727
SIGMAS= option, 1710
SUBGROUPN= option, 1688, 1721, 1723
TABLE= data set, 1695, 1715, 1716
TABLELEGEND option, 1717, 1718
TABLETESTS option, 1717, 1718
TESTS= option, 1717, 1718, 2119
VREF= option, 1726, 1727
VREFLABELS= option, 1726, 1727
VREFLABPOS= option, 1726, 1727
YSCALE= option, 1723
ZONELABELS option, 1717, 1718
SHEWHART procedure, PROC SHEWHART statement
CIINDICES= option, 2002
SHEWHART procedure, RCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1754, 1761, 1762
DATA= data set, 1757

HISTORY= data set, 1733, 1734, 1736, 1758, 1759
LIMITN= option, 1754
LIMITS= data set, 1740, 1741, 1758, 1763–1765
LSL= option, 1755
NOCHART option, 1736, 1737
NOLIMIT0 option, 1765
OUTHISTORY= data set, 1736, 1737, 1755, 1756
OUTLIMITS= data set, 1737, 1738, 1754, 1755, 1761, 1762
OUTTABLE= data set, 1738, 1739, 1756, 1757
READALPHA option, 1763
SIGMA0= option, 1754, 1765
SIGMAS= option, 1754
SMETHOD= option, 1760, 1761
TABLE= data set, 1739, 1759, 1760
TARGET= option, 1755
TESTS2= option, 2134
USL= option, 1755
SHEWHART procedure, SCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1790
DATA= data set, 1794
HISTORY= data set, 1771, 1773, 1795, 1796
LIMITN= option, 1790
LIMITS= data set, 1776–1778, 1794, 1795
LSL= option, 1792
OUTHISTORY= data set, 1773, 1774, 1792
OUTLIMITS= data set, 1774, 1775, 1790, 1792
OUTTABLE= data set, 1775, 1776, 1793
SIGMA0= option, 1790, 1798, 1799
SIGMAS= option, 1790
SMETHOD= option, 1797, 1798
SSYMBOL= option, 1798, 1799
TABLE= data set, 1776, 1796, 1797
TARGET= option, 1792
TESTS2= option, 2134
USL= option, 1792
SHEWHART procedure, UCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1825
DATA= data set, 1828
HISTORY= data set, 1808–1811, 1829, 1830
LIMITN= option, 1825
LIMITS= data set, 1807, 1808, 1829
LTESTS= option, 1832
NOCHART option, 1805
OUTHISTORY= data set, 1811, 1812, 1826, 1827
OUTLIMITS= data set, 1805, 1825, 1826, 1835–1838
OUTTABLE= data set, 1805, 1806, 1827, 1828
SIGMAS= option, 1825
SUBGROUPN= option, 1803, 1812, 1836–1838
TABLE= data set, 1806, 1830, 1831
TABLETESTS option, 1832
TESTS= option, 1832, 2119
U0= option, 1825, 1833, 1835
USYMBOL= option, 1833, 1835
ZONELABELS option, 1832
SHEWHART procedure, XCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1864
BLOCKLABELPOS= option, 2078, 2167, 2168
BLOCKLABTYPE= option, 2167, 2168
BLOCKPOS= option, 2076–2078
CBLOCKVAR= option, 2077, 2078
CFRAME= option, 2079
CNEEDLES= option, 2104
CPHASELEG= option, 2079
DATA= data set, 1868
HISTORY= data set, 1842, 1844, 1845, 1869, 1870
LABELFONT= option, 2099
LIMITN= option, 1864
LIMITS= data set, 1849, 1850, 1868, 1869
LSL= option, 1865
LSTARCIRCLES= option, 2094, 2099
LTESTS= option, 1874
LTMARGIN= option, 2078
MU0= option, 1864
NOCHART option, 1845, 1846
NOLEGEND option, 1873, 1874, 2076–2079
OUTHISTORY= data set, 1845, 1846, 1866
OUTINDEX= option, 1878
OUTLIMITS= data set, 1846, 1847, 1864–1866, 1880
OUTTABLE= data set, 1847, 1848, 1867
PHASELEGEND option, 2079, 2083–2085, 2087, 2088
PHASEREF option, 2079, 2083–2085, 2087, 2088
READINDEXES= option, 2081, 2083–2085, 2087–2089
READPHASES= option, 2079–2081, 2083–2085, 2087–2089
SIGMA0= option, 1864
SIGMAS= option, 1864
SMETHOD= option, 1871–1873, 1877, 1878
STARBDRADIUS= option, 2099
STARCIRCLES= option, 2094
STARINRADIUS= option, 2094
STARLABEL= option, 2099
STARLEGEND= option, 2099
STAROUTRADIUS= option, 2094
STARSPECS= option, 2098, 2099
STARSTART= option, 2094–2097, 2099
STARTYPE= option, 2095–2097

STARVERTICES= option, 2092, 2094–2097, 2099
STDDEVIATIONS option, 1877, 1878
SYMBOLCHARS= option, 2073
SYMBOLLEGEND= option, 2073
TABLE= data set, 1848, 1870, 1871
TABLECENTRAL option, 1873, 1874
TABLELEGEND option, 1873, 1874
TABLETESTS option, 1873, 1874
TARGET= option, 1865
TESTS= option, 1873, 1874, 2119
TRENDVAR= option, 2104, 2167
USL= option, 1865
WSTARCIRCLES= option, 2094
ZONELABELS option, 1873, 1874
SHEWHART procedure, XRCHART statement, *see also* SHEWHART procedure, all chart statements
subgroup-variable, 1970, 1971
ALLN option, 1923, 2123
ALPHA= option, 1908
CLIPFACTOR= option, 2106–2108
CLIPLEGEND= option, 2108
CLIPLEGPOS= option, 2108
CLIPSUBCHAR= option, 2108
CLIPSYMBOL= option, 2108
CTESTS= option, 2134
CZONES= option, 2134
DATA= data set, 1912
HISTORY= data set, 1885–1887, 1913, 1914
LIMITN= option, 1908, 1922, 2123
LIMITS= data set, 1892, 1893, 1912, 1913, 1920
LSL= option, 1910
LTESTS= option, 2134
MU0= option, 1908, 1919, 1921, 2123
NMARKERS option, 1924
NOCHART option, 1888, 1889
OUTHISTORY= data set, 1888, 1910
OUTLIMITS= data set, 1889, 1890, 1909, 1910
OUTTABLE= data set, 1890, 1892, 1911, 1912
PHASEBREAK option, 2131, 2132
READINDEXES= option, 2131
READPHASES= option, 2129, 2131
SIGMA0= option, 1908, 1919, 1921, 2123
SIGMAS= option, 1908
SMETHOD= option, 1915, 1916, 1924, 2127
TABLE= data set, 1892, 1914, 1915
TABLETESTS option, 1917
TARGET= option, 1910, 1973
TESTACROSS option, 2132, 2134
TESTCHAR= option, 2134
TESTLABEL= option, 2128, 2134
TESTLABEL*n*= option, 2134
TESTNMETHOD= option, 2125, 2132, 2134
TESTS2= option, 2134
TESTS= option, 2119, 2122, 2123
USL= option, 1910
XSYMBOL= option, 1920
ZONECHAR= option, 2134
ZONELABELS option, 1917, 2134
ZONES option, 2134
SHEWHART procedure, XSCHART statement, *see also* SHEWHART procedure, all chart statements
ALPHA= option, 1951, 1959
DATA= data set, 1954
HISTORY= data set, 1955, 1956
LIMITN= option, 1951
LIMITS= data set, 1955
LSL= option, 1952
MU0= option, 1951
NOHLABEL option, 1964, 1966
NOLEGEND option, 1964, 1966
OUTHISTORY= data set, 1932, 1952, 1953
OUTLIMITS= data set, 1933, 1951, 1952, 1959
OUTTABLE= data set, 1933, 1935, 1953, 1954
SIGMA0= option, 1951
SIGMAS= option, 1951
SPLIT= option, 2112
TABLE= data set, 1935, 1956, 1957
TARGET= option, 1952
TESTS2= option, 2134
TESTS= option, 2119
USL= option, 1952
SHIFT= option
CUSUM procedure, 582
SIGMA0= option
CUSUM procedure, 583
MACONTROL procedure, 819, 873
SHEWHART procedure, 2029, 2123, 2170
SIGMAS= option
CUSUM procedure, 583
MACONTROL procedure, 820, 873
SCORECHART statement, 966
SHEWHART procedure, 2030
SIMULATEQCFONT option
ANOM procedure, 2056
CUSUM procedure, 2056
MACONTROL procedure, 2056
SHEWHART procedure, 2056
SIZE statement, FACTEX procedure, *see* FACTEX procedure, SIZE statement
syntax, 639
SKIPHLABELS= option
CUSUM procedure, 2030
MACONTROL procedure, 2030
SHEWHART procedure, 2030
SLICE statement

RELIABILITY procedure, 1338
SMETHOD= option
CUSUM procedure, 583
MACONTROL procedure, 2031
SHEWHART procedure, 2031, 2127
SPAN= option
MACONTROL procedure, 874
SPEC statement
options summarized by function, 215
syntax, 214
SPECHART statement
MVPMONITOR procedure, 967
SPLIT= option
CUSUM procedure, 2031
MACONTROL procedure, 2031
SHEWHART procedure, 2031, 2112
SSYMBOL= option
SHEWHART procedure, 2031
STARBDRADIUS= option
CUSUM procedure, 2032
MACONTROL procedure, 2032
SHEWHART procedure, 2032, 2099
STARCIRCLES= option
CUSUM procedure, 2032
MACONTROL procedure, 2032
SHEWHART procedure, 2032, 2094
STARFILL= option
ANOM procedure, 2033
CUSUM procedure, 2033
MACONTROL procedure, 2033
SHEWHART procedure, 2033
STARINRADIUS= option
CUSUM procedure, 2033
MACONTROL procedure, 2033
SHEWHART procedure, 2033, 2094
STARLABEL= option
CUSUM procedure, 2033
MACONTROL procedure, 2033
SHEWHART procedure, 2033, 2099
STARLEGEND= option
CUSUM procedure, 2034
MACONTROL procedure, 2034
SHEWHART procedure, 2034, 2099
STARLEGENDLAB= option
CUSUM procedure, 2034
MACONTROL procedure, 2034
SHEWHART procedure, 2034
STAROUTRADIUS= option
CUSUM procedure, 2034
MACONTROL procedure, 2034
SHEWHART procedure, 2034, 2094
STARS= option
ANOM procedure, 2034
CUSUM procedure, 2034
MACONTROL procedure, 2034
SHEWHART procedure, 2034
STARSPECS= option
CUSUM procedure, 2035
MACONTROL procedure, 2035
SHEWHART procedure, 2035, 2098, 2099
STARSTART= option
CUSUM procedure, 2036
MACONTROL procedure, 2036
SHEWHART procedure, 2036, 2094–2097, 2099
STARTRANSPARENCY= option
ANOM procedure, 2056
CUSUM procedure, 2056
MACONTROL procedure, 2056
SHEWHART procedure, 2056
STARTYPE= option
CUSUM procedure, 2036
MACONTROL procedure, 2036
SHEWHART procedure, 2036, 2095–2097
STARVERTICES= option
CUSUM procedure, 2036
MACONTROL procedure, 2036
SHEWHART procedure, 2036, 2092, 2094–2097, 2099
STATREFLABELS= option
CAPABILITY procedure, 538
STDDEVIATIONS option
SHEWHART procedure, 2031, 2036, 2153
STDMED function, 2235, 2236
STDSCORES option
PROC MVPMODEL statement, 940
STORE statement
RELIABILITY procedure, 1339
subgroup-variable, CUSUM procedure
XCHART statement, 570
subgroup-variable, MACONTROL procedure
EWMACHART statement, 807
MACHART statement, 861
subgroup-variable, SHEWHART procedure
BOXCHART statement, 1433
CCHART statement, 1493
IRCHART statement, 1529
MCHART statement, 1573
MRCHART statement, 1616
NPCHART statement, 1657
PCHART statement, 1698
RCHART statement, 1742
SCHART statement, 1779
UCHART statement, 1813
XCHART statement, 1851
XRCHART statement, 1895
XSCHART statement, 1937
SUBGROUPN= option
SHEWHART procedure, 2037

SYMBOL statement, 360, 363
symbol-variable, ANOM procedure
BOXCHART statement, 54
PCHART statement, 86
UCHART statement, 111
XCHART statement, 138
symbol-variable, CUSUM procedure
XCHART statement, 570
symbol-variable, MACONTROL procedure
EWMACHART statement, 808
MACHART statement, 862
symbol-variable, SHEWHART procedure
BOXCHART statement, 1433
CCHART statement, 1493
displaying, 2037, 2070
IRCHART statement, 1530
MCHART statement, 1574
MRCHART statement, 1616
NPCHART statement, 1657
PCHART statement, 1698
RCHART statement, 1743
SCHART statement, 1779
UCHART statement, 1813
XCHART statement, 1852
XRCHART statement, 1895
XSCHART statement, 1937
SYMBOLCHAR= option
PARETO procedure, 1117
SYMBOLCHARS= option
CUSUM procedure, 2070
MACONTROL procedure, 2070
SHEWHART procedure, 2070, 2073
SYMBOLLEGEND= option
CUSUM procedure, 2037
MACONTROL procedure, 2037
SHEWHART procedure, 2037, 2073
SYMBOLORDER= option
CUSUM procedure, 2037
MACONTROL procedure, 2037
SHEWHART procedure, 2037

TABLE option
MACONTROL procedure, 2037
SHEWHART procedure, 2037
TABLE= option
PROC MVPMONITOR statement, 964, 977
PROC RAREEVENTS statement, 1176, 1189
TABLEALL option
CUSUM procedure, 583
MACONTROL procedure, 2037
SHEWHART procedure, 2037
TABLEBOX= option
SHEWHART procedure, 2038
TABLECENTRAL option
MACONTROL procedure, 2038
SHEWHART procedure, 2038
TABLECHART option
CUSUM procedure, 583
TABLECOMP option
CUSUM procedure, 583
TABLEID option
CUSUM procedure, 583
MACONTROL procedure, 2038
SHEWHART procedure, 2038
TABLELEGEND option
SHEWHART procedure, 2038
TABLEOUT option
CUSUM procedure, 584
TABLEOUTLIM option
MACONTROL procedure, 2038
SHEWHART procedure, 2038
tables
extreme observations, number, 209
extreme values, number, 209
robust estimates of scale, 209
specialized capability indices, 210
TABLESUMMARY option
CUSUM procedure, 584
TABLETESTS option
SHEWHART procedure, 2038
TARGET= option
SHEWHART procedure, 2038
TEST statement
RELIABILITY procedure, 1339
TEST2RESET= option
SHEWHART procedure, 2039
TEST2RUN= option
SHEWHART procedure, 2039, 2125
TEST3RUN= option
SHEWHART procedure, 2039, 2125
TESTACROSS option
SHEWHART procedure, 2039, 2132, 2134
TESTCHAR= option
SHEWHART procedure, 2070, 2134
TESTFONT= option
SHEWHART procedure, 2068
TESTHEIGHT= option
SHEWHART procedure, 2068
TESTLABBOX option
SHEWHART procedure, 2040
TESTLABEL= option
SHEWHART procedure, 2040, 2128, 2134
TESTLABEL*n*= option
SHEWHART procedure, 2040, 2134
TESTNMETHOD= option
SHEWHART procedure, 2040, 2125, 2132, 2134
TESTOVERLAP option
SHEWHART procedure, 2041

TESTRESET= option
SHEWHART procedure, 2041
TESTS2= option
SHEWHART procedure, 2043, 2134
TESTS= option
SHEWHART procedure, 2041, 2122, 2123
TESTSYMBOL= option
SHEWHART procedure, 2068
TESTSYMBOLHT= option
SHEWHART procedure, 2068
TILELEGEND= option
PARETO procedure, 1116
TILELEGLABEL= option
PARETO procedure, 1116
TIME statement
MVPDIAGNOSE procedure, 912
MVPMONITOR procedure, 967
TOTPANELS= option
CHART statement, 1179
chart statement, 972
CUSUM procedure, 2044
MACONTROL procedure, 2044
SHEWHART procedure, 2044
TRENDVAR= option
CUSUM procedure, 2044
MACONTROL procedure, 2044
SHEWHART procedure, 2044, 2104, 2167
TRUNCATE option
CLASS statement (OPTEX), 1019
CLASS statement (RELIABILITY), 1279
TSQUARECHART statement
MVPMONITOR procedure, 968
TURNALL option
CUSUM procedure, 2068
MACONTROL procedure, 2068
SHEWHART procedure, 2068
TURNHLABELS option
CUSUM procedure, 2068
MACONTROL procedure, 2068
SHEWHART procedure, 2068
TURNVLABEL option
PARETO procedure, 1116
TURNVLABELS option
CAPABILITY procedure, 542
TYPE= option
ANOM procedure, 185
CONTRIBUTIONPANEL statement, 909
CONTRIBUTIONPLOT statement, 910
CUSUM procedure, 584
MACONTROL procedure, 2044
SHEWHART procedure, 2044

U0= option
SHEWHART procedure, 2045
U= option
ANOM procedure, 185
UCHART statement, ANOM procedure, *see* ANOM procedure, UCHART statement
examples, advanced, 127
examples, introductory, 106
options summarized by function, 111
overview, 105
syntax, 110
UCHART statement, SHEWHART procedure, *see* SHEWHART procedure, UCHART statement
examples, advanced, 1831
examples, introductory, 1802
options summarized by function, 1814
overview, 1801
syntax, 1812
UCLLABEL2= option
SHEWHART procedure, 2045
UCLLABEL= option
MACONTROL procedure, 2045
SHEWHART procedure, 2045
UDLLABEL= option
ANOM procedure, 185
UNITEFFECT statement, FACTEX procedure, *see* FACTEX procedure, UNITEFFECT statement
URL2= option
CUSUM procedure, 2056
MACONTROL procedure, 2056
SHEWHART procedure, 2056
URL= option
ANOM procedure, 2056
CUSUM procedure, 2056
MACONTROL procedure, 2056
PARETO procedure, 1109
SHEWHART procedure, 2056
USL= option
SHEWHART procedure, 2045
USYMBOL= option
ANOM procedure, 185
SHEWHART procedure, 2045

VAR statement
MVPMODEL procedure, 941
VAXIS2= option
SHEWHART procedure, 2046
VAXIS= option
CUSUM procedure, 2046
MACONTROL procedure, 2046
SHEWHART procedure, 2046
VAXISLABEL= option
CAPABILITY procedure, 538
VFORMAT2= option

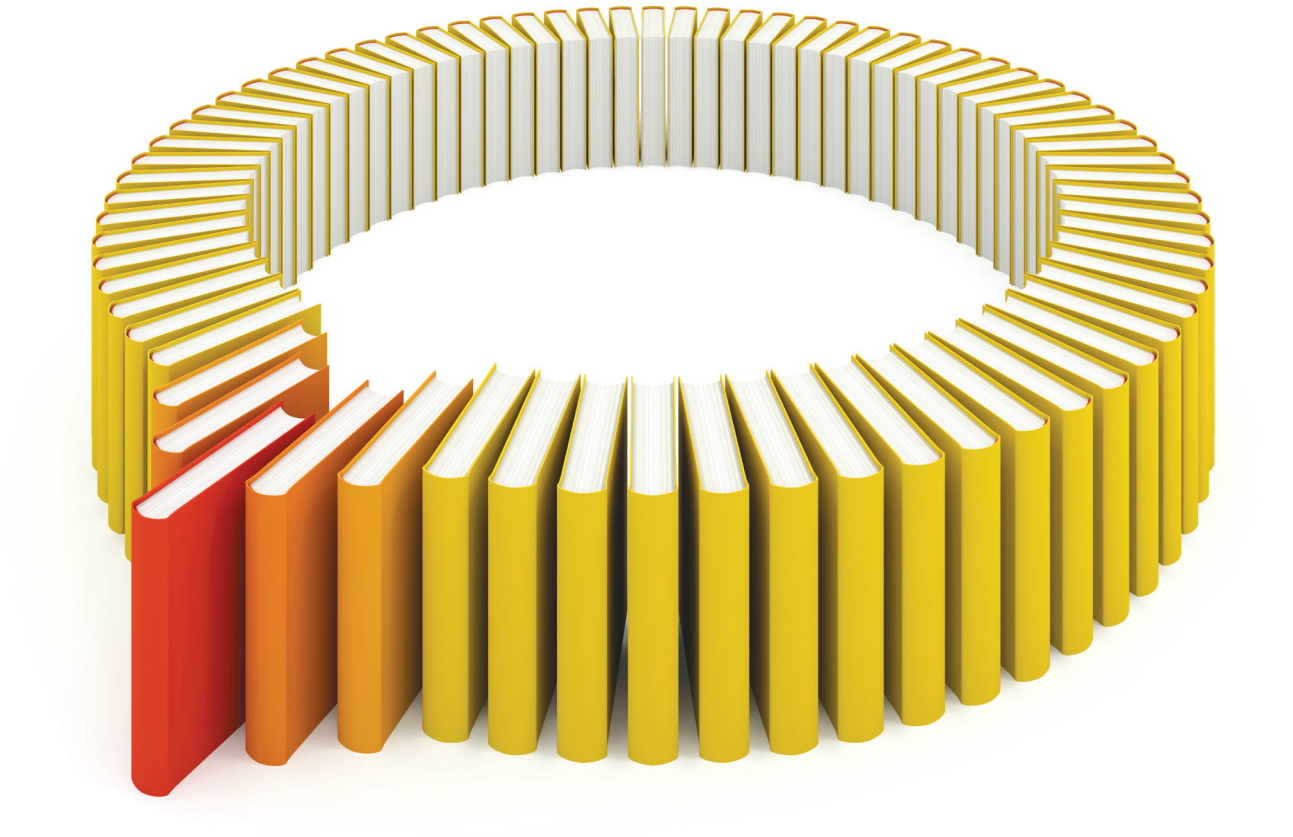
SHEWHART procedure, 2046
VFORMAT= option
SHEWHART procedure, 2046
VMINOR= option
CAPABILITY procedure, 542
CUSUM procedure, 2069
MACONTROL procedure, 2069
SHEWHART procedure, 2069
VOFFSET= option
CUSUM procedure, 2046
MACONTROL procedure, 2046
SHEWHART procedure, 2046
VREF2= option
CUSUM procedure, 2047
MACONTROL procedure, 2047
SHEWHART procedure, 2047
VREF2LABELS= option
CUSUM procedure, 2047
MACONTROL procedure, 2047
SHEWHART procedure, 2047
VREF= option
CAPABILITY procedure, 538
CUSUM procedure, 2046
MACONTROL procedure, 2046
SHEWHART procedure, 2046
VREFCHAR= option
CUSUM procedure, 2071
MACONTROL procedure, 2071
PARETO procedure, 1117
SHEWHART procedure, 2071
VREFLABELS= option
CAPABILITY procedure, 539
CUSUM procedure, 2047
MACONTROL procedure, 2047
SHEWHART procedure, 2047
VREFLABPOS= option
CAPABILITY procedure, 539
CUSUM procedure, 2047
MACONTROL procedure, 2047
PARETO procedure, 1110
SHEWHART procedure, 2047
VZERO option
SHEWHART procedure, 2048
VZERO2 option
SHEWHART procedure, 2048

W= option
CAPABILITY procedure, 542
WAXIS= option
CAPABILITY procedure, 542
CUSUM procedure, 2069
MACONTROL procedure, 2069
PARETO procedure, 1116
SHEWHART procedure, 2069
WBARLINE= option
PARETO procedure, 1117
WBOXES= option
ANOM procedure, 2056
SHEWHART procedure, 2056
WEBOUT= option
CUSUM procedure, 2069
MACONTROL procedure, 2069
SHEWHART procedure, 2069
WEIGHT= option
MACONTROL procedure, 820
PARETO procedure, 1110
WESTGARD= option
SHEWHART procedure, 2048
WGRID2= option
PARETO procedure, 1117
WGRID= option
CUSUM procedure, 2069
MACONTROL procedure, 2069
PARETO procedure, 1117
SHEWHART procedure, 2069
WHERE statement
SHEWHART procedure, 2114, 2116, 2117
WHISKERPERCENTILE= option
SHEWHART procedure, 2049
WLIMITS= option
ANOM procedure, 186
CUSUM procedure, 584
MACONTROL procedure, 2069
SHEWHART procedure, 2069
WMASK= option
CUSUM procedure, 584
WNEEDLES= option
MACONTROL procedure, 2069
SHEWHART procedure, 2069
WOVERLAY2= option
SHEWHART procedure, 2069
WOVERLAY= option
SHEWHART procedure, 2069
WSTARCIRCLES= option
CUSUM procedure, 2069
MACONTROL procedure, 2069
SHEWHART procedure, 2069, 2094
WSTARS= option
CUSUM procedure, 2069
MACONTROL procedure, 2069
SHEWHART procedure, 2069
WTESTS= option
SHEWHART procedure, 2069
WTREND= option
CUSUM procedure, 2070
MACONTROL procedure, 2070
SHEWHART procedure, 2070

XCHART statement, ANOM procedure, *see also* ANOM procedure, XCHART statement
  examples, advanced, 157
  examples, introductory, 130
  options summarized by function, 138, 146
  overview, 129
  syntax, 137
XCHART statement, CUSUM procedure, *see also* CUSUM procedure, XCHART statement
  examples, advanced, 602
  examples, introductory, 555
  notation, 585
  overview, 554
  syntax, 569
XCHART statement, SHEWHART procedure, *see also* SHEWHART procedure, XCHART statement
  examples, advanced, 1873
  examples, introductory, 1839
  options summarized by function, 1852
  overview, 1838
  syntax, 1851
XCOMP= option
  SCOREPLOT statement, 912
XRCHART statement, *see* SHEWHART procedure, XRCHART statement
  examples, advanced, 1916
  examples, introductory, 1882
  options summarized by function, 1896
  overview, 1881
  syntax, 1894
XSCHART statement, *see* SHEWHART procedure, XSCHART statement
  examples, advanced, 1959
  examples, introductory, 1926
  options summarized by function, 1938
  overview, 1925
  syntax, 1936
XSYMBOL= option
  ANOM procedure, 186
  MACONTROL procedure, 835, 889, 2049
  SHEWHART procedure, 1920, 2049

YCOMP= option
  SCOREPLOT statement, 912
YPCT1= option
  CUSUM procedure, 2050
  MACONTROL procedure, 2050
  SHEWHART procedure, 2050
YSCALE= option
  SHEWHART procedure, 2050

ZEROSTD option
  SHEWHART procedure, 2050
ZONE2LABELS option
  SHEWHART procedure, 2050
ZONE2VALUES option
  SHEWHART procedure, 2050
ZONECHAR= option
  SHEWHART procedure, 2071, 2134
ZONELABELS option
  SHEWHART procedure, 2050, 2134
ZONES option
  SHEWHART procedure, 2050, 2134
ZONES2 option
  SHEWHART procedure, 2050
ZONEVALPOS= option
  CUSUM procedure, 2050
  SHEWHART procedure, 2050
ZONEVALUES option
  SHEWHART procedure, 2051



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.

sas
THE POWER TO KNOW®

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613