SAS/STAT® 14.2 User's Guide

The MIANALYZE Procedure

This document is an individual chapter from *SAS/STAT® 14.2 User's Guide*.

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

SAS/STAT® 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.

Chapter 77
The MIANALYZE Procedure

Contents

Overview: MIANALYZE Procedure **6082**
Getting Started: MIANALYZE Procedure **6083**
Syntax: MIANALYZE Procedure **6085**
PROC MIANALYZE Statement 6086
BY Statement 6089
CLASS Statement 6089
MODELEFFECTS Statement 6089
STDERR Statement 6090
TEST Statement 6090
Details: MIANALYZE Procedure **6092**
Input Data Sets 6092
Combining Inferences from Imputed Data Sets 6096
Multiple Imputation Efficiency 6097
Multivariate Inferences 6098
Testing Linear Hypotheses about the Parameters 6099
Examples of the Complete-Data Inferences 6100
ODS Table Names 6102
Examples: MIANALYZE Procedure **6102**
Example 77.1: Reading Means and Standard Errors from a DATA= Data Set 6105
Example 77.2: Reading Means and Covariance Matrices from a DATA= COV Data Set 6107
Example 77.3: Reading Regression Results from a DATA= EST Data Set 6109
Example 77.4: Reading Mixed Model Results from PARMS= and COVB= Data Sets . 6111
Example 77.5: Reading Generalized Linear Model Results 6113
Example 77.6: Reading GLM Results from PARMS= and XPXI= Data Sets 6115
Example 77.7: Reading Logistic Model Results from a PARMS= Data Set 6117
Example 77.8: Reading Mixed Model Results with Classification Covariates 6118
Example 77.9: Reading Nominal Logistic Model Results 6120
Example 77.10: Using a TEST statement 6124
Example 77.11: Combining Correlation Coefficients 6125
Example 77.12: Sensitivity Analysis with Control-Based Pattern Imputation 6127
Example 77.13: Sensitivity Analysis with the Tipping-Point Approach 6130
References **6135**

Overview: MIANALYZE Procedure

The MIANALYZE procedure combines the results of the analyses of imputations and generates valid statistical inferences. Multiple imputation provides a useful strategy for analyzing data sets with missing values. Instead of filling in a single value for each missing value, Rubin's (1976, 1987) multiple imputation strategy replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute.

Multiple imputation inference involves three distinct phases:

1. The missing data are filled in *m* times to generate *m* complete data sets.
2. The *m* complete data sets are analyzed using standard statistical analyses.
3. The results from the *m* complete data sets are combined to produce inferential results.

A companion procedure, PROC MI, creates multiply imputed data sets for incomplete multivariate data. It uses methods that incorporate appropriate variability across the *m* imputations.

The analyses of imputations are obtained by using standard SAS procedures (such as PROC REG) for complete data. No matter which complete-data analysis is used, the process of combining results from different imputed data sets is essentially the same and results in valid statistical inferences that properly reflect the uncertainty due to missing values. These results of analyses are combined in the MIANALYZE procedure to derive valid inferences.

The MIANALYZE procedure reads parameter estimates and associated standard errors or covariance matrix that are computed by the standard statistical procedure for each imputed data set. The MIANALYZE procedure then derives valid univariate inference for these parameters. With an additional assumption about the population between and within imputation covariance matrices, multivariate inference based on Wald tests can also be derived.

The MODELEFFECTS statement lists the effects to be analyzed, and the CLASS statement lists the classification variables in the MODELEFFECTS statement. The variables in the MODELEFFECTS statement that are not specified in a CLASS statement are assumed to be continuous.

When each effect in the MODELEFFECTS statement is a continuous variable by itself, a STDERR statement specifies the standard errors when both parameter estimates and associated standard errors are stored as variables in the same data set.

For some parameters of interest, you can use TEST statements to test linear hypotheses about the parameters. For others, it is not straightforward to compute estimates and associated covariance matrices with standard statistical SAS procedures. Examples include correlation coefficients between two variables and ratios of variable means. These special cases are described in the section "Examples of the Complete-Data Inferences" on page 6100.

Getting Started: MIANALYZE Procedure

The Fitness data described in the REG procedure are measurements of 31 individuals in a physical fitness course. See Chapter 99, "The REG Procedure," for more information. The Fitness1 data set is constructed from the Fitness data set and contains three variables: Oxygen, RunTime, and RunPulse. Some values have been set to missing, and the resulting data set has an arbitrary pattern of missingness in these three variables.

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;
data Fitness1;
   input Oxygen RunTime RunPulse @@;
   datalines;
44.609  11.37  178     45.313  10.07  185
54.297   8.65  156     59.571    .      .
49.874   9.22    .     44.811  11.63  176
  .     11.95  176          .  10.85    .
39.442  13.08  174     60.055   8.63  170
50.541    .      .     37.388  14.03  186
44.754  11.12  176     47.273    .      .
51.855  10.33  166     49.156   8.95  180
40.836  10.95  168     46.672  10.00    .
46.774  10.25    .     50.388  10.08  168
39.407  12.63  174     46.080  11.17  156
45.441   9.63  164       .      8.92    .
45.118  11.08    .     39.203  12.88  168
45.790  10.47  186     50.545   9.93  148
48.673   9.40  186     47.920  11.50  170
47.467  10.50  170
;
```

Suppose that the data are multivariate normally distributed and that the missing data are missing at random (see the section "Statistical Assumptions for Multiple Imputation" on page 5980 in Chapter 76, "The MI Procedure," for more information about these assumptions). The following statements use the MI procedure to impute missing values for the Fitness1 data set:

```
proc mi data=Fitness1 seed=3237851 noprint out=outmi;
   var Oxygen RunTime RunPulse;
run;
```

The MI procedure creates imputed data sets, which are stored in the Outmi data set. A variable named _Imputation_ indicates the imputation numbers. Based on m imputations, m different sets of the point and variance estimates for a parameter can be computed. In PROC MI, $m = 25$ is the default.

The following statements generate regression coefficients for each of the 25 imputed data sets:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display (in Figure 77.1) output parameter estimates and covariance matrices from PROC REG for the first two imputed data sets:

```
proc print data=outreg(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'Parameter Estimates from Imputed Data Sets';
run;
```

Figure 77.1 Parameter Estimates

Parameter Estimates from Imputed Data Sets

Obs	_Imputation_	_TYPE_	_NAME_	Intercept	RunTime	RunPulse
1	1	PARMS		86.544	-2.82231	-0.05873
2	1	COV	Intercept	100.145	-0.53519	-0.55077
3	1	COV	RunTime	-0.535	0.10774	-0.00345
4	1	COV	RunPulse	-0.551	-0.00345	0.00343
5	2	PARMS		83.021	-3.00023	-0.02491
6	2	COV	Intercept	79.032	-0.66765	-0.41918
7	2	COV	RunTime	-0.668	0.11456	-0.00313
8	2	COV	RunPulse	-0.419	-0.00313	0.00264

The following statements combine the 25 sets of regression coefficients:

```
proc mianalyze data=outreg;
   modeleffects Intercept RunTime RunPulse;
run;
```

The “Model Information” table in Figure 77.2 lists the input data set(s) and the number of imputations.

Figure 77.2 Model Information Table

The MIANALYZE Procedure

Model Information	
Data Set	WORK.OUTREG
Number of Imputations	25

The “Variance Information” table in Figure 77.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences. It also displays the degrees of freedom for the total variance, the relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency for each parameter estimate.

Figure 77.3 Variance Information Table

Variance Information (25 Imputations)							
	Variance						
Parameter	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	22.485821	75.413875	98.799129	428.38	0.310092	0.240234	0.990482
RunTime	0.021126	0.124930	0.146902	1072.9	0.175870	0.151147	0.993990
RunPulse	0.000656	0.002622	0.003304	562.35	0.260376	0.209393	0.991694

The "Parameter Estimates" table in Figure 77.4 displays a combined estimate and standard error for each regression coefficient (parameter). Inferences are based on t distributions. The table displays a 95% confidence interval and a t test with the associated p-value for the hypothesis that the parameter is equal to the value specified with the THETA0= option (in this case, zero by default). The minimum and maximum parameter estimates from the imputed data sets are also displayed.

Figure 77.4 Parameter Estimates

Parameter Estimates (25 Imputations)										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	92.700420	9.939775	73.16362	112.2372	428.38	83.020730	100.839807	0	9.33	<.0001
RunTime	-3.030325	0.383278	-3.78238	-2.2783	1072.9	-3.280042	-2.754668	0	-7.91	<.0001
RunPulse	-0.079621	0.057482	-0.19253	0.0333	562.35	-0.135862	-0.024910	0	-1.39	0.1666

Syntax: MIANALYZE Procedure

The following statements are available in the MIANALYZE procedure:

```
PROC MIANALYZE <options> ;
    BY variables ;
    CLASS variables ;
    MODELEFFECTS effects ;
    <label:> TEST equation1 <, ..., <equationk>> </options> ;
    STDERR variables ;
```

The BY statement specifies groups in which separate analyses are performed.

The CLASS statement lists the classification variables in the MODELEFFECTS statement. Classification variables can be either character or numeric.

The required MODELEFFECTS statement lists the effects to be analyzed. The variables in the statement that are not specified in a CLASS statement are assumed to be continuous.

The STDERR statement lists the standard errors associated with the effects in the MODELEFFECTS statement when both parameter estimates and standard errors are saved as variables in the same DATA= data set. The STDERR statement can be used only when each effect in the MODELEFFECTS statement is a continuous variable by itself.

The TEST statement tests linear hypotheses about the parameters. An F statistic is used to jointly test the null hypothesis ($H_0 : \mathbf{L}_{\epsilon} = \mathbf{c}$) specified in a single TEST statement. Several TEST statements can be used.

The PROC MIANALYZE and MODELEFFECTS statements are required for the MIANALYZE procedure. The rest of this section provides detailed syntax information for each of these statements, beginning with the PROC MIANALYZE statement. The remaining statements are in alphabetical order.

PROC MIANALYZE Statement

PROC MIANALYZE < *options* > **;**

The PROC MIANALYZE statement invokes the MIANALYZE procedure. Table 77.1 summarizes the options available in the PROC MIANALYZE statement.

Table 77.1 Summary of PROC MIANALYZE Options

Option	Description
Input Data Sets	
DATA=	Specifies the COV, CORR, or EST type data set
DATA=	Specifies the data set for parameter estimates and standard errors
PARMS=	Specifies the data set for parameter estimates
PARMINFO=	Specifies the data set for parameter information
COVB=	Specifies the data set for covariance matrices
XPXI=	Specifies the data set for $(\mathbf{X}'\mathbf{X})^{-1}$ matrices
Statistical Analysis	
THETA0=	Specifies parameters under the null hypothesis
ALPHA=	Specifies the level for the confidence interval
EDF=	Specifies the complete-data degrees of freedom
Printed Output	
WCOV	Displays the within-imputation covariance matrix
BCOV	Displays the between-imputation covariance matrix
TCOV	Displays the total covariance matrix
MULT	Displays multivariate inferences

The following options can be used in the PROC MIANALYZE statement. They are listed in alphabetical order.

ALPHA=α

specifies that confidence limits are to be constructed for the parameter estimates with confidence level $100(1-\alpha)\%$, where $0 < \alpha < 1$. The default is ALPHA=0.05.

BCOV

displays the between-imputation covariance matrix.

COVB <(EFFECTVAR=STACKING | ROWCOL)> =*SAS-data-set*

names an input SAS data set that contains covariance matrices of the parameter estimates from imputed data sets. If you provide a COVB= data set, you must also provide a PARMS= data set.

The EFFECTVAR= option identifies the variables for parameters displayed in the covariance matrix and is used only when the PARMINFO= option is not specified. The default is EFFECTVAR= STACKING.

See the section "Input Data Sets" on page 6092 for a detailed description of the COVB= option.

DATA=*SAS-data-set*

names an input SAS data set.

If the input DATA= data set is not a specially structured SAS data set, the data set contains both the parameter estimates and associated standard errors. The parameter estimates are specified in the MODELEFFECTS statement and the standard errors are specified in the STDERR statement.

If the data set is a specially structured input SAS data set, it must have a TYPE of EST, COV, or CORR that contains estimates from imputed data sets:

- If TYPE=EST, the data set contains the parameter estimates and associated covariance matrices.
- If TYPE=COV, the data set contains the sample means, sample sizes, and covariance matrices. Each covariance matrix for variables is divided by the sample size n to create the covariance matrix for parameter estimates.
- If TYPE=CORR, the data set contains the sample means, sample sizes, standard errors, and correlation matrices. The covariance matrices are computed from the correlation matrices and associated standard errors. Each covariance matrix for variables is divided by the sample size n to create the covariance matrix for parameter estimates.

If you do not specify an input data set with the DATA= or PARMS= option, then the most recently created SAS data set is used as an input DATA= data set. See the section "Input Data Sets" on page 6092 for a detailed description of the input data sets.

EDF=*number*

specifies the complete-data degrees of freedom for the parameter estimates. This is used to compute an adjusted degrees of freedom for each parameter estimate. By default, EDF=∞ and the degrees of freedom for each parameter estimate are not adjusted.

MULT

MULTIVARIATE

requests multivariate inference for the parameters. It is based on Wald tests and is a generalization of the univariate inference. See the section "Multivariate Inferences" on page 6098 for a detailed description of the multivariate inference.

PARMINFO=*SAS-data-set*

names an input SAS data set that contains parameter information associated with variables PRM1, PRM2,..., and so on. These variables are used as variables for parameters in a COVB= data set. See the section "Input Data Sets" on page 6092 for a detailed description of the PARMINFO= option.

PARMS < (*options*) > =*SAS-data-set*

names an input SAS data set that contains parameter estimates computed from imputed data sets. When a COVB= data set is not specified, the input PARMS= data set also contains standard errors associated with these parameter estimates. If multivariate inference is requested, you must also provide a COVB= or XPXI= data set.

The available *options* are as follows:

CLASSVAR=FULL | LEVEL | CLASSVAL

identifies the associated classification variables when reading the classification levels from observations. The CLASSVAR= option is applicable only when the model effects contain classification variables. The default is CLASSVAR= FULL.

LINK=NONE | LOGIT | GLOGIT

identifies the type of parameter estimates. The LINK=NONE option (which is the default) indicates the parameter estimates that are derived from a procedure other than the LOGISTIC procedure.

The LINK=LOGIT option indicates the parameter estimates that are derived from the LOGISTIC procedure for ordinal responses. It is applicable only when the variable Intercept is in the MODELEFFECTS statement and the logistic model has more than two response levels. Otherwise, LINK=NONE should be used.

The LINK=GLOGIT option indicates the parameter estimates that are derived from the LOGISTIC procedure for nominal responses.

For a detailed description of the PARMS= option, see the section "PARMS < (*parms-options*) >= Data Set" on page 6093

TCOV

displays the total covariance matrix derived by assuming that the population between-imputation and within-imputation covariance matrices are proportional to each other.

THETA0=*numbers*

MU0=*numbers*

specifies the parameter values $\boldsymbol{\theta}_0$ under the null hypothesis $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ in the t tests for location for the effects. If only one number $\boldsymbol{\theta}_0$ is specified, that number is used for all effects. If more than one number is specified, the specified numbers correspond to effects in the MODELEFFECTS statement in the order in which they appear in the statement. When an effect contains classification variables, the corresponding value is not used and the test is not performed.

WCOV

displays the within-imputation covariance matrices.

XPXI=*SAS-data-set*

names an input SAS data set that contains the $(\mathbf{X}'\mathbf{X})^{-1}$ matrices associated with the parameter estimates computed from imputed data sets. If you provide an XPXI= data set, you must also provide a PARMS= data set. In this case, PROC MIANALYZE reads the standard errors of the estimates from the PARMS= data. The standard errors and $(\mathbf{X}'\mathbf{X})^{-1}$ matrices are used to derive the covariance matrices.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MIANALYZE 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 MIANALYZE 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 *variables* **;**

The CLASS statement specifies the classification variables in the MODELEFFECTS statement. Classification variables can be either character or numeric. Classification levels are determined from the formatted values of the classification variables. See "The FORMAT Procedure" in the *Base SAS Procedures Guide* for details.

MODELEFFECTS Statement

MODELEFFECTS *effects* **;**

The MODELEFFECTS statement lists the effects in the data set to be analyzed. Each effect is a variable or a combination of variables, and is specified with a special notation that uses variable names and operators.

Each variable is either a classification (or CLASS) variable or a continuous variable. If a variable is not declared in the CLASS statement, it is assumed to be continuous. Crossing and nesting operators can be used in an effect to create crossed and nested effects.

One general form of an effect involving several variables is

X1 * *X2* * *A* * *B* * *C* (*D E*)

where A, B, C, D, and E are classification variables and X1 and X2 are continuous variables.

When the input DATA= data set is not a specially structured SAS data set, you must also specify standard errors of the parameter estimates in an STDERR statement.

STDERR Statement

STDERR *variables* **;**

The STDERR statement lists standard errors associated with effects in the MODELEFFECTS statement, when the input DATA= data set contains both parameter estimates and standard errors as variables in the data set.

With the STDERR statement, only continuous effects are allowed in the MODELEFFECTS statement. The specified standard errors correspond to parameter estimates in the order in which they appear in the MODELEFFECTS statement.

For example, you can use the following MODELEFFECTS and STDERR statements to identify both the parameter estimates and associated standard errors in a SAS data set:

```
proc mianalyze;
   modeleffects y1 y2;
   stderr sy1 sy2;
run;
```

TEST Statement

< *label*: > **TEST** *equation1* <, . . . , < *equationk* > > < / *options* > **;**

The TEST statement tests linear hypotheses about the parameters $\boldsymbol{\beta}$. An F test is used to jointly test the null hypotheses ($H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$) specified in a single TEST statement in which the MULT option is specified.

Each *equation* specifies a linear hypothesis (a row of the $\mathbf{L}$ matrix and the corresponding element of the $\mathbf{c}$ vector); multiple *equations* are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output. You can submit multiple TEST statements. When a label is not included in a TEST statement, a label of "Test *j*" is used for the *j*th TEST statement.

The form of an *equation* is as follows:

term < ± *term* . . . > < = ± *term* < ± *term* . . . > >

where *term* is a parameter of the model, or a constant, or a constant times a parameter. When no equal sign appears, the expression is set to 0. Only parameters for regressor effects (continuous variables by themselves) are allowed.

For each TEST statement, PROC MIANALYZE displays a "Test Specification" table of the $\mathbf{L}$ matrix and the $\mathbf{c}$ vector. The procedure also displays a "Variance Information" table of the between-imputation, within-imputation, and total variances for combining complete-data inferences, and a "Parameter Estimates" table of a combined estimate and standard error for each linear component. The linear components are labeled TestPrm1, TestPrm2, ... in the tables.

The following statements illustrate possible uses of the TEST statement:

```
proc mianalyze;
   modeleffects intercept a1 a2 a3;
   test1: test intercept + a2 = 0;
   test2: test intercept + a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

The first and second TEST statements are equivalent and correspond to the specification in Figure 77.5.

Figure 77.5 Test Specification for test1 and test2

The MIANALYZE Procedure
Test: test1

Test Specification					
	L Matrix				
Parameter	intercept	a1	a2	a3	C
TestPrm1	1.000000	0	1.000000	0	0

The third and fourth TEST statements are also equivalent and correspond to the specification in Figure 77.6.

Figure 77.6 Test Specification for test3 and test4

The MIANALYZE Procedure
Test: test3

Test Specification					
	L Matrix				
Parameter	intercept	a1	a2	a3	C
TestPrm1	0	1.000000	-1.000000	0	0
TestPrm2	0	0	1.000000	-1.000000	0

The ALPHA= and EDF options specified in the PROC MIANALYZE statement are also applied to the TEST statement. You can specify the following options in the TEST statement after a slash(/):

BCOV
displays the between-imputation covariance matrix.

MULT
displays the multivariate inference for parameters.

TCOV
displays the total covariance matrix.

WCOV
displays the within-imputation covariance matrix.

For more information, see the section "Testing Linear Hypotheses about the Parameters" on page 6099.

Details: MIANALYZE Procedure

Input Data Sets

You specify input data sets based on the type of inference you requested. For univariate inference, you can use one of the following options:

- a DATA= data set, which provides both parameter estimates and the associated standard errors
- a DATA=EST, COV, or CORR data set, which provides both parameter estimates and the associated standard errors either explicitly (type CORR) or through the covariance matrix (type EST, COV)
- PARMS= data set, which provides both parameter estimates and the associated standard errors

For multivariate inference, which includes the testing of linear hypotheses about parameters, you can use one of the following option combinations:

- a DATA=EST, COV, or CORR data set, which provides parameter estimates and the associated covariance matrix either explicitly (type EST, COV) or through the correlation matrix and standard errors (type CORR) in a single data set
- PARMS= and COVB= data sets, which provide parameter estimates in a PARMS= data set and the associated covariance matrix in a COVB= data set
- PARMS=, COVB=, and PARMINFO= data sets, which provide parameter estimates in a PARMS= data set, the associated covariance matrix in a COVB= data set with variables named PRM1, PRM2, . . . , and the effects associated with these variables in a PARMINFO= data set
- PARMS= and XPXI= data sets, which provide parameter estimates and the associated standard errors in a PARMS= data set and the associated $(\mathbf{X}'\mathbf{X})^{-1}$ matrix in an XPXI= data set

The appropriate combination depends on the type of inference and the SAS procedure you used to create the data sets. For instance, if you used PROC REG to create an OUTEST= data set that contains the parameter estimates and covariance matrix, you would use the DATA= option to read the OUTEST= data set.

When the input DATA= data set is a specially structured SAS data set, the data set must contain the variable _Imputation_ to identify the imputation by number. Otherwise, each observation corresponds to an imputation and contains both parameter estimates and associated standard errors.

If you do not specify an input data set with the DATA= or PARMS= option, then the most recently created SAS data set is used as an input DATA= data set. Note that with a DATA= data set, each effect represents a continuous variable; only regressor effects (continuous variables by themselves) are allowed in the MODELEFFECTS statement.

DATA= SAS Data Set

The DATA= data set provides both parameter estimates and the associated standard errors computed from imputed data sets. Such data sets are typically created with an OUTPUT statement in procedures such as PROC MEANS and PROC UNIVARIATE.

The MIANALYZE procedure reads parameter estimates from observations with variables in the MODEL-EFFECTS statement, and standard errors for parameter estimates from observations with variables in the STDERR statement. The order of the variables for standard errors must match the order of the variables for parameter estimates.

DATA=EST, COV, or CORR SAS Data Set

The specially structured DATA= data set provides both parameter estimates and the associated covariance matrix computed from imputed data sets. Such data sets are created by procedures such as PROC CORR (type COV, CORR) and PROC REG (type EST).

With a DATA=EST data set, the MIANALYZE procedure reads parameter estimates from observations with _TYPE_='PARM', _TYPE_='PARMS', _TYPE_='OLS', or _TYPE_='FINAL', and covariance matrices for parameter estimates from observations with _TYPE_='COV' or _TYPE_='COVB'.

With a DATA=COV data set, the procedure reads sample means from observations with _TYPE_='MEAN', sample size *n* from observations with _TYPE_='N', and covariance matrices for variables from observations with _TYPE_='COV'.

With a DATA=CORR data set, the procedure reads sample means from observations with _TYPE_='MEAN', sample size *n* from observations with _TYPE_='N', correlation matrices for variables from observations with _TYPE_='CORR', and standard errors for variables from observations with _TYPE_='STD'. The standard errors and correlation matrix are used to generate a covariance matrix for the variables.

Note that with a DATA=COV or DATA=CORR data set, each covariance matrix for the variables is divided by *n* to create the covariance matrix for the sample means.

PARMS <(*parms-options*) >= Data Set

The PARMS= data set contains both parameter estimates and the associated standard errors computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement in procedures such as PROC GENMOD, PROC GLM, PROC LOGISTIC, and PROC MIXED.

The MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate and standard errors for parameter estimates from observations with the variable StdErr.

The available *parms-options* include the CLASSVAR= option to identify classification variables and the LINK= option to input logistic regression results. When the parameter estimates are derived from the LOGISTIC procedure, the LINK= option can be used to identify the variable required when the parameter estimates are read from observations. The available options are as follows:

- LINK=NONE (which is the default), in which each model effect is completely identified from the effect name. This option should be used for all procedures except PROC LOGISTIC.
- LINK=LOGIT, in which the variable ClassVal0 is used to identify response levels for Intercept from PROC LOGISTIC for ordinal responses. This option is applicable only when the variable Intercept is in

the MODELEFFECTS statement and the logistic model has more than two response levels. Otherwise, LINK=NONE should be used.

- LINK=GLOGIT, in which the variable Response is used to identify response levels for the parameters from PROC LOGISTIC for nominal responses.

When the effects contain classification variables, the CLASSVAR= option can be used to identify the variables when reading the classification levels from observations. The available options are:

- CLASSVAR=FULL (which is the default), the data set contains the classification variables explicitly. PROC MIANALYZE reads the classification levels from observations with their corresponding classification variables. PROC MIXED generates this type of table.
- CLASSVAR=LEVEL, PROC MIANALYZE reads the classification levels for the effect from observations with variables Level1, Level2, and so on, where the variable Level1 contains the classification level for the first classification variable in the effect, and the variable Level2 contains the classification level for the second classification variable in the effect. For each effect, the variables in the crossed list are displayed before the variables in the nested list. The variable order in the CLASS statement is used for variables inside each list. PROC GENMOD generates this type of table.

  For example, with the following statements, the variable Level1 has the classification level of the variable c2 for the effect c2:

  ```
  proc mianalyze parms(classvar=Level)=dataparm;
     class c1 c2 c3;
     modeleffects c2 c3(c2 c1);
  run;
  ```

  For the effect c3(c2 c1), the variable Level1 has the classification level of the variable c3, Level2 has the level of c1, and Level3 has the level of c2.
- CLASSVAR=CLASSVAL, PROC MIANALYZE reads the classification levels for the effect from observations with variables ClassVal0, ClassVal1, and so on, where the variable ClassVal0 contains the classification level for the first classification variable in the effect, and the variable ClassVal1 contains the classification level for the second classification variable in the effect. For each effect, the variables in the crossed list are displayed before the variables in the nested list. The variable order in the CLASS statement is used for variables inside each list. PROC LOGISTIC generates this type of tables.

PARMS <(*parms-options*) >= and COVB <(EFFECTVAR=*etype*) >= Data Sets

The PARMS= data set contains parameter estimates, and the COVB= data set contains associated covariance matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement in procedures such as PROC LOGISTIC, PROC MIXED, and PROC REG.

When you specify a PARMS= data set, the MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate.

The available *parms-options* include the CLASSVAR= option to identify classification variables and the LINK= option to input logistic regression results. For a detailed description of the PARMS= option, see the section "PARMS <(*parms-options*) >= Data Set" on page 6093.

The EFFECTVAR=*etype* option identifies the variables for parameters displayed in the covariance matrix. The available types are STACKING and ROWCOL:

- EFFECTVAR=STACKING (which is the default), each parameter is displayed by stacking variables in the effect. Begin with the variables in the crossed list, followed by the continuous list, then followed by the nested list. Each classification variable is displayed with its classification level attached. PROC LOGISTIC generates this type of table. When each effect is a continuous variable by itself, each stacked parameter name reduces to the effect name. PROC REG generates this type of table.

  The MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the stacked variables in a COVB= data set.

- EFFECTVAR=ROWCOL, parameters are displayed by the variables Col1, Col2, ... The parameter associated with the variable Col1 is identified by the observation with value 1 for the variable Row. The parameter associated with the variable Col2 is identified by the observation with value 2 for the variable Row. PROC MIXED generates this type of table.

  The MIANALYZE procedure reads the parameter indices from observations with the variable Row and the effect names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the variables Col1, Col2, and so on in a COVB= data set.

  When the effects contain classification variables, the data set contains the classification variables explicitly and the MIANALYZE procedure also reads the classification levels from their corresponding classification variables.

PARMS <(CLASSVAR= *ctype*)> =, PARMINFO=, and COVB= Data Sets

The input PARMS= data set contains parameter estimates, the PARMINFO= data set identifies parameters with the variables Prm1, Prm2, and so on, and the COVB= data set contains associated covariance matrices with the variables Prm1, Prm2, and so on. Such data sets are typically created with an ODS OUTPUT statement using procedure such as PROC GENMOD.

When you specify a PARMS= data set, the MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate.

When the effects contain classification variables, the option CLASSVAR= *ctype* can be used to identify the associated classification variables when reading the classification levels from observations. The available types are FULL, LEVEL, and CLASSVAL, and they are described in the section "PARMS <(*parms-options*) >= Data Set" on page 6093. The default is CLASSVAR= FULL.

When you specify a COVB= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the variables Prm1, Prm2, and so on.

The parameters associated with the variables Prm1, Prm2, and so on are identified in the PARMINFO= data set. PROC MIANALYZE reads the parameter names from observations with the variable Parameter and the corresponding effect from observations with the variable Effect. When the effects contain classification variables, the data set contains the classification variables explicitly and the MIANALYZE procedure also reads the classification levels from observations with their corresponding classification variables.

PARMS= and XPXI= Data Sets

The input PARMS= data set contains parameter estimates, and the input XPXI= data set contains associated $(\mathbf{X}'\mathbf{X})^{-1}$ matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement in a procedure such as PROC GLM.

When you specify a PARMS= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate and standard errors for parameter estimates from observations with the variable StdErr.

When you specify a XPXI= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter and $(\mathbf{X}'\mathbf{X})^{-1}$ matrices from observations with the parameter variables in the data set.

Note that this combination can be used only when each effect is a continuous variable by itself.

Combining Inferences from Imputed Data Sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Suppose that $\hat{Q}_i$ and $\hat{W}_i$ are the point and variance estimates, respectively, from the ith imputed data set, $i = 1, 2, \ldots, m$. Then the combined point estimate for Q from multiple imputation is the average of the m complete-data estimates:

$$\overline{Q} = \frac{1}{m}\sum_{i=1}^{m}\hat{Q}_i$$

Suppose that $\overline{W}$ is the within-imputation variance, which is the average of the m complete-data estimates:

$$\overline{W} = \frac{1}{m}\sum_{i=1}^{m}\hat{W}_i$$

And suppose that B is the between-imputation variance:

$$B = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{Q}_i - \overline{Q})^2$$

Then the variance estimate associated with $\overline{Q}$ is the total variance (Rubin 1987)

$$T = \overline{W} + (1 + \frac{1}{m})B$$

The statistic $(Q - \overline{Q})T^{-(1/2)}$ is approximately distributed as t with v_m degrees of freedom (Rubin 1987), where

$$v_m = (m-1)\left[1 + \frac{\overline{W}}{(1+m^{-1})B}\right]^2$$

The degrees of freedom v_m depend on m and the ratio

$$r = \frac{(1+m^{-1})B}{\overline{W}}$$

The ratio r is called the relative increase in variance due to nonresponse (Rubin 1987). When there is no missing information about Q, the values of r and B are both zero. With a large value of m or a small value of r, the degrees of freedom v_m will be large and the distribution of $(Q - \overline{Q})T^{-(1/2)}$ will be approximately normal.

Another useful statistic is the fraction of missing information about Q:

$$\hat{\lambda} = \frac{r + 2/(v_m + 3)}{r + 1}$$

Both statistics r and λ are helpful diagnostics for assessing how the missing data contribute to the uncertainty about Q.

When the complete-data degrees of freedom v_0 are small, and there is only a modest proportion of missing data, the computed degrees of freedom, v_m, can be much larger than v_0, which is inappropriate. For example, with $m = 5$ and $r = 10\%$, the computed degrees of freedom $v_m = 484$, which is inappropriate for data sets with complete-data degrees of freedom less than 484.

Barnard and Rubin (1999) recommend the use of adjusted degrees of freedom

$$v_m^* = \left[\frac{1}{v_m} + \frac{1}{\hat{v}_{obs}}\right]^{-1}$$

where $\hat{v}_{obs} = (1-\gamma)\, v_0(v_0+1)/(v_0+3)$ and $\gamma = (1+m^{-1})B/T$.

If you specify the complete-data degrees of freedom v_0 with the EDF= option, the MIANALYZE procedure uses the adjusted degrees of freedom, v_m^*, for inference. Otherwise, the degrees of freedom v_m are used.

Multiple Imputation Efficiency

The relative efficiency (RE) of using the finite m imputation estimator, rather than using an infinite number for the fully efficient imputation, in units of variance, is approximately a function of m and λ (Rubin 1987, p. 114):

$$\mathrm{RE} = (1 + \frac{\lambda}{m})^{-1}$$

Table 77.2 shows relative efficiencies with different values of m and λ.

Table 77.2 Relative Efficiencies

	λ				
m	**10%**	**20%**	**30%**	**50%**	**70%**
3	0.9677	0.9375	0.9091	0.8571	0.8108
5	0.9804	0.9615	0.9434	0.9091	0.8772
10	0.9901	0.9804	0.9709	0.9524	0.9346
20	0.9950	0.9901	0.9852	0.9756	0.9662

The table shows that for situations with little missing information, only a small number of imputations are necessary. In practice, the number of imputations needed can be informally verified by replicating sets of m imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

Multivariate Inferences

Multivariate inference based on Wald tests can be done with m imputed data sets. The approach is a generalization of the approach taken in the univariate case (Rubin 1987, p. 137; Schafer 1997, p. 113). Suppose that $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{W}}_i$ are the point and covariance matrix estimates for a p-dimensional parameter $\mathbf{Q}$ (such as a multivariate mean) from the ith imputed data set, $i = 1, 2, \ldots, m$. Then the combined point estimate for $\mathbf{Q}$ from the multiple imputation is the average of the m complete-data estimates:

$$\overline{\mathbf{Q}} = \frac{1}{m}\sum_{i=1}^{m}\hat{\mathbf{Q}}_i$$

Suppose that $\overline{\mathbf{W}}$ is the within-imputation covariance matrix, which is the average of the m complete-data estimates:

$$\overline{\mathbf{W}} = \frac{1}{m}\sum_{i=1}^{m}\hat{\mathbf{W}}_i$$

And suppose that $\mathbf{B}$ is the between-imputation covariance matrix:

$$\mathbf{B} = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{\mathbf{Q}}_i - \overline{\mathbf{Q}})(\hat{\mathbf{Q}}_i - \overline{\mathbf{Q}})'$$

Then the covariance matrix associated with $\overline{\mathbf{Q}}$ is the total covariance matrix

$$\mathbf{T}_0 = \overline{\mathbf{W}} + (1 + \frac{1}{m})\mathbf{B}$$

The natural multivariate extension of the t statistic used in the univariate case is the F statistic

$$F_0 = (\mathbf{Q} - \overline{\mathbf{Q}})'\mathbf{T}_0^{-1}(\mathbf{Q} - \overline{\mathbf{Q}})$$

with degrees of freedom p and

$$v = (m-1)(1+1/r)^2$$

where

$$r = (1+\frac{1}{m})\,\mathrm{trace}(\mathbf{B}\overline{\mathbf{W}}^{-1})/p$$

is an average relative increase in variance due to nonresponse (Rubin 1987, p. 137; Schafer 1997, p. 114).

However, the reference distribution of the statistic F_0 is not easily derived. Especially for small m, the between-imputation covariance matrix $\mathbf{B}$ is unstable and does not have full rank for $m \leq p$ (Schafer 1997, p. 113).

One solution is to make an additional assumption that the population between-imputation and within-imputation covariance matrices are proportional to each other (Schafer 1997, p. 113). This assumption implies that the fractions of missing information for all components of $\mathbf{Q}$ are equal. Under this assumption, a more stable estimate of the total covariance matrix is

$$\mathbf{T} = (1+r)\overline{\mathbf{W}}$$

With the total covariance matrix $\mathbf{T}$, the F statistic (Rubin 1987, p. 137)

$$F = (\mathbf{Q}-\overline{\mathbf{Q}})'\mathbf{T}^{-1}(\mathbf{Q}-\overline{\mathbf{Q}})/p$$

has an F distribution with degrees of freedom p and v_1, where

$$v_1 = \frac{1}{2}(p+1)(m-1)(1+\frac{1}{r})^2$$

For $t = p(m-1) \leq 4$, PROC MIANALYZE uses the degrees of freedom v_1 in the analysis. For $t = p(m-1) > 4$, PROC MIANALYZE uses v_2, a better approximation of the degrees of freedom given by Li, Raghunathan, and Rubin (1991):

$$v_2 = 4 + (t-4)\left[1+\frac{1}{r}(1-\frac{2}{t})\right]^2$$

Testing Linear Hypotheses about the Parameters

Linear hypotheses for parameters $\boldsymbol{\beta}$ are expressed in matrix form as

$$H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses and $\mathbf{c}$ is a vector of constants.

Suppose that $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{U}}_i$ are the point and covariance matrix estimates, respectively, for a p-dimensional parameter $\mathbf{Q}$ from the $i\,\mathrm{th}$ imputed data set, i=1, 2, . . . , m. Then for a given matrix $\mathbf{L}$, the point and covariance matrix estimates for the linear functions $\mathbf{LQ}$ in the $i\,\mathrm{th}$ imputed data set are, respectively,

$$\mathbf{L}\hat{\mathbf{Q}}_i$$

$$\mathbf{L}\hat{\mathbf{U}}_i\mathbf{L}'$$

The inferences described in the section "Combining Inferences from Imputed Data Sets" on page 6096 and the section "Multivariate Inferences" on page 6098 are applied to these linear estimates for testing the null hypothesis $H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$.

For each TEST statement, the "Test Specification" table displays the $\mathbf{L}$ matrix and the $\mathbf{c}$ vector, the "Variance Information" table displays the between-imputation, within-imputation, and total variances for combining complete-data inferences, and the "Parameter Estimates" table displays a combined estimate and standard error for each linear component.

With the WCOV and BCOV options in the TEST statement, the procedure displays the within-imputation and between-imputation covariance matrices, respectively.

With the TCOV option, the procedure displays the total covariance matrix derived under the assumption that the population between-imputation and within-imputation covariance matrices are proportional to each other.

With the MULT option in the TEST statement, the "Multivariate Inference" table displays an F test for the null hypothesis $\mathbf{L}\boldsymbol{\beta} = \mathbf{c}$ of the linear components.

Examples of the Complete-Data Inferences

For a given parameter of interest, it is not always possible to compute the estimate and associated covariance matrix directly from a SAS procedure. This section describes examples of parameters with their estimates and associated covariance matrices, which provide the input to the MIANALYZE procedure. Some are straightforward, and others require special techniques.

Means

For a population mean vector $\boldsymbol{\mu}$, the usual estimate is the sample mean vector

$$\overline{\mathbf{y}} = \frac{1}{n} \sum \mathbf{y}_i$$

A variance estimate for $\overline{\mathbf{y}}$ is $\frac{1}{n}\mathbf{S}$, where $\mathbf{S}$ is the sample covariance matrix

$$\mathbf{S} = \frac{1}{n-1} \sum (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})'$$

These statistics can be computed from a procedure such as PROC CORR. This approach is illustrated in Example 77.2.

Regression Coefficients

Many SAS procedures are available for regression analysis. Among them, PROC REG provides the most general analysis capabilities, and others like PROC LOGISTIC and PROC MIXED provide more specialized analyses.

Some regression procedures, such as REG and LOGISTIC, create an EST type data set that contains both the parameter estimates for the regression coefficients and their associated covariance matrix. You can read an EST type data set in the MIANALYZE procedure with the DATA= option. This approach is illustrated in Example 77.3.

Other procedures, such as GLM, MIXED, and GENMOD, do not generate EST type data sets for regression coefficients. For PROC MIXED and PROC GENMOD, you can use ODS OUTPUT statement to save parameter estimates in a data set and the associated covariance matrix in a separate data set. These data sets are then read in the MIANALYZE procedure with the PARMS= and COVB= options, respectively. This approach is illustrated in Example 77.4 for PROC MIXED and in Example 77.5 for PROC GENMOD.

PROC GLM does not display tables for covariance matrices. However, you can use the ODS OUTPUT statement to save parameter estimates and associated standard errors in a data set and the associated $(\mathbf{X}'\mathbf{X})^{-1}$ matrix in a separate data set. These data sets are then read in the MIANALYZE procedure with the PARMS= and XPXI= options, respectively. This approach is illustrated in Example 77.6.

For univariate inference, only parameter estimates and associated standard errors are needed. You can use the ODS OUTPUT statement to save parameter estimates and associated standard errors in a data set. This data set is then read in the MIANALYZE procedure with the PARMS= option. This approach is illustrated in Example 77.4.

Correlation Coefficients

For the population correlation coefficient ρ, a point estimate is the sample correlation coefficient r. However, for nonzero ρ, the distribution of r is skewed.

The distribution of r can be normalized through Fisher's z transformation

$$z(r) = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right)$$

$z(r)$ is approximately normally distributed with mean $z(\rho)$ and variance $1/(n-3)$.

With a point estimate $\hat{z}$ and an approximate 95% confidence interval (z_1, z_2) for $z(\rho)$, a point estimate $\hat{r}$ and a 95% confidence interval (r_1, r_2) for ρ can be obtained by applying the inverse transformation

$$r = \tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$$

to $z = \hat{z}, z_1$, and z_2.

This approach is illustrated in Example 77.11.

Ratios of Variable Means

For the ratio μ_1/μ_2 of means for variables Y_1 and Y_2, the point estimate is $\overline{y}_1/\overline{y}_2$, the ratio of the sample means. The Taylor expansion and delta method can be applied to the function y_1/y_2 to obtain the variance estimate (Schafer 1997, p. 196)

$$\frac{1}{n}\left[\left(\frac{\overline{y}_1}{\overline{y}_2^2}\right)^2 s_{22} - 2\left(\frac{\overline{y}_1}{y_2^2}\right)\left(\frac{1}{\overline{y}_2}\right)s_{12} + \left(\frac{1}{\overline{y}_2}\right)^2 s_{11}\right]$$

where s_{11} and s_{22} are the sample variances of Y_1 and Y_2, respectively, and s_{12} is the sample covariance between Y_1 and Y_2.

A ratio of sample means will be approximately unbiased and normally distributed if the coefficient of variation of the denominator (the standard error for the mean divided by the estimated mean) is 10% or less (Cochran 1977, p. 166; Schafer 1997, p. 196).

ODS Table Names

PROC MIANALYZE assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in Table 77.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 77.3 ODS Tables Produced by PROC MIANALYZE

ODS Table Name	Description	Statement	Option
BCov	Between-imputation covariance matrix		BCOV
ModelInfo	Model information		
MultStat	Multivariate inference		MULT
ParameterEstimates	Parameter estimates		
TCov	Total covariance matrix		TCOV
TestBCov	Between-imputation covariance matrix for $\mathbf{L}\boldsymbol{\beta}$	TEST	BCOV
TestMultStat	Multivariate inference for $\mathbf{L}\boldsymbol{\beta}$	TEST	MULT
TestParameterEstimates	Parameter estimates for $\mathbf{L}\boldsymbol{\beta}$	TEST	
TestSpec	Test specification, $\mathbf{L}$ and $\mathbf{c}$	TEST	
TestTCov	Total covariance matrix for $\mathbf{L}\boldsymbol{\beta}$	TEST	TCOV
TestVarianceInfo	Variance information for $\mathbf{L}\boldsymbol{\beta}$	TEST	
TestWCov	Within-imputation covariance matrix for $\mathbf{L}\boldsymbol{\beta}$	TEST	WCOV
VarianceInfo	Variance information		
WCov	Within-imputation covariance matrix		WCOV

Examples: MIANALYZE Procedure

The following statements generate five imputed data sets to be used in this section. The data set Fitness1 was created in the section "Getting Started: MIANALYZE Procedure" on page 6083. For more information about the MI procedure, See Chapter 76, "The MI Procedure."

```
proc mi data=Fitness1 seed=3237851 noprint out=outmi;
   var Oxygen RunTime RunPulse;
run;
```

The Fish data described in the STEPDISC procedure are measurements of 159 fish of seven species caught in Finland's Lake Laengelmaevesi. For each fish, the length, height, and width are measured. See Chapter 110, "The STEPDISC Procedure," for more information.

The Fish2 data set is constructed from the Fish data set and contains two species of fish. Some values have been set to missing, and the resulting data set has a monotone missing pattern in the variables Length, Width, and Species.

The following statements create the Fish2 data set. It contains two species of fish in the Fish data set.

```
*----------------------------Fish2 Data-----------------------------*
| The data set contains two species of the fish (Parkki and Perch)   |
| and two measurements: Length and Width.                            |
| Some values have been set to missing, and the resulting data set   |
| has a monotone missing pattern in the variables                    |
| Length, Width, and Species.                                        |
*--------------------------------------------------------------------*;
data Fish2;
   title 'Fish Measurement Data';
   input Species $ Length Width @@;
   datalines;
Parkki  16.5  2.3265    Parkki  17.4  2.3142    .      19.8   .
Parkki  21.3  2.9181    Parkki  22.4  3.2928    .      23.2  3.2944
Parkki  23.2  3.4104    Parkki  24.1  3.1571    .      25.8  3.6636
Parkki  28.0  4.1440    Parkki  29.0  4.2340    Perch   8.8  1.4080
.       14.7  1.9992    Perch   16.0  2.4320    Perch  17.2  2.6316
Perch   18.5  2.9415    Perch   19.2  3.3216    .      19.4   .
Perch   20.2  3.0502    Perch   20.8  3.0368    Perch  21.0  2.7720
Perch   22.5  3.5550    Perch   22.5  3.3075    .      22.5   .
Perch   22.8  3.5340    .       23.5   .        Perch  23.5  3.5250
Perch   23.5  3.5250    Perch   23.5  3.5250    Perch  23.5  3.9950
.       24.0   .        Perch   24.0  3.6240    Perch  24.2  3.6300
Perch   24.5  3.6260    Perch   25.0  3.7250    .      25.5  3.7230
Perch   25.5  3.8250    Perch   26.2  4.1658    Perch  26.5  3.6835
.       27.0  4.2390    Perch   28.0  4.1440    Perch  28.7  5.1373
.       28.9  4.3350    .       28.9   .        .      28.9  4.5662
Perch   29.4  4.2042    Perch   30.1  4.6354    Perch  31.6  4.7716
Perch   34.0  6.0180    .       36.5  6.3875    .      37.3  7.7957
.       39.0   .        .       38.3   .        Perch  39.4  6.2646
Perch   39.3  6.3666    Perch   41.4  7.4934    Perch  41.4  6.0030
Perch   41.3  7.3514    .       42.3   .        Perch  42.5  7.2250
Perch   42.4  7.4624    Perch   42.5  6.6300    Perch  44.6  6.8684
Perch   45.2  7.2772    Perch   45.5  7.4165    Perch  46.0  8.1420
Perch   46.6  7.5958
;
```

The following statements generate 25 imputed data sets to be used in this section. The default regression method is used to impute missing values in continuous variable Width, and the discriminant function method is used to impute the variable Species.

```
proc mi data=Fish2 seed=1305417 out=outfish2;
   class Species;
   monotone logistic( Species= Length Width);
   var Length Width Species;
run;
```

The Fish3 data set is constructed from the Fish data set and contains three species of fish. Some values have been set to missing, and the resulting data set has an arbitrary missing pattern in the variables Length, Width, and Species.

The following statements create the Fish3 data set. It contains two species of fish in the Fish data set.

```
*-----------------------------Fish3 Data-----------------------------*
| The data set contains three species of the fish                    |
| (Parkki, Perch, and Roach) and two measurements: Length and Width. |
| Some values have been set to missing, and the resulting data set   |
| has an arbitrary missing pattern in the variables                  |
| Length, Width, and Species.                                        |
*--------------------------------------------------------------------*;
data Fish3;
   title 'Fish Measurement Data';
   input Species $ Length Width @@;
   datalines;
Roach   16.2  2.2680    Roach   20.3  2.8217    Roach   21.2   .
Roach     .   3.1746    Roach   22.2  3.5742    Roach   22.8  3.3516
Roach   23.1  3.3957    .       23.7   .        Roach   24.7  3.7544
Roach   24.3  3.5478    Roach   25.3   .        Roach   25.0  3.3250
Roach   25.0  3.8000    Roach   27.2  3.8352    Roach   26.7  3.6312
Roach   26.8  4.1272    Roach   27.9  3.9060    Roach   29.2  4.4968
Roach   30.6  4.7736    Roach   35.0  5.3550    Parkki  16.5  2.3265
Parkki  17.4   .        Parkki  19.8  2.6730    Parkki  21.3  2.9181
Parkki  22.4  3.2928    Parkki  23.2  3.2944    Parkki  23.2  3.4104
Parkki  24.1  3.1571    .         .   3.6636    Parkki  28.0  4.1440
Parkki  29.0  4.2340    Perch    8.8  1.4080    .       14.7  1.9992
Perch   16.0  2.4320    Perch   17.2  2.6316    Perch   18.5  2.9415
Perch   19.2  3.3216    .       19.4  3.1234    Perch   20.2   .
Perch   20.8  3.0368    Perch   21.0  2.7720    Perch   22.5  3.5550
Perch   22.5  3.3075    Perch   22.5  3.6675    Perch     .   3.5340
Perch   23.5  3.4075    Perch   23.5  3.5250    Perch   23.5  3.5250
.       23.5  3.5250    Perch   23.5  3.9950    Perch   24.0  3.6240
Perch   24.0  3.6240    Perch   24.2  3.6300    Perch   24.5  3.6260
Perch   25.0  3.7250    Perch     .   3.7230    Perch   25.5  3.8250
Perch     .   4.1658    Perch   26.5  3.6835    .       27.0  4.2390
Perch     .   4.1440    Perch   28.7  5.1373    .       28.9  4.3350
Perch   28.9  4.3350    Perch   28.9  4.5662    Perch   29.4  4.2042
Perch   30.1  4.6354    Perch   31.6  4.7716    Perch   34.0  6.0180
Perch   36.5  6.3875    Perch   37.3  7.7957    Perch   39.0   .
Perch   38.3  6.7408    Perch     .   6.2646    .       39.3   .
Perch   41.4  7.4934    Perch   41.4  6.0030    Perch   41.3  7.3514
Perch   42.3  7.1064    Perch   42.5  7.2250    Perch   42.4  7.4624
Perch   42.5  6.6300    Perch   44.6  6.8684    Perch   45.2  7.2772
Perch   45.5  7.4165    Perch   46.0  8.1420    .       46.6  7.5958
;
```

The following statements generate 25 imputed data sets to be used in this section. The default regression method is used to impute missing values in continuous variable Width, and the nominal logistic regression method is used to impute the variable Species.

```
proc mi data=Fish3 seed=30535 out=outfish3;
   class Species;
   fcs logistic ( Species= Length Width / link=glogit);
   var Length Width Species;
run;
```

Example 77.1 through Example 77.7 use different input option combinations to combine parameter estimates computed from different procedures. Example 77.8 combines parameter estimates with classification variables, and Example 77.9 combines nominal logistic regression parameter estimates Example 77.10 shows the use of a TEST statement, and Example 77.11 combines statistics that are not directly derived from procedures.

The MI procedure provides sensitivity analysis for the MAR assumption. Example 77.12 illustrate sensitivity analysis by using the pattern-mixture model approach, and Example 77.13 performs sensitivity analysis by searching and examining the tipping point that reverses the study conclusion.

Example 77.1: Reading Means and Standard Errors from a DATA= Data Set

This example creates an ordinary SAS data set that contains sample means and standard errors computed from imputed data sets. These estimates are then combined to generate valid univariate inferences about the population means.

The following statements use the UNIVARIATE procedure to generate sample means and standard errors for the variables in each imputed data set:

```
proc univariate data=outmi noprint;
   var Oxygen RunTime RunPulse;
   output out=outuni mean=Oxygen RunTime RunPulse
                     stderr=SOxygen SRunTime SRunPulse;
   by _Imputation_;
run;
```

The following statements display the output data set from PROC UNIVARIATE shown in Output 77.1.1:

```
proc print data=outuni (obs=10);
   title 'UNIVARIATE Means and Standard Errors (First 10 Imputations)';
run;
```

Output 77.1.1 UNIVARIATE Output Data Set

UNIVARIATE Means and Standard Errors (First 10 Imputations)

Obs	_Imputation_	Oxygen	RunTime	RunPulse	SOxygen	SRunTime	SRunPulse
1	1	47.0120	10.4441	171.216	0.95984	0.28520	1.59910
2	2	47.2407	10.5040	171.244	0.93540	0.26661	1.75638
3	3	47.4995	10.5922	171.909	1.00766	0.26302	1.85795
4	4	47.1485	10.5279	171.146	0.95439	0.26405	1.75011
5	5	47.0042	10.4913	172.072	0.96528	0.27275	1.84807
6	6	47.2949	10.6135	169.969	0.95669	0.24533	2.06949
7	7	46.9617	10.5080	171.701	0.93181	0.27111	2.28368
8	8	46.9729	10.5222	170.622	0.94561	0.25687	1.69879
9	9	46.9864	10.5545	171.356	0.93664	0.26233	1.71076
10	10	46.9587	10.5518	171.197	0.94270	0.25214	1.89054

The following statements combine the means and standard errors from imputed data sets, The EDF= option requests that the adjusted degrees of freedom be used in the analysis. For sample means based on 31 observations, the complete-data error degrees of freedom is 30.

```
proc mianalyze data=outuni edf=30;
   modeleffects Oxygen RunTime RunPulse;
   stderr SOxygen SRunTime SRunPulse;
run;
```

The "Model Information" table in Output 77.1.2 lists the input data set(s) and the number of imputations.

Output 77.1.2 Model Information

The MIANALYZE Procedure

Model Information	
Data Set	WORK.OUTUNI
Number of Imputations	25

The "Variance Information" table in Output 77.1.3 displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency for each imputed variable are also displayed. A detailed description of these statistics is provided in the section "Combining Inferences from Imputed Data Sets" on page 6096 and the section "Multiple Imputation Efficiency" on page 6097.

Output 77.1.3 Variance Information

Variance Information (25 Imputations)

	Variance						
Parameter	**Between**	**Within**	**Total**	**DF**	**Relative Increase in Variance**	**Fraction Missing Information**	**Relative Efficiency**
Oxygen	0.026098	0.925531	0.952673	27.354	0.029325	0.028556	0.998859
RunTime	0.002938	0.068197	0.071253	26.918	0.044810	0.043035	0.998282
RunPulse	0.598494	3.345356	3.967790	23.196	0.186059	0.158595	0.993696

The "Parameter Estimates" table in Output 77.1.4 displays the estimated mean and corresponding standard error for each variable. The table also displays a 95% confidence interval for the mean and a t statistic with the associated p-value for testing the hypothesis that the mean is equal to the value specified. You can use the THETA0= option to specify the value for the null hypothesis, which is zero by default. The table also displays the minimum and maximum parameter estimates from the imputed data sets.

Output 77.1.4 Parameter Estimates

Parameter Estimates (25 Imputations)							
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum
Oxygen	47.084579	0.976050	45.0831	49.0861	27.354	46.850020	47.499541
RunTime	10.549499	0.266933	10.0017	11.0973	26.918	10.428848	10.680797
RunPulse	171.388418	1.991931	167.2697	175.5071	23.196	169.881385	173.125658

Parameter Estimates (25 Imputations)			
Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Oxygen	0	48.24	<.0001
RunTime	0	39.52	<.0001
RunPulse	0	86.04	<.0001

Note that the results in this example could also have been obtained with the MI procedure.

Example 77.2: Reading Means and Covariance Matrices from a DATA= COV Data Set

This example creates a COV-type data set that contains sample means and covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the population means.

The following statements use the CORR procedure to generate sample means and a covariance matrix for the variables in each imputed data set:

```
proc corr data=outmi cov nocorr noprint out=outcov(type=cov);
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display (in Output 77.2.1) output sample means and covariance matrices from PROC CORR for the first two imputed data sets:

```
proc print data=outcov(obs=12);
   title 'CORR Means and Covariance Matrices'
         ' (First Two Imputations)';
run;
```

Output 77.2.1 COV Data Set

CORR Means and Covariance Matrices (First Two Imputations)

Obs	_Imputation_	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	1	COV	Oxygen	28.5603	-7.2652	-11.812
2	1	COV	RunTime	-7.2652	2.5214	2.536
3	1	COV	RunPulse	-11.8121	2.5357	79.271
4	1	MEAN		47.0120	10.4441	171.216
5	1	STD		5.3442	1.5879	8.903
6	1	N		31.0000	31.0000	31.000
7	2	COV	Oxygen	27.1240	-6.6761	-10.217
8	2	COV	RunTime	-6.6761	2.2035	2.611
9	2	COV	RunPulse	-10.2170	2.6114	95.631
10	2	MEAN		47.2407	10.5040	171.244
11	2	STD		5.2081	1.4844	9.779
12	2	N		31.0000	31.0000	31.000

Note that the covariance matrices in the data set Outcov are estimated covariance matrices of variables, $V(\mathbf{y})$. The estimated covariance matrix of the sample means is $V(\overline{\mathbf{y}}) = V(\mathbf{y})/n$, where n is the sample size, and is not the same as an estimated covariance matrix for variables.

The following statements combine the results for the imputed data sets, and derive both univariate and multivariate inferences about the means. The EDF= option is specified to request that the adjusted degrees of freedom be used in the analysis. For sample means based on 31 observations, the complete-data error degrees of freedom is 30.

```
proc mianalyze data=outcov edf=30;
   modeleffects Oxygen RunTime RunPulse;
run;
```

The "Variance Information" and "Parameter Estimates" tables display the same results as in Output 77.1.2 and Output 77.1.3, respectively, in Example 77.1.

With the WCOV, BCOV, and TCOV options, as in the following statements, the procedure displays the between-imputation covariance matrix, within-imputation covariance matrix, and total covariance matrix assuming that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix in Output 77.2.2.

```
proc mianalyze data=outcov edf=30 wcov bcov tcov mult;
   modeleffects Oxygen RunTime RunPulse;
run;
```

Output 77.2.2 Covariance Matrices

The MIANALYZE Procedure

Within-Imputation Covariance Matrix			
	Oxygen	RunTime	RunPulse
Oxygen	0.925531500	-0.215584249	-0.621865880
RunTime	-0.215584249	0.068197432	0.116221427
RunPulse	-0.621865880	0.116221427	3.345355692

Output 77.2.2 *continued*

Between-Imputation Covariance Matrix			
	Oxygen	**RunTime**	**RunPulse**
Oxygen	0.0260976444	0.0018582126	0.0215847560
RunTime	0.0018582126	0.0029383770	0.0064057721
RunPulse	0.0215847560	0.0064057721	0.5984941329

Total Covariance Matrix			
	Oxygen	**RunTime**	**RunPulse**
Oxygen	1.106311852	-0.257693455	-0.743332446
RunTime	-0.257693455	0.081518163	0.138922492
RunPulse	-0.743332446	0.138922492	3.998790592

With the MULT option, the procedure assumes that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix and displays a multivariate inference for all the parameters taken jointly.

Output 77.2.3 Multivariate Inference

Multivariate Inference Assuming Proportionality of Between/Within Covariance Matrices				
Avg Relative Increase in Variance	**Num DF**	**Den DF**	**F for H0: Parameter=Theta0**	**Pr > F**
0.195326	3	2433.6	13296.2	<.0001

The "Multivariate Inference" table in Output 77.2.3 shows a significant p-value for the null hypothesis that the population means are all equal to zero.

Example 77.3: Reading Regression Results from a DATA= EST Data Set

This example creates an EST-type data set that contains regression coefficients and their corresponding covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the regression model.

The following statements use the REG procedure to generate regression coefficients for each imputed data set:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display (in Output 77.3.1) output regression coefficients and their covariance matrices from PROC REG for the first two imputed data sets:

```
proc print data=outreg(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'REG Model Coefficients and Covariance Matrices'
         ' (First Two Imputations)';
run;
```

Output 77.3.1 EST-Type Data Set

REG Model Coefficients and Covariance Matrices (First Two Imputations)

Obs	_Imputation_	_TYPE_	_NAME_	Intercept	RunTime	RunPulse
1	1	PARMS		86.544	-2.82231	-0.05873
2	1	COV	Intercept	100.145	-0.53519	-0.55077
3	1	COV	RunTime	-0.535	0.10774	-0.00345
4	1	COV	RunPulse	-0.551	-0.00345	0.00343
5	2	PARMS		83.021	-3.00023	-0.02491
6	2	COV	Intercept	79.032	-0.66765	-0.41918
7	2	COV	RunTime	-0.668	0.11456	-0.00313
8	2	COV	RunPulse	-0.419	-0.00313	0.00264

The following statements combine the results for the imputed data sets. The EDF= option is specified to request that the adjusted degrees of freedom be used in the analysis. For a regression model with three independent variables (including the Intercept) and 31 observations, the complete-data error degrees of freedom is 28.

```
proc mianalyze data=outreg edf=28;
   modeleffects Intercept RunTime RunPulse;
run;
```

Output 77.3.2 Variance Information

The MIANALYZE Procedure

Variance Information (25 Imputations)

	Variance						
Parameter	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	22.485821	75.413875	98.799129	19.102	0.310092	0.240234	0.990482
RunTime	0.021126	0.124930	0.146902	21.823	0.175870	0.151147	0.993990
RunPulse	0.000656	0.002622	0.003304	20.042	0.260376	0.209393	0.991694

The "Variance Information" table in Output 77.3.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

The "Parameter Estimates" table in Output 77.3.3 displays the estimated mean and standard error of the regression coefficients. The inferences are based on the t distribution. The table also displays a 95% mean confidence interval and a t test with the associated p-value for the hypothesis that the regression coefficient is equal to zero. Since the p-value for RunPulse is 0.1812, this variable can be removed from the regression model.

Output 77.3.3 Parameter Estimates

Parameter Estimates (25 Imputations)										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	92.700420	9.939775	71.90376	113.4971	19.102	83.020730	100.839807	0	9.33	<.0001
RunTime	-3.030325	0.383278	-3.82557	-2.2351	21.823	-3.280042	-2.754668	0	-7.91	<.0001
RunPulse	-0.079621	0.057482	-0.19951	0.0403	20.042	-0.135862	-0.024910	0	-1.39	0.1812

Example 77.4: Reading Mixed Model Results from PARMS= and COVB= Data Sets

This example creates data sets that contains parameter estimates and covariance matrices computed by a mixed model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the parameters.

The following PROC MIXED statements generate the fixed-effect parameter estimates and covariance matrix for each imputed data set:

```
ods select none;
proc mixed data=outmi;
   model Oxygen= RunTime RunPulse RunTime*RunPulse/solution covb;
   by _Imputation_;
   ods output SolutionF=mixparms CovB=mixcovb;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The ODS OUTPUT statement creates two SAS data sets, named mixparms and mixcovb, from the two ODS tables. The following statements display (in Output 77.4.1) output parameter estimates for the first two imputed data sets:

```
proc print data=mixparms (obs=8);
   var _Imputation_ Effect Estimate StdErr;
   title 'MIXED Model Coefficients (First Two Imputations)';
run;
```

Output 77.4.1 PROC MIXED Model Coefficients

MIXED Model Coefficients (First Two Imputations)

Obs	_Imputation_	Effect	Estimate	StdErr
1	1	Intercept	148.09	81.5231
2	1	RunTime	-8.8115	7.8794
3	1	RunPulse	-0.4123	0.4684
4	1	RunTime*RunPulse	0.03437	0.04517
5	2	Intercept	64.3607	64.6034
6	2	RunTime	-1.1270	6.4307
7	2	RunPulse	0.08160	0.3688
8	2	RunTime*RunPulse	-0.01069	0.03664

The following statements display (in Output 77.4.2) the output covariance matrices associated with the parameter estimates from PROC MIXED for the first two imputed data sets:

```
proc print data=mixcovb (obs=8);
   var _Imputation_ Row Effect Col1 Col2 Col3 Col4;
   title 'Covariance Matrices (First Two Imputations)';
run;
```

Output 77.4.2 PROC MIXED Covariance Matrices

Covariance Matrices (First Two Imputations)

Obs	_Imputation_	Row	Effect	Col1	Col2	Col3	Col4
1	1	1	Intercept	6646.01	-637.40	-38.1515	3.6542
2	1	2	RunTime	-637.40	62.0842	3.6548	-0.3556
3	1	3	RunPulse	-38.1515	3.6548	0.2194	-0.02099
4	1	4	RunTime*RunPulse	3.6542	-0.3556	-0.02099	0.002040
5	2	1	Intercept	4173.59	-411.46	-23.7889	2.3441
6	2	2	RunTime	-411.46	41.3545	2.3414	-0.2353
7	2	3	RunPulse	-23.7889	2.3414	0.1360	-0.01338
8	2	4	RunTime*RunPulse	2.3441	-0.2353	-0.01338	0.001343

Note that the variables Col1, Col2, Col3, and Col4 are used to identify the effects Intercept, RunTime, RunPulse, and RunTime*RunPulse, respectively, through the variable Row.

For univariate inference, only parameter estimates and their associated standard errors are needed. The following statements use the MIANALYZE procedure with the input PARMS= data set to produce univariate results:

```
proc mianalyze parms=mixparms edf=28;
   modeleffects Intercept RunTime RunPulse RunTime*RunPulse;
run;
```

The "Variance Information" table in Output 77.4.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 77.4.3 Variance Information

The MIANALYZE Procedure

Variance Information (25 Imputations)

Parameter	Variance Between	Variance Within	Variance Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	1457.114815	4592.351134	6107.750541	18.748	0.329983	0.251939	0.990023
RunTime	12.937905	43.960426	57.415848	19.175	0.306080	0.237831	0.990576
RunPulse	0.045956	0.152072	0.199867	19.026	0.314288	0.242732	0.990384
RunTime*RunPulse	0.000409	0.001447	0.001872	19.398	0.293964	0.230483	0.990865

The "Parameter Estimates" table in Output 77.4.4 displays the estimated mean and standard error of the regression coefficients.

Output 77.4.4 Parameter Estimates

Parameter Estimates (25 Imputations)							
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum
Intercept	155.866055	78.152099	-7.8574	319.5895	18.748	64.360719	207.413571
RunTime	-9.265145	7.577325	-25.1149	6.5846	19.175	-14.937943	-1.127010
RunPulse	-0.443201	0.447065	-1.3788	0.4924	19.026	-0.738608	0.081597
RunTime*RunPulse	0.035841	0.043272	-0.0546	0.1263	19.398	-0.010690	0.068289

Parameter Estimates (25 Imputations)			
Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	1.99	0.0609
RunTime	0	-1.22	0.2362
RunPulse	0	-0.99	0.3340
RunTime*RunPulse	0	0.83	0.4176

Since each covariance matrix contains variables Row, Col1, Col2, Col3, and Col4 for parameters, the EFFECTVAR=ROWCOL option is needed when you specify the COVB= option. The following statements illustrate the use of the MIANALYZE procedure with input PARMS= and COVB(EFFECTVAR=ROWCOL)= data sets:

```
proc mianalyze parms=mixparms edf=28
               covb(effectvar=rowcol)=mixcovb;
   modeleffects Intercept RunTime RunPulse RunTime*RunPulse;
run;
```

Example 77.5: Reading Generalized Linear Model Results

This example creates data sets that contains parameter estimates and corresponding covariance matrices computed by a generalized linear model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC GENMOD to generate the parameter estimates and covariance matrix for each imputed data set:

```
ods select none;
proc genmod data=outmi;
   model Oxygen= RunTime RunPulse/covb;
   by _Imputation_;
   ods output ParameterEstimates=gmparms
              ParmInfo=gmpinfo
              CovB=gmcovb;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements print (in Output 77.5.1) the output parameter estimates and covariance matrix from PROC GENMOD for the first two imputed data sets:

```
proc print data=gmparms (obs=8);
   var _Imputation_ Parameter Estimate StdErr;
   title 'GENMOD Model Coefficients (First Two Imputations)';
run;
```

Output 77.5.1 PROC GENMOD Model Coefficients

GENMOD Model Coefficients (First Two Imputations)

Obs	_Imputation_	Parameter	Estimate	StdErr
1	1	Intercept	86.5440	9.5107
2	1	RunTime	-2.8223	0.3120
3	1	RunPulse	-0.0587	0.0556
4	1	Scale	2.6692	0.3390
5	2	Intercept	83.0207	8.4489
6	2	RunTime	-3.0002	0.3217
7	2	RunPulse	-0.0249	0.0488
8	2	Scale	2.5727	0.3267

The following statements display the parameter information table in Output 77.5.2. The table identifies parameter names used in the covariance matrices. The parameters Prm1, Prm2, and Prm3 are used for the effects Intercept, RunTime, and RunPulse, respectively, in each covariance matrix.

```
proc print data=gmpinfo (obs=6);
   title 'GENMOD Parameter Information (First Two Imputations)';
run;
```

Output 77.5.2 PROC GENMOD Model Information

GENMOD Parameter Information (First Two Imputations)

Obs	_Imputation_	Parameter	Effect
1	1	Prm1	Intercept
2	1	Prm2	RunTime
3	1	Prm3	RunPulse
4	2	Prm1	Intercept
5	2	Prm2	RunTime
6	2	Prm3	RunPulse

The following statements display (in Output 77.5.3) the output covariance matrices from PROC GENMOD for the first two imputed data sets. Note that the GENMOD procedure computes maximum likelihood estimates for each covariance matrix.

```
proc print data=gmcovb (obs=8);
   var _Imputation_ RowName Prm1 Prm2 Prm3;
   title 'GENMOD Covariance Matrices (First Two Imputations)';
run;
```

Output 77.5.3 PROC GENMOD Covariance Matrices

GENMOD Covariance Matrices (First Two Imputations)

Obs	_Imputation_	RowName	Prm1	Prm2	Prm3
1	1	Prm1	90.453923	-0.483394	-0.497473
2	1	Prm2	-0.483394	0.0973159	-0.003113
3	1	Prm3	-0.497473	-0.003113	0.0030954
4	1	Scale	1.344E-15	-1.09E-17	-6.12E-18
5	2	Prm1	71.383332	-0.603037	-0.378616
6	2	Prm2	-0.603037	0.1034766	-0.002826
7	2	Prm3	-0.378616	-0.002826	0.0023843
8	2	Scale	1.602E-14	1.755E-16	-1.02E-16

The following statements use the MIANALYZE procedure with input PARMS=, PARMINFO=, and COVB= data sets:

```
proc mianalyze parms=gmparms covb=gmcovb parminfo=gmpinfo;
   modeleffects Intercept RunTime RunPulse;
run;
```

Since the GENMOD procedure computes maximum likelihood estimates for the covariance matrix, the EDF= option is not used. The "Parameter Estimates" table in Output 77.5.4 displays the estimated mean and standard error of the regression coefficients.

Output 77.5.4 Parameter Estimates

The MIANALYZE Procedure

Parameter Estimates (25 Imputations)

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	92.700420	9.565616	73.89020	111.5106	367.43	83.020730	100.839807	0	9.69	<.0001
RunTime	-3.030325	0.367167	-3.75093	-2.3097	903.54	-3.280042	-2.754668	0	-8.25	<.0001
RunPulse	-0.079621	0.055231	-0.18815	0.0289	479.31	-0.135862	-0.024910	0	-1.44	0.1501

The resulting parameter estimates are identical to the estimates in Output 77.3.3 in Example 77.3. However, the standard errors are slightly different because in this example, maximum likelihood estimates for the standard errors are combined without the EDF= option, whereas in Example 77.3, unbiased estimates for the standard errors are combined with the EDF= option.

Example 77.6: Reading GLM Results from PARMS= and XPXI= Data Sets

This example creates data sets that contains parameter estimates and corresponding $(\mathbf{X}'\mathbf{X})^{-1}$ matrices computed by a general linear model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC GLM to generate the parameter estimates and $(\mathbf{X}'\mathbf{X})^{-1}$ matrix for each imputed data set:

```
ods select none;
proc glm data=outmi;
   model Oxygen= RunTime RunPulse/inverse;
   by _Imputation_;
   ods output ParameterEstimates=glmparms
              InvXPX=glmxpxi;
quit;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements display (in Output 77.6.1) the output parameter estimates and standard errors from PROC GLM for the first two imputed data sets:

```
proc print data=glmparms (obs=6);
   var _Imputation_ Parameter Estimate StdErr;
   title 'GLM Model Coefficients (First Two Imputations)';
run;
```

Output 77.6.1 PROC GLM Model Coefficients

GLM Model Coefficients (First Two Imputations)

Obs	_Imputation_	Parameter	Estimate	StdErr
1	1	Intercept	86.5440339	10.00726811
2	1	RunTime	-2.8223108	0.32824165
3	1	RunPulse	-0.0587292	0.05854109
4	2	Intercept	83.0207303	8.88996885
5	2	RunTime	-3.0002288	0.33847204
6	2	RunPulse	-0.0249103	0.05137859

The following statements display (in Output 77.6.2) $(\mathbf{X}'\mathbf{X})^{-1}$ matrices from PROC GLM for the first two imputed data sets:

```
proc print data=glmxpxi (obs=8);
   var _Imputation_ Parameter Intercept RunTime RunPulse;
   title 'GLM X''X Inverse Matrices (First Two Imputations)';
run;
```

Output 77.6.2 PROC GLM $(\mathbf{X}'\mathbf{X})^{-1}$ Matrices

GLM X'X Inverse Matrices (First Two Imputations)

Obs	_Imputation_	Parameter	Intercept	RunTime	RunPulse
1	1	Intercept	12.696250656	-0.067849956	-0.069826009
2	1	RunTime	-0.067849956	0.0136594055	-0.000436938
3	1	RunPulse	-0.069826009	-0.000436938	0.0004344762
4	1	Oxygen	86.544033929	-2.822310769	-0.058729234
5	2	Intercept	10.784620785	-0.091107072	-0.057201387
6	2	RunTime	-0.091107072	0.0156332765	-0.000426902
7	2	RunPulse	-0.057201387	-0.000426902	0.0003602208
8	2	Oxygen	83.020730343	-3.000228818	-0.024910305

The standard errors for the estimates in the output Glmparms data set are needed to create the covariance matrix from the $(\mathbf{X}'\mathbf{X})^{-1}$ matrix. The following statements use the MIANALYZE procedure with input PARMS= and XPXI= data sets to produce the same results as displayed in Output 77.3.2 and Output 77.3.3 in Example 77.3:

```
proc mianalyze parms=glmparms xpxi=glmxpxi edf=28;
   modeleffects Intercept RunTime RunPulse;
run;
```

Example 77.7: Reading Logistic Model Results from a PARMS= Data Set

This example creates data sets that contains parameter estimates computed by a logistic regression analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC LOGISTIC to generate the parameter estimates for each imputed data set:

```
ods select none;
proc logistic data=outfish2;
   class Species;
   model Species= Length Width / covb;
   by _Imputation_;
   ods output ParameterEstimates=lgsparms;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements display (in Output 77.7.1) the output logistic regression coefficients from PROC LOGISTIC for the first two imputed data sets:

```
proc print data=lgsparms (obs=6);
   title 'LOGISTIC Model Coefficients (First Two Imputations)';
run;
```

Output 77.7.1 PROC LOGISTIC Model Coefficients

LOGISTIC Model Coefficients (First Two Imputations)

Obs	_Imputation_	Variable	DF	Estimate	StdErr	WaldChiSq	ProbChiSq	_ESTTYPE_
1	1	Intercept	1	0.1637	1.8405	0.0079	0.9291	MLE
2	1	Length	1	1.4543	0.5167	7.9231	0.0049	MLE
3	1	Width	1	-10.2950	3.4860	8.7216	0.0031	MLE
4	2	Intercept	1	0.6473	1.9003	0.1160	0.7334	MLE
5	2	Length	1	1.2831	0.4778	7.2123	0.0072	MLE
6	2	Width	1	-9.2991	3.2187	8.3469	0.0039	MLE

The following statements use the MIANALYZE procedure with input PARMS= data set:

```
proc mianalyze parms=lgsparms;
   modeleffects Intercept Length Width;
run;
```

The "Variance Information" table in Output 77.7.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 77.7.2 Variance Information

The MIANALYZE Procedure

Variance Information (25 Imputations)

Parameter	Variance Between	Variance Within	Variance Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.602599	3.034752	3.661455	819.21	0.206509	0.173178	0.993121
Length	0.077877	0.188227	0.269219	265.18	0.430288	0.306054	0.987906
Width	3.757896	8.382542	12.290754	237.36	0.466232	0.323655	0.987219

The "Parameter Estimates" table in Output 77.7.3 displays the combined parameter estimates with associated standard errors.

Output 77.7.3 Parameter Estimates

Parameter Estimates (25 Imputations)

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	-0.130560	1.913493	-3.8865	3.62537	819.21	-2.321742	0.827356	0	-0.07	0.9456
Length	1.169782	0.518863	0.1482	2.19140	265.18	0.385118	1.554454	0	2.25	0.0250
Width	-8.284998	3.505817	-15.1915	-1.37851	237.36	-11.149787	-2.878900	0	-2.36	0.0189

Example 77.8: Reading Mixed Model Results with Classification Covariates

This example creates data sets that contains parameter estimates and corresponding covariance matrices with classification variables computed by a mixed regression model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC MIXED to generate the parameter estimates and covariance matrix for each imputed data set:

```
ods select none;
proc mixed data=outfish2;
   class Species;
   model Length= Species Width/ solution;
   by _Imputation_;
   ods output SolutionF=mxparms;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements display (in Output 77.8.1) the output mixed model coefficients from PROC MIXED for the first two imputed data sets:

```
proc print data=mxparms (obs=8);
   var _Imputation_ Effect Species Estimate StdErr;
   title 'MIXED Model Coefficients (First Two Imputations)';
run;
```

Output 77.8.1 PROC MIXED Model Coefficients

MIXED Model Coefficients (First Two Imputations)

Obs	_Imputation_	Effect	Species	Estimate	StdErr
1	1	Intercept		4.5106	0.8244
2	1	Species	Parkki	1.5774	0.7020
3	1	Species	Perch	0	.
4	1	Width		5.2585	0.1599
5	2	Intercept		4.5250	0.8771
6	2	Species	Parkki	1.4885	0.7693
7	2	Species	Perch	0	.
8	2	Width		5.2389	0.1701

The following statements use the MIANALYZE procedure with an input PARMS= data set:

```
proc mianalyze parms(classvar=full)=mxparms;
   class Species;
   modeleffects Intercept Species Width;
run;
```

The "Variance Information" table in Output 77.8.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 77.8.2 Variance Information

The MIANALYZE Procedure

Variance Information (25 Imputations)

		Variance						
Parameter	Species	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept		0.065665	0.668667	0.736959	2794.9	0.102131	0.093316	0.996281
Species	Parkki	0.077291	0.525640	0.606023	1364.1	0.152924	0.133909	0.994672
Species	Perch	0	.	.	.	.	.	.
Width		0.002276	0.025876	0.028243	3416	0.091488	0.084356	0.996637

The "Parameter Estimates" table in Output 77.8.3 displays the combined parameter estimates with associated standard errors.

Output 77.8.3 Parameter Estimates

Parameter Estimates (25 Imputations)								
Parameter	**Species**	**Estimate**	**Std Error**	**95% Confidence Limits**		**DF**	**Minimum**	**Maximum**
Intercept		4.519259	0.858463	2.83597	6.202545	2794.9	4.013624	5.162463
Species	Parkki	1.277902	0.778475	-0.24924	2.805039	1364.1	0.647898	1.710559
Species	Perch	0	.	.	.	.	0	0
Width		5.284285	0.168056	4.95478	5.613786	3416	5.206516	5.369231

Parameter Estimates (25 Imputations)				
Parameter	**Species**	**Theta0**	**t for H0: Parameter=Theta0**	**Pr > |t|**
Intercept		0	5.26	<.0001
Species	Parkki	0	1.64	0.1009
Species	Perch	0	.	.
Width		0	31.44	<.0001

Example 77.9: Reading Nominal Logistic Model Results

This example creates data sets to contain parameter estimates that are computed by a nominal logistic regression analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC LOGISTIC to generate the parameter estimates and covariance matrix for each imputed data set:

```
ods select none;
proc logistic data=outfish3;
   class Species;
   model Species= Length Width / link=glogit covb;
   by _Imputation_;
   ods output ParameterEstimates=lgsparms
              CovB=lgscovb;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements display (in Output 77.9.1) the output logistic regression coefficients from PROC LOGISTIC for the first two imputed data sets:

```
proc print data=lgsparms (obs=12);
   title 'LOGISTIC Model Coefficients (First Two Imputations)';
run;
```

Output 77.9.1 PROC LOGISTIC Model Coefficients

LOGISTIC Model Coefficients (First Two Imputations)

Obs	_Imputation_	Variable	Response	DF	Estimate	StdErr	WaldChiSq	ProbChiSq	_ESTTYPE_
1	1	Intercept	Parkki	1	1.7737	1.7712	1.0029	0.3166	MLE
2	1	Intercept	Perch	1	1.1036	1.3426	0.6757	0.4111	MLE
3	1	Length	Parkki	1	-0.0353	0.2700	0.0171	0.8960	MLE
4	1	Length	Perch	1	-0.8560	0.2635	10.5529	0.0012	MLE
5	1	Width	Parkki	1	-0.3784	1.6650	0.0517	0.8202	MLE
6	1	Width	Perch	1	5.6213	1.6333	11.8455	0.0006	MLE
7	2	Intercept	Parkki	1	2.3507	1.7930	1.7188	0.1898	MLE
8	2	Intercept	Perch	1	0.6321	1.3370	0.2235	0.6364	MLE
9	2	Length	Parkki	1	-0.3479	0.2460	2.0004	0.1573	MLE
10	2	Length	Perch	1	-0.6108	0.2130	8.2274	0.0041	MLE
11	2	Width	Parkki	1	1.5786	1.5300	1.0645	0.3022	MLE
12	2	Width	Perch	1	4.1610	1.3110	10.0734	0.0015	MLE

The following statements display the covariance matrices that are associated with parameter estimates derived from the first two imputations in Output 77.9.2:

```
proc print data=lgscovb (obs=12);
   title 'LOGISTIC Model Covariance Matrices (First Two Imputations)';
run;
```

Output 77.9.2 PROC LOGISTIC Covariance Matrices

LOGISTIC Model Covariance Matrices (First Two Imputations)

Obs	_Imputation_	Parameter	Intercept_Parkki	Intercept_Perch	Length_Parkki
1	1	Intercept_Parkki	3.137016	1.150943	-0.25136
2	1	Intercept_Perch	1.150943	1.80259	-0.12448
3	1	Length_Parkki	-0.25136	-0.12448	0.072903
4	1	Length_Perch	-0.11416	-0.16709	0.028705
5	1	Width_Parkki	0.857307	0.557913	-0.43386
6	1	Width_Perch	0.484917	0.676397	-0.16464
7	2	Intercept_Parkki	3.214747	1.25981	-0.19425
8	2	Intercept_Perch	1.25981	1.787564	-0.11454
9	2	Length_Parkki	-0.19425	-0.11454	0.060501
10	2	Length_Perch	-0.10076	-0.13446	0.029263
11	2	Width_Parkki	0.436385	0.460885	-0.35903
12	2	Width_Perch	0.365388	0.463036	-0.17062

Obs	Length_Perch	Width_Parkki	Width_Perch
1	-0.11416	0.857307	0.484917
2	-0.16709	0.557913	0.676397
3	0.028705	-0.43386	-0.16464
4	0.069437	-0.16666	-0.42309
5	-0.16666	2.77239	1.00217
6	-0.42309	1.00217	2.66758
7	-0.10076	0.436385	0.365388
8	-0.13446	0.460885	0.463036
9	0.029263	-0.35903	-0.17062
10	0.04535	-0.17499	-0.27173
11	-0.17499	2.34089	1.081586
12	-0.27173	1.081586	1.718756

The following statements use the MIANALYZE procedure with the input PARMS= and COVB= data sets:

```
proc mianalyze parms(link=glogit)=lgsparms
               covb(effectvar=stacking)=lgscovb
               mult;
   modeleffects Intercept Length Width;
run;
```

The “Variance Information” table in Output 77.9.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 77.9.3 Variance Information

The MIANALYZE Procedure

Variance Information (25 Imputations)								
		Variance						
Parameter	Response	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	Parkki	0.539659	3.540706	4.101951	1282	0.158512	0.138167	0.994504
Intercept	Perch	0.182814	1.502203	1.692330	1901.5	0.126565	0.113278	0.995489
Length	Parkki	0.069216	0.089623	0.161607	120.96	0.803187	0.454374	0.982149
Length	Perch	0.027025	0.047967	0.076073	175.82	0.585945	0.376513	0.985163
Width	Parkki	3.131748	3.792307	7.049325	112.43	0.858849	0.471354	0.981495
Width	Perch	1.079472	1.816671	2.939322	164.52	0.617972	0.389321	0.984666

The "Parameter Estimates" table in Output 77.9.4 displays the combined parameter estimates and their associated standard errors.

Output 77.9.4 Parameter Estimates

Parameter Estimates (25 Imputations)								
Parameter	Response	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum
Intercept	Parkki	1.247968	2.025327	-2.72535	5.22129	1282	-0.450300	2.607579
Intercept	Perch	0.466573	1.300896	-2.08476	3.01791	1901.5	-0.536319	1.129313
Length	Parkki	0.285276	0.402004	-0.51060	1.08115	120.96	-0.347887	0.748875
Length	Perch	-0.566649	0.275813	-1.11098	-0.02232	175.82	-0.856010	-0.254865
Width	Parkki	-2.531763	2.655056	-7.79220	2.72867	112.43	-5.690349	1.578570
Width	Perch	3.835827	1.714445	0.45068	7.22098	164.52	1.818114	5.621285

Parameter Estimates (25 Imputations)				
Parameter	Response	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	Parkki	0	0.62	0.5379
Intercept	Perch	0	0.36	0.7199
Length	Parkki	0	0.71	0.4793
Length	Perch	0	-2.05	0.0414
Width	Parkki	0	-0.95	0.3424
Width	Perch	0	2.24	0.0266

The "Multivariate Inference" table in Output 77.9.5 displays multivariate inference for the parameters assuming proportionality of the between-imputation and within-imputation covariance matrices.

Output 77.9.5 Multivariate Inference

Multivariate Inference Assuming Proportionality of Between/Within Covariance Matrices				
Avg Relative Increase in Variance	Num DF	Den DF	F for H0: Parameter=Theta0	Pr > F
0.321721	6	2317.5	3.14	0.0046

Example 77.10: Using a TEST statement

This example creates a DATA=EST data set to contain regression coefficients and their corresponding covariance matrices that are computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the regression model. A TEST statement is used to test linear hypotheses about the parameters.

The following statements use the REG procedure to generate regression coefficients for each imputed data set:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements combine the results for the imputed data sets. A TEST statement is used to test linear hypotheses of INTERCEPT=0 and RUNTIME=RUNPULSE.

```
proc mianalyze data=outreg edf=28;
   modeleffects Intercept RunTime RunPulse;
   test Intercept, RunTime=RunPulse / mult;
run;
```

The "Test Specification" table in Output 77.10.1 displays the **L** matrix and the **c** vector in a TEST statement. Because no label is specified for the TEST statement, "Test 1" is used as the label.

Output 77.10.1 Test Specification

The MIANALYZE Procedure
Test: Test 1

Test Specification				
	L Matrix			
Parameter	Intercept	RunTime	RunPulse	C
TestPrm1	1.000000	0	0	0
TestPrm2	0	1.000000	-1.000000	0

The "Variance Information" table in Output 77.10.2 displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. A detailed description of these statistics is provided in the section "Combining Inferences from Imputed Data Sets" on page 6096 and the section "Multiple Imputation Efficiency" on page 6097.

Output 77.10.2 Variance Information

Variance Information							
	Variance						
Parameter	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
TestPrm1	22.485821	75.413875	98.799129	19.102	0.310092	0.240234	0.990482
TestPrm2	0.022075	0.136285	0.159243	21.99	0.168452	0.145646	0.994208

The "Parameter Estimates" table in Output 77.10.3 displays the estimated mean and standard error of the linear components. The inferences are based on the t distribution. The table also displays a 95% mean confidence interval and a t test along with the associated p-value for the hypothesis that each linear component of $\mathbf{L}\boldsymbol{\beta}$ is equal to 0.

Output 77.10.3 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	C	t for H0: Parameter=C	Pr > |t|
TestPrm1	92.700420	9.939775	71.90376	113.4971	19.102	83.020730	100.839807	0	9.33	<.0001
TestPrm2	-2.950704	0.399052	-3.77831	-2.1231	21.99	-3.210987	-2.699158	0	-7.39	<.0001

When you specify the MULT option, PROC MIANALYZE assumes that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix and displays a multivariate inference for all the linear components that are taken jointly in Output 77.10.4.

Output 77.10.4 Multivariate Inference

Multivariate Inference
Assuming Proportionality of Between/Within Covariance Matrices

Avg Relative Increase in Variance	Num DF	Den DF	F for H0: Parameter=Theta0	Pr > F
0.238124	2	1114.8	68.41	<.0001

Example 77.11: Combining Correlation Coefficients

This example combines sample correlation coefficients that are computed from a set of imputed data sets by using Fisher's z transformation.

Fisher's z transformation of the sample correlation r is

$$z = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right)$$

The statistic z is approximately normally distributed, with mean

$$\log\left(\frac{1+\rho}{1-\rho}\right)$$

and variance $1/(n-3)$, where ρ is the population correlation coefficient and n is the number of observations.

The following statements use the CORR procedure to compute the correlation r and its associated Fisher's z statistic between the variables Oxygen and RunTime for each imputed data set.

```
ods select none;
proc corr data=outmi fisher(biasadj=no);
   var Oxygen RunTime;
   by _Imputation_;
   ods output FisherPearsonCorr=outz;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The ODS OUTPUT statement is used to save Fisher's z statistic in an output data set. The following statements display the number of observations and Fisher's z statistic for each imputed data set in Output 77.11.1:

```
proc print data=outz (obs=10);
   title 'Fisher''s Correlation Statistics (First 10 Imputations)';
   var _Imputation_ NObs ZVal;
run;
```

Output 77.11.1 Output z Statistics

Fisher's Correlation Statistics (First 10 Imputations)

Obs	_Imputation_	NObs	ZVal
1	1	31	-1.27869
2	2	31	-1.30715
3	3	31	-1.27922
4	4	31	-1.39243
5	5	31	-1.40146
6	6	31	-1.22323
7	7	31	-1.27163
8	8	31	-1.17783
9	9	31	-1.36075
10	10	31	-1.23652

The following statements generate the standard error associated with the z statistic, $1/\sqrt{n-3}$:

```
data outz;
   set outz;
   StdZ= 1. / sqrt(NObs-3);
run;
```

The following statements use the MIANALYZE procedure to generate a combined parameter estimate $\hat{z}$ and its variance, as shown in Output 77.11.2. The ODS OUTPUT statement is used to save the parameter estimates in an output data set.

```
proc mianalyze data=outz;
   ods output ParameterEstimates=parms;
   modeleffects ZVal;
   stderr StdZ;
run;
```

Output 77.11.2 Combining Fisher's z Statistics

The MIANALYZE Procedure

Parameter Estimates (25 Imputations)

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
ZVal	-1.292006	0.202699	-1.68963	-0.89438	1403.6	-1.401459	-1.098356	0	-6.37	<.0001

In addition to the estimate for z, PROC MIANALYZE also generates 95% confidence limits for z, $\hat{z}_{.025}$ and $\hat{z}_{.975}$. The following statements print the estimate and 95% confidence limits for z in Output 77.11.3:

```
proc print data=parms;
   title 'Parameter Estimates with 95% Confidence Limits';
   var Estimate LCLMean UCLMean;
run;
```

Output 77.11.3 Parameter Estimates with 95% Confidence Limits

Parameter Estimates with 95% Confidence Limits

Obs	Estimate	LCLMean	UCLMean
1	-1.292006	-1.68963	-0.89438

An estimate of the correlation coefficient with its corresponding 95% confidence limits is then generated from the following inverse transformation as described in the section "Correlation Coefficients" on page 6101:

$$r = \tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

for $z = \hat{z}$, $\hat{z}_{.025}$, and $\hat{z}_{.975}$.

The following statements generate and display an estimate of the correlation coefficient and its 95% confidence limits, as shown in Output 77.11.4:

```
data corr_ci;
   set parms;
   r=       tanh( Estimate);
   r_lower= tanh( LCLMean);
   r_upper= tanh( UCLMean);
run;
proc print data=corr_ci;
   title 'Estimated Correlation Coefficient'
         ' with 95% Confidence Limits';
   var r r_lower r_upper;
run;
```

Output 77.11.4 Estimated Correlation Coefficient

Estimated Correlation Coefficient with 95% Confidence Limits

Obs	r	r_lower	r_upper
1	-0.85965	-0.93410	-0.71355

Example 77.12: Sensitivity Analysis with Control-Based Pattern Imputation

This example illustrates sensitivity analysis in multiple imputation under the MNAR assumption by creating control-based pattern imputation.

Suppose that a pharmaceutical company is conducting a clinical trial to test the efficacy of a new drug. The trial is conducted with two groups that have an equal number of patients: a treatment group that receives the new drug and a placebo control group. The variable Trt is an indicator variable, with a value of 1 for patients in the treatment group and a value of 0 for patients in the control group. The variable Y0 is the baseline efficacy score, and the variable Y1 is the efficacy score at a follow-up visit.

If the data set does not contain any missing values, then you can use a regression model such as the following to test the treatment effect:

$$Y1 = Trt \;\; Y0$$

Suppose that the variables Trt and Y0 are fully observed and the variable Y1 contains missing values in both the treatment and control groups, as shown in Table 77.4.

Table 77.4 Variables

Variables		
Trt	**Y0**	**Y1**
0	X	X
1	X	X
0	X	.
1	X	.

Suppose the data set Mono1 contains the data from the trial that have missing values in Y1. Output 77.12.1 lists the first 10 observations.

Output 77.12.1 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt	y0	y1
1	0	10.5212	11.3604
2	0	8.5871	8.5178
3	0	9.3274	.
4	0	9.7519	.
5	0	9.3495	9.4369
6	1	11.5192	13.2344
7	1	10.7841	.
8	1	9.7717	10.9407
9	1	10.1455	10.8279
10	1	8.2463	9.6844

Multiple imputation often assumes that missing values are missing at random (MAR), and the following statements use the MI procedure to impute missing values under this assumption:

```
proc mi data=Mono1 seed=14823 nimpute=20 out=outex12a;
   class Trt;
   monotone reg;
   var Trt y0 y1;
run;
```

The following statements generate regression coefficients for each of the 20 imputed data sets:

```
ods select none;
proc reg data=outex12a;
   model y1= Trt y0;
   by  _Imputation_;
   ods output parameterestimates=regparms;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements combine the 20 sets of regression coefficients:

```
proc mianalyze parms=regparms;
   modeleffects Trt;
run;
```

The "Parameter Estimates" table in Output 77.12.2 displays a combined estimate and standard error for the regression coefficient for Trt. The table shows a t test statistic of 3.39, with the associated p-value 0.0008 for the test that the regression coefficient is equal to 0.

Output 77.12.2 Parameter Estimates

The MIANALYZE Procedure

Parameter Estimates (20 Imputations)

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Trt	0.890609	0.262600	0.372745	1.408473	197.29	0.624115	1.228827	0	3.39	0.0008

The conclusion in Output 77.12.2 is based on the MAR assumption. But if missing Y1 values for individuals in the treatment group imply that these individuals no longer receive the treatment, then it is reasonable to assume that the conditional distribution of Y1, given Y0 for individuals who have missing Y1 values in the treatment group, is similar to the corresponding distribution of individuals in the control group.

Ratitch and O'Kelly (2011) describe an implementation of the pattern-mixture model approach that uses a control-based pattern imputation. That is, an imputation model for the missing observations in the treatment group is constructed not from the observed data in the treatment group but rather from the observed data in the control group. This model is also the imputation model that is used to impute missing observations in the control group.

The following statements implement the control-based pattern imputation:

```
proc mi data=Mono1 seed=14823 nimpute=20 out=outex12b;
   class Trt;
   monotone reg;
   mnar model( y1 /modelobs=(Trt='0'));
   var y0 y1;
run;
```

The MNAR statement imputes missing values for scenarios under the MNAR assumption. The MODEL option specifies that only observations where TRT=0 are used to derive the imputation model for the variable Y1. Thus, Y0 and Y1 (but not Trt) are specified in the VAR list.

The following statements generate regression coefficients for each of the 20 imputed data sets:

```
ods select none;
proc reg data=outex12b;
   model y1= Trt y0;
   by _Imputation_;
   ods output parameterestimates=regparms;
run;
ods select all;
```

The following statements combine the 20 sets of regression coefficients:

```
proc mianalyze parms=regparms;
   modeleffects Trt;
run;
```

Output 77.12.3 Parameter Estimates

The MIANALYZE Procedure

Parameter Estimates (20 Imputations)

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Trt	0.708802	0.279447	0.157975	1.259628	213.68	0.329363	0.919297	0	2.54	0.0119

The "Parameter Estimates" table in Output 77.12.3 shows a t test statistic of 2.54, with the p-value 0.0119 for the test that the parameter is equal to 0. Thus, for a two-sided Type I error level of 0.05, the significance of the treatment effect is not reversed by control-based pattern imputation.

Example 77.13: Sensitivity Analysis with the Tipping-Point Approach

This example illustrates sensitivity analysis in multiple imputation under the MNAR assumption in which you search for a tipping point that reverses the study conclusion.

Suppose that a pharmaceutical company is conducting a clinical trial to test the efficacy of a new drug. The trial is conducted with two groups that have an equal number of patients: a treatment group that receives the new drug and a placebo control group. The variable Trt is an indicator variable, with a value of 1 for patients in the treatment group and a value of 0 for patients in the control group. The variable Y0 is the baseline efficacy score, and the variable Y1 is the efficacy score at a follow-up visit.

If the data set does not contain any missing values, then you can use a regression model such as the following to test the efficacy of the treatment effect:

$$\mathrm{Y1} = \mathrm{Trt}\ \ \mathrm{Y0}$$

Now suppose in a trial that the variables Trt and Y0 are fully observed and the variable Y1 contains missing values in both the treatment and control groups. The data set Mono2 contains the data from the trial and Figure 77.13.1 lists the first 10 observations.

Output 77.13.1 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt	y0	y1
1	0	11.4826	.
2	0	10.0090	10.8667
3	0	11.3643	10.6660
4	0	11.3098	10.8297
5	0	11.3094	.
6	1	10.3815	10.5587
7	1	11.2001	13.7616
8	1	9.7002	10.3460
9	1	10.0801	.
10	1	11.2667	11.0634

Multiple imputation often assumes that missing values are missing at random (MAR), and the following statements use the MI procedure to impute missing values under this assumption:

```
proc mi data=Mono2 seed=14823 out=outmi;
   class Trt;
   monotone reg;
   var Trt y0 y1;
run;
```

The following statements generate regression coefficients for each of the 25 imputed data sets:

```
ods select none;
proc reg data=outmi;
   model y1= Trt y0;
   by  _Imputation_;
   ods output parameterestimates=regparms;
run;
ods select all;
```

Because of the ODS SELECT statements, no output is displayed. The following statements combine the 20 sets of regression coefficients:

```
proc mianalyze parms=regparms;
   modeleffects Trt;
run;
```

The “Parameter Estimates” table in Output 77.13.2 displays a combined estimate and standard error for the regression coefficient for Trt. The table displays a 95% confidence interval (0.3353, 1.3208), which does not contain 0. The table also shows a t statistic of 3.31, with the associated p-value 0.0011 for the test that the regression coefficient is equal to 0.

Output 77.13.2 Parameter Estimates

The MIANALYZE Procedure

Parameter Estimates (25 Imputations)										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Trt	0.828052	0.249933	0.335262	1.320842	203.53	0.556144	1.124339	0	3.31	0.0011

The conclusion in Output 77.13.2 is based on the MAR assumption. But if it is plausible that, for the treatment group, the distribution of missing Y1 responses has a lower expected value than that of the corresponding distribution of the observed Y1 responses, the conclusion under the MAR assumption should be examined.

The following macro performs multiple imputation analysis for a specified sequence of shift parameters, which adjust the imputed values for observations in the treatment group (TRT=1):

```
/*-------------------------------------------------------*/
/*--- Performs multiple imputation analysis            ---*/
/*--- for specified shift parameters:                  ---*/
/*--- data= input data set                             ---*/
/*--- smin= min shift parameter                        ---*/
/*--- smax= max shift parameter                        ---*/
/*--- sinc= increment of the shift parameter           ---*/
/*--- outparms= output reg parameters                  ---*/
/*-------------------------------------------------------*/
%macro miparms( data=, smin=, smax=, sinc=, outparms=);
ods select none;

data &outparms;
   set _null_;
run;

/*------------ # of shift values ------------*/
%let ncase= %sysevalf( (&smax-&smin)/&sinc, ceil );

/*---- Multiple imputation analysis for each shift ----*/
%do jc=0 %to &ncase;
   %let sj= %sysevalf( &smin + &jc * &sinc);

  /*---- Generates 25 imputed data sets ----*/
   proc mi data=&data seed=14823 out=outmi;
      class Trt;
      monotone reg;
      mnar adjust( y1 / shift=&sj  adjustobs=(Trt='1') );
      var Trt y0 y1;
   run;

  /*------ Perform reg test -------*/
   proc reg data=outmi;
      model y1= Trt y0;
      by  _Imputation_;
      ods output parameterestimates=regparm;
   run;
```

```
  /*------ Combine reg results -------*/
   proc mianalyze parms=regparm;
      modeleffects Trt;
      ods output parameterestimates=miparm;
   run;

   data miparm;
      set miparm;
      Shift= &sj;
      keep Shift Probt;
   run;

  /*----- Output multiple imputation results ----*/
   data &outparms;
      set &outparms miparm;
   run;

%end;

ods select all;
%mend miparms;
```

Assume that the tipping point for the shift parameter that reverses the study conclusion is between –3 and 0. The following statement performs multiple imputation analysis for each of the shift parameters –3.0, –2.9, . . . , 0:

```
%miparms( data=Mono2, smin=-3, smax=0, sinc=0.1, outparms=parm1);
```

The following statements display the plot of *p*-values for shift parameters between –3 and 0:

```
title 'P-value Plot';
proc sgplot data=parm1;
   series x=Shift y=Probt / lineattrs=graphfit(color=blue);
   refline 0.05 / axis=y lineattrs=graphfit(thickness=1px pattern=shortdash);
   xaxis label="Shift Parameter";
   yaxis label="P-value";
run;
```

Output 77.13.3 Finding Tipping Point for Shift Parameter between –3 and 0

For a two-sided Type I error level of 0.05, the tipping point for the shift parameter is slightly less than –1.5. The following statement performs multiple imputation analysis for shift parameters –1.60, –1.59, . . . , –1.50:

```
%miparms( data=Mono2, smin=-1.6, smax=-1.5, sinc=0.01, outparms=parm2);
```

The following statements display the *p*-values that are associated with the shift parameters:

```
proc print label data=parm2;
   var Shift Probt;
   title 'P-values for Shift Parameters';
   label Probt='Pr > |t|';
   format Probt 8.4;
run;
```

Output 77.13.4 Finding Tipping Point for Shift Parameter between –1.6 and –1.5

P-values for Shift Parameters

Obs	Shift	Pr > |t|
1	-1.60	0.0586
2	-1.59	0.0574
3	-1.58	0.0562
4	-1.57	0.0551
5	-1.56	0.0539
6	-1.55	0.0528
7	-1.54	0.0517
8	-1.53	0.0507
9	-1.52	0.0496
10	-1.51	0.0486
11	-1.50	0.0476

The study conclusion under MAR is reversed when the shift parameter is –1.53. Thus, if the shift parameter –1.53 is plausible, the conclusion under MAR is questionable.

References

Allison, P. D. (2000). "Multiple Imputation for Missing Data: A Cautionary Tale." *Sociological Methods and Research* 28:301–309.

Allison, P. D. (2001). *Missing Data*. Thousand Oaks, CA: Sage Publications.

Barnard, J., and Rubin, D. B. (1999). "Small-Sample Degrees of Freedom with Multiple Imputation." *Biometrika* 86:948–955.

Cochran, W. G. (1977). *Sampling Techniques*. 3rd ed. New York: John Wiley & Sons.

Gadbury, G. L., Coffey, C. S., and Allison, D. B. (2003). "Modern Statistical Methods for Handling Missing Repeated Measurements in Obesity Trial Data: Beyond LOCF." *Obesity Reviews* 4:175–184.

Horton, N. J., and Lipsitz, S. R. (2001). "Multiple Imputation in Practice: Comparison of Software Packages for Regression Models with Missing Variables." *American Statistician* 55:244–254.

Li, K. H., Raghunathan, T. E., and Rubin, D. B. (1991). "Large-Sample Significance Levels from Multiply Imputed Data Using Moment-Based Statistics and an F Reference Distribution." *Journal of the American Statistical Association* 86:1065–1073.

Little, R. J. A., and Rubin, D. B. (2002). *Statistical Analysis with Missing Data*. 2nd ed. Hoboken, NJ: John Wiley & Sons.

Ratitch, B., and O'Kelly, M. (2011). "Implementation of Pattern-Mixture Models Using Standard SAS/STAT Procedures." In *Proceedings of PharmaSUG 2011 (Pharmaceutical Industry SAS Users Group)*. Paper SP04. Cary, NC: SAS Institute Inc.

Rubin, D. B. (1976). "Inference and Missing Data." *Biometrika* 63:581–592.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley & Sons.

Rubin, D. B. (1996). "Multiple Imputation after 18+ Years." *Journal of the American Statistical Association* 91:473–489.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. New York: Chapman & Hall.

Subject Index

adjusted degrees of freedom
MIANALYZE procedure, 6097
average relative increase in variance
MIANALYZE procedure, 6099

between-imputation covariance matrix
MIANALYZE procedure, 6098
between-imputation variance
MIANALYZE procedure, 6096

combining inferences
MIANALYZE procedure, 6096
control-based pattern imputation
MIANALYZE procedure, 6127

degrees of freedom
MIANALYZE procedure, 6097, 6099

fraction of missing information
MIANALYZE procedure, 6097

input data sets
MIANALYZE procedure, 6092

MIANALYZE procedure
adjusted degrees of freedom, 6097
average relative increase in variance, 6099
between-imputation covariance matrix, 6098
between-imputation variance, 6096
combining inferences, 6096
control-based pattern imputation, 6127
degrees of freedom, 6097, 6099
fraction of missing information, 6097
input data sets, 6092
introductory example, 6083
multiple imputation efficiency, 6097
multivariate inferences, 6098
ODS table names, 6102
relative efficiency, 6097
relative increase in variance, 6097
sensitivity analysis, 6105, 6127, 6130
syntax, 6085
testing linear hypotheses, 6090, 6099
tipping-point approach, 6130
total covariance matrix, 6098, 6099
total variance, 6096
within-imputation covariance matrix, 6098
within-imputation variance, 6096
multiple imputation efficiency
MIANALYZE procedure, 6097
multiple imputations analysis, 6082
multivariate inferences
MIANALYZE procedure, 6098

relative efficiency
MIANALYZE procedure, 6097
relative increase in variance
MIANALYZE procedure, 6097

sensitivity analysis
MIANALYZE procedure, 6105, 6127, 6130

testing linear hypotheses
MIANALYZE procedure, 6090, 6099
tipping-point approach
MIANALYZE procedure, 6130
total covariance matrix
MIANALYZE procedure, 6098, 6099
total variance
MIANALYZE procedure, 6096

within-imputation covariance matrix
MIANALYZE procedure, 6098
within-imputation variance
MIANALYZE procedure, 6096

Syntax Index

ALPHA= option
    PROC MIANALYZE statement, 6086

BCOV option
    PROC MIANALYZE statement, 6086
    TEST statement (MIANALYZE), 6091
BY statement
    MIANALYZE procedure, 6089

CLASS statement
    MIANALYZE procedure, 6089
CLASSVAR= option
    PROC MIANALYZE statement, 6088
COVB= option
    PROC MIANALYZE statement, 6087

DATA= option
    PROC MIANALYZE statement, 6087

EDF= option
    PROC MIANALYZE statement, 6087
EFFECTVAR= option
    PROC MIANALYZE statement, 6087

LINK= option
    PROC MIANALYZE statement, 6088

MIANALYZE procedure, BY statement, 6089
MIANALYZE procedure, CLASS statement, 6089
MIANALYZE procedure, MODELEFFECTS statement, 6089
MIANALYZE procedure, PROC MIANALYZE statement, 6086
    ALPHA= option, 6086
    BCOV option, 6086
    CLASSVAR= option, 6088
    COVB= option, 6087
    DATA= option, 6087
    EDF= option, 6087
    EFFECTVAR= option, 6087
    LINK= option, 6088
    MU0= option, 6088
    MULT option, 6087
    PARMINFO= option, 6087
    PARMS= option, 6088
    TCOV option, 6088
    THETA0= option, 6088
    WCOV option, 6088
    XPXI= option, 6088
MIANALYZE procedure, STDERR statement, 6090
MIANALYZE procedure, TEST statement, 6090
    BCOV option, 6091
    MULT option, 6091
    TCOV option, 6091
    WCOV option, 6091
MODELEFFECTS statement
    MIANALYZE procedure, 6089
MU0= option
    PROC MIANALYZE statement, 6088
MULT option
    PROC MIANALYZE statement, 6087
    TEST statement (MIANALYZE), 6091

PARMINFO= option
    PROC MIANALYZE statement, 6087
PARMS= option
    PROC MIANALYZE statement, 6088
PROC MIANALYZE statement, *see* MIANALYZE procedure

STDERR statement
    MIANALYZE procedure, 6090

TCOV option
    PROC MIANALYZE statement, 6088
    TEST statement (MIANALYZE), 6091
TEST statement
    MIANALYZE procedure, 6090
THETA0= option
    PROC MIANALYZE statement, 6088

WCOV option
    PROC MIANALYZE statement, 6088
    TEST statement (MIANALYZE), 6091

XPXI= option
    PROC MIANALYZE statement, 6088