SAS/STAT® 14.3 User's Guide

The MI Procedure

This document is an individual chapter from *SAS/STAT® 14.3 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS/STAT® 14.3 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 14.3 User's Guide

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

September 2017

Chapter 77
The MI Procedure

Contents

Overview: MI Procedure **6064**
Getting Started: MI Procedure **6067**
Syntax: MI Procedure **6070**
PROC MI Statement 6071
BY Statement 6075
CLASS Statement 6075
EM Statement 6075
FCS Statement 6076
FREQ Statement 6081
MCMC Statement 6082
MNAR Statement 6086
MONOTONE Statement 6088
TRANSFORM Statement 6092
VAR Statement 6093
Details: MI Procedure **6093**
Descriptive Statistics 6093
EM Algorithm for Data with Missing Values 6094
Statistical Assumptions for Multiple Imputation 6096
Missing Data Patterns 6096
Imputation Methods 6098
Monotone Methods for Data Sets with Monotone Missing Patterns 6099
Monotone and FCS Regression Methods 6100
Monotone and FCS Predictive Mean Matching Methods 6101
Monotone and FCS Discriminant Function Methods 6102
Monotone and FCS Logistic Regression Methods 6104
Monotone Propensity Score Method 6108
FCS Methods for Data Sets with Arbitrary Missing Patterns 6108
Checking Convergence in FCS Methods 6110
MCMC Method for Arbitrary Missing Multivariate Normal Data 6111
Producing Monotone Missingness with the MCMC Method 6115
MCMC Method Specifications 6117
Checking Convergence in MCMC 6118
Input Data Sets 6120
Output Data Sets 6121
Combining Inferences from Multiply Imputed Data Sets 6123
Multiple Imputation Efficiency 6124

Number of Imputations 6125
Imputer's Model Versus Analyst's Model 6126
Parameter Simulation versus Multiple Imputation 6127
Sensitivity Analysis for the MAR Assumption 6127
Multiple Imputation with Pattern-Mixture Models 6128
Specifying Sets of Observations for Imputation in Pattern-Mixture Models 6130
Adjusting Imputed Values in Pattern-Mixture Models 6131
Summary of Issues in Multiple Imputation 6135
Plot Options Superseded by ODS Graphics 6136
ODS Table Names 6141
ODS Graphics 6142
Examples: MI Procedure **6143**
Example 77.1: EM Algorithm for MLE 6145
Example 77.2: Monotone Propensity Score Method 6148
Example 77.3: Monotone Regression Method 6150
Example 77.4: Monotone Logistic Regression Method for CLASS Variables 6153
Example 77.5: Monotone Discriminant Function Method for CLASS Variables 6155
Example 77.6: FCS Methods for Continuous Variables 6157
Example 77.7: FCS Method for CLASS Variables 6160
Example 77.8: FCS Method with Trace Plot 6164
Example 77.9: MCMC Method 6168
Example 77.10: Producing Monotone Missingness with MCMC 6170
Example 77.11: Checking Convergence in MCMC 6172
Example 77.12: Saving and Using Parameters for MCMC 6174
Example 77.13: Transforming to Normality 6176
Example 77.14: Multistage Imputation 6178
Example 77.15: Creating Control-Based Pattern Imputation in Sensitivity Analysis . . 6181
Example 77.16: Adjusting Imputed Continuous Values in Sensitivity Analysis 6184
Example 77.17: Adjusting Imputed Classification Levels in Sensitivity Analysis . . . 6187
Example 77.18: Adjusting Imputed Values with Parameters in a Data Set 6190
References **6194**

Overview: MI Procedure

Missing values are an issue in a substantial number of statistical analyses. Most SAS statistical procedures exclude observations with any missing variable values from the analysis. These observations are called incomplete cases. While using only complete cases is simple, you lose information that is in the incomplete cases. Excluding observations with missing values also ignores the possible systematic difference between the complete cases and incomplete cases, and the resulting inference might not be applicable to the population of all cases, especially with a smaller number of complete cases.

Some SAS procedures use all the available cases in an analysis—that is, cases with useful information. For example, the CORR procedure estimates a variable mean by using all cases with nonmissing values for this

variable, ignoring the possible missing values in other variables. The CORR procedure also estimates a correlation by using all cases with nonmissing values for this pair of variables. This estimation might make better use of the available data, but the resulting correlation matrix might not be positive definite.

Another strategy is single imputation, in which you substitute a value for each missing value. Standard statistical procedures for complete data analysis can then be used with the filled-in data set. For example, each missing value can be imputed from the variable mean of the complete cases. This approach treats missing values as if they were known in the complete-data analyses. Single imputation does not reflect the uncertainty about the predictions of the unknown missing values, and the resulting estimated variances of the parameter estimates are biased toward zero (Rubin 1987, p. 13).

Instead of filling in a single value for each missing value, multiple imputation replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute (Rubin 1976, 1987). The multiply imputed data sets are then analyzed by using standard procedures for complete data and combining the results from these analyses. No matter which complete-data analysis is used, the process of combining results from different data sets is essentially the same.

Multiple imputation does not attempt to estimate each missing value through simulated values, but rather to represent a random sample of the missing values. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values; for example, valid confidence intervals for parameters.

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 by using standard procedures.
3. The results from the m complete data sets are combined for the inference.

The MI procedure is a multiple imputation procedure that creates multiply imputed data sets for incomplete p-dimensional multivariate data. It uses methods that incorporate appropriate variability across the m imputations. The imputation method of choice depends on the patterns of missingness in the data and the type of the imputed variable.

A data set with variables $Y_1, Y_2, \ldots, Y_p$ (in that order) is said to have a *monotone missing pattern* when the event that a variable Y_j is missing for a particular individual implies that all subsequent variables $Y_k, k > j$, are missing for that individual.

For data sets with monotone missing patterns, the variables with missing values can be imputed sequentially with covariates constructed from their corresponding sets of preceding variables. To impute missing values for a continuous variable, you can use a regression method (Rubin 1987, pp. 166–167), a predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996), or a propensity score method (Rubin 1987, pp. 124, 158; Lavori, Dawson, and Shera 1995). To impute missing values for a classification variable, you can use a logistic regression method when the classification variable has a binary, nominal, or ordinal response, or you can use a discriminant function method when the classification variable has a binary or nominal response.

For data sets with arbitrary missing patterns, you can use either of the following methods to impute missing values: a Markov chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality, or a fully conditional specification (FCS) method (Brand 1999; Van Buuren 2007) that assumes the existence of a joint distribution for all variables.

You can use the MCMC method to impute either all the missing values or just enough missing values to make the imputed data sets have monotone missing patterns. With a monotone missing data pattern, you have greater flexibility in your choice of imputation models, such as the monotone regression method that do not use Markov chains. You can also specify a different set of covariates for each imputed variable.

An FCS method does not start with an explicitly specified multivariate distribution for all variables, but rather uses a separate conditional distribution for each imputed variable. For each imputation, the process contains two phases: the preliminary filled-in phase followed by the imputation phase. At the filled-in phase, the missing values for all variables are filled in sequentially over the variables taken one at a time. These filled-in values provide starting values for these missing values at the imputation phase. At the imputation phase, the missing values for each variable are imputed sequentially for a number of burn-in iterations before the imputation.

As in methods for data sets with monotone missing patterns, you can use a regression method or a predictive mean matching method to impute missing values for a continuous variable, a logistic regression method to impute missing values for a classification variable with a binary or ordinal response, and a discriminant function method to impute missing values for a classification variable with a binary or nominal response.

After the *m* complete data sets are analyzed using standard SAS procedures, you can use the MIANALYZE procedure to generate valid statistical inferences about these parameters by combining results from the *m* analyses.

The number of imputations, *m*, must be specified in advance. The relative efficiency of an estimator based on a small number of imputations is high for cases with modest missing information (Rubin 1987, p. 114), and often a value of *m* as low as three or five is adequate (Rubin 1996, p. 480). For more information about relative efficiency, see the section "Multiple Imputation Efficiency" on page 6124.

Although a small number of imputations can suffice for high relative efficiency, they might not be adequate for other aspects of inference, such as confidence intervals and *p*-values. Recent studies examine these aspects and recommend much larger values of *m* than the traditionally advised values of three to five (Allison 2012; Van Buuren 2012, pp. 49–50). For more information, see the section "Number of Imputations" on page 6125.

Multiple imputation inference assumes that the model (variables) you used to analyze the multiply imputed data (the analyst's model) is the same as the model used to impute missing values in multiple imputation (the imputer's model). But in practice, the two models might not be the same. The consequences for different scenarios (Schafer 1997, pp. 139–143) are discussed in the section "Imputer's Model Versus Analyst's Model" on page 6126.

Multiple imputation usually assumes that the data are missing at random (MAR). That is, for a variable Y, the probability that an observation is missing depends only on the observed values of other variables, not on the unobserved values of Y. The MAR assumption cannot be verified, because the missing values are not observed. For a study that assumes MAR, the sensitivity of inferences to departures from the MAR assumption should be examined.

The pattern-mixture model approach to sensitivity analysis models the distribution of a response as the mixture of a distribution of the observed responses and a distribution of the missing responses. Missing values can then be imputed under a plausible scenario for which the missing data are missing not at random (MNAR). If this scenario leads to a conclusion that is different from inference under MAR, then the conclusion under MAR is not robust to MNAR.

Getting Started: MI Procedure

The Fitness data described in the REG procedure are measurements of 31 individuals in a physical fitness course. See Chapter 100, "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. Certain values have been set to     |
| missing and the resulting data set has an arbitrary missing pattern. |
| Only selected variables of                                           |
| Oxygen (intake rate, ml per kg body weight per minute),              |
| Runtime (time to run 1.5 miles in minutes),                          |
| RunPulse (heart rate while running) are used.                        |
*----------------------------------------------------------------------*;
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 the missing data are missing at random (MAR). That is, the probability that an observation is missing can depend on the observed variable values of the individual, but not on the missing variable values of the individual. See the section "Statistical Assumptions for Multiple Imputation" on page 6096 for a detailed description of the MAR assumption.

The following statements invoke the MI procedure and impute missing values for the Fitness1 data set:

```
proc mi data=Fitness1 seed=501213 mu0=50 10 180 out=outmi;
   mcmc;
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Figure 77.1 describes the method used in the multiple imputation process. By default, the MCMC statement uses the Markov chain Monte Carlo (MCMC) method with a single chain to

create 25 imputations. The posterior mode, the highest observed-data posterior density, with a noninformative prior, is computed from the expectation-maximization (EM) algorithm and is used as the starting value for the chain.

Figure 77.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	25
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	501213

The MI procedure takes 200 burn-in iterations before the first imputation and 100 iterations between imputations. In a Markov chain, the information in the current iteration influences the state of the next iteration. The burn-in iterations are iterations in the beginning of each chain that are used both to eliminate the series of dependence on the starting value of the chain and to achieve the stationary distribution. The between-imputation iterations in a single chain are used to eliminate the series of dependence between the two imputations.

The "Missing Data Patterns" table in Figure 77.2 lists distinct missing data patterns with their corresponding frequencies and percentages. An "X" means that the variable is observed in the corresponding group, and a "." means that the variable is missing. The table also displays group-specific variable means. The MI procedure sorts the data into groups based on whether the analysis variables are observed or missing. For a detailed description of missing data patterns, see the section "Missing Data Patterns" on page 6096.

Figure 77.2 Missing Data Patterns

Missing Data Patterns

						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	21	67.74	46.353810	10.809524	171.666667
2	X	X	.	4	12.90	47.109500	10.137500	.
3	X	.	.	3	9.68	52.461667	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	.	2	6.45	.	9.885000	.

After the completion of *m* imputations, the "Variance Information" table in Figure 77.3 displays the between-imputation variance, within-imputation variance, and total variance 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 (in units of variance) for each variable are also displayed. A detailed description of these statistics is provided in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Figure 77.3 Variance Information

Variance Information (25 Imputations)

	Variance						
Variable	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.037126	0.936472	0.975084	27.018	0.041231	0.039724	0.998414
RunTime	0.001317	0.065716	0.067086	27.593	0.020843	0.020451	0.999183
RunPulse	1.386290	3.394043	4.835784	18.43	0.424786	0.303282	0.988014

The "Parameter Estimates" table in Figure 77.4 displays the estimated mean and standard error of the mean for each variable. The inferences are based on the *t* distribution. The table also displays a 95% confidence interval for the mean and a *t* statistic with the associated *p*-value for the hypothesis that the population mean is equal to the value specified with the MU0= option. A detailed description of these statistics is provided in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Figure 77.4 Parameter Estimates

Parameter Estimates (25 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Oxygen	47.100050	0.987463	45.0740	49.1261	27.018	46.774347	47.434726	50.000000	-2.94	0.0067
RunTime	10.564553	0.259010	10.0336	11.0955	27.593	10.472584	10.636629	10.000000	2.18	0.0380
RunPulse	171.490381	2.199042	166.8781	176.1027	18.43	169.175377	173.421951	180.000000	-3.87	0.0011

In addition to the output tables, the procedure also creates a data set with imputed values. The imputed data sets are stored in the Outmi data set, with the index variable _Imputation_ indicating the imputation numbers. The data set can now be analyzed using standard statistical procedures with _Imputation_ as a BY variable.

The following statements list the first 10 observations of data set Outmi:

```
proc print data=outmi (obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

The table in Figure 77.5 shows that the precision of the imputed values differs from the precision of the observed values. You can use the ROUND= option to make the imputed values consistent with the observed values.

Figure 77.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	RunPulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	8.0747	155.925
5	1	49.8740	9.2200	176.837
6	1	44.8110	11.6300	176.000
7	1	42.8857	11.9500	176.000
8	1	46.9992	10.8500	173.099
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

Syntax: MI Procedure

The following statements are available in the MI procedure:

PROC MI < *options* > **;**
BY *variables* **;**
CLASS *variables* **;**
EM < *options* > **;**
FCS < *options* > **;**
FREQ *variable* **;**
MCMC < *options* > **;**
MNAR *options* **;**
MONOTONE < *options* > **;**
TRANSFORM *transform* **(***variables*< */ options* >**)** < *. . . transform* **(***variables*< */ options* >**)** > **;**
VAR *variables* **;**

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

The CLASS statement lists the classification variables in the VAR statement. If the MNAR statement is specified, the CLASS statement also includes the identification variables in the MNAR statement. Classification variables can be either character or numeric.

The EM statement uses the EM algorithm to compute the maximum likelihood estimate (MLE) of the data with missing values, assuming a multivariate normal distribution for the data.

The FREQ statement specifies the variable that represents the frequency of occurrence for other values in the observation.

For a data set with a monotone missing pattern, you can use the MONOTONE statement to specify applicable monotone imputation methods; otherwise, you can use either the MCMC statement assuming multivariate normality or the FCS method assuming a joint distribution for variables exists. Note that you can specify no more than one of these statements. When none of these three statements is specified, the MCMC method with its default options is used.

The FCS statement uses a multivariate imputation by chained equations method to impute values for a data set with an arbitrary missing pattern, assuming a joint distribution exists for the data.

The MCMC statement uses a Markov chain Monte Carlo method to impute values for a data set with an arbitrary missing pattern, assuming a multivariate normal distribution for the data.

The MNAR statement imputes missing values, assuming that the missing data are missing not at random (MNAR). The MNAR statement is applicable only if you also specify either an FCS or MONOTONE statement.

The MONOTONE statement specifies monotone methods to impute continuous and classification variables for a data set with a monotone missing pattern.

The TRANSFORM statement specifies the variables to be transformed before the imputation process; the imputed values of these transformed variables are reverse-transformed to the original forms before the imputation.

The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables not listed in other statements are used.

The PROC MI statement is the only required statement for the MI procedure. The rest of this section provides detailed syntax information for each of these statements, beginning with the PROC MI statement. The remaining statements are presented in alphabetical order.

PROC MI Statement

PROC MI < *options* > **;**

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

Table 77.1 Summary of PROC MI Options

Option	Description
Data Sets	
DATA=	Specifies the input data set
OUT=	Specifies the output data set with imputed values
Imputation Details	
NIMPUTE=	Specifies the number of imputations
SEED=	Specifies the seed to begin random number generator
ROUND=	Specifies units to round imputed variable values
MAXIMUM=	Specifies maximum values for imputed variable values
MINIMUM=	Specifies minimum values for imputed variable values
MINMAXITER=	Specifies the maximum number of iterations to impute values in the specified range
SINGULAR=	Specifies the singularity criterion
Statistical Analysis	
ALPHA=	Specifies the level for the confidence interval, $(1 - \alpha)$
MU0=	Specifies means under the null hypothesis

Table 77.1 *continued*

Option	Description
Printed Output	
DISPLAYPATTERN=	Displays missing data patterns table
NOPRINT	Suppresses all displayed output
SIMPLE	Displays univariate statistics and correlations

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

ALPHA=*α*

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

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC MI. By default, the procedure uses the most recently created SAS data set.

DISPLAYPATTERN=ALL | NOMEANS | NONE

requests (except when DISPLAYPATTERN=NONE is specified) a missing data patterns table:

ALL displays both the missing data patterns and the group means in the table.

NOMEANS displays only the missing data patterns in the table.

NONE does not display the missing data patterns table.

By default, DISPLAYPATTERN=ALL.

MAXIMUM=*numbers*

specifies maximum values for imputed variables. When an intended imputed value is greater than the maximum, PROC MI redraws another value for imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default number is a missing value, which indicates no restriction on the maximum for the corresponding variable

The MAXIMUM= option is related to the MINIMUM= and ROUND= options, which are used to make the imputed values more consistent with the observed variable values. These options apply only if you use the MCMC method, the monotone regression method, or the FCS regression method. For more information about these methods, see the section "Imputation Methods" on page 6098.

When you specify a maximum for the first variable only, you must also specify a missing value after the maximum. Otherwise, the maximum is used for all variables. For example, "MAXIMUM= 100 ." sets a maximum of 100 only for the first analysis variable and no maximum for the remaining variables. "MAXIMUM= . 100" sets a maximum of 100 only for the second analysis variable and no maximum for the other variables.

MINIMUM=*numbers*

specifies the minimum values for imputed variables. When an intended imputed value is less than the minimum, PROC MI redraws another value for imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default number is a missing value, which indicates no restriction on the minimum for the corresponding variable

MINMAXITER=*number*

specifies the maximum number of iterations for imputed values to be in the specified range when the option MINIMUM or MAXIMUM is also specified. The default is MINMAXITER=100.

MU0=*numbers*

THETA0=*numbers*

specifies the parameter values $\boldsymbol{\mu}_0$ under the null hypothesis $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ for the population means corresponding to the analysis variables. Each hypothesis is tested with a *t* test. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default is MU0=0.

If a variable is transformed as specified in a TRANSFORM statement, then the same transformation for that variable is also applied to its corresponding specified MU0= value in the *t* test. If the parameter values $\boldsymbol{\mu}_0$ for a transformed variable are not specified, then a value of zero is used for the resulting $\boldsymbol{\mu}_0$ after transformation.

NIMPUTE=*n* | **PCTMISSING < (** *range-options* **) >**

specifies the number of imputations. NIMPUTE=*n* specifies the number explicitly, and NIMPUTE=PCTMISSING uses the percentage of incomplete cases as the number of imputations. By default, NIMPUTE=25.

When you specify NIMPUTE=PCTMISSING, the number of imputations is the resulting percentage rounded up to an integer. You can use the following *range-options* to set the range for the number of imputations:

MIN=*min*

specifies the minimum number of imputations, $2 \leq min \leq 100$. If the resulting number of imputations is less than *min*, then *min* is used. By default, MIN=5.

MAX=*max*

specifies the maximum number of imputations, $2 \leq max \leq 100$. If the resulting number of imputations is greater than *max*, then *max* is used. By default, MAX=50.

The classic advice of using only a small number of imputations is based on considerations of relative efficiency. Recent studies, based on other aspects such as confidence intervals and *p*-values, recommend a much larger number of imputations. Thus, the default number of imputations has been increased from 5 to 25 in SAS/STAT 14.1. For more information, see the section "Number of Imputations" on page 6125.

You can specify NIMPUTE=0 to skip the imputation. In this case, only tables of model information, missing data patterns, descriptive statistics (SIMPLE option), and the MLE from the EM algorithm (EM statement) are displayed.

NOPRINT
suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*
creates an output SAS data set that contains imputation results. The data set includes an index variable, _Imputation_, to identify the imputation number. For each imputation, the data set contains all variables in the input data set with missing values being replaced by the imputed values. See the section "Output Data Sets" on page 6121 for a description of this data set.

ROUND=*numbers*
specifies the units to round variables in the imputation. If only one number is specified, that number is used for all continuous variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. When the classification variables are listed in the VAR statement, their corresponding roundoff units are not used. The default number is a missing value, which indicates no rounding for imputed variables.

When specifying a roundoff unit for the first variable only, you must also specify a missing value after the roundoff unit. Otherwise, the roundoff unit is used for all variables. For example, the option "ROUND= 10 ." sets a roundoff unit of 10 for the first analysis variable only and no rounding for the remaining variables. The option "ROUND= . 10" sets a roundoff unit of 10 for the second analysis variable only and no rounding for other variables.

The ROUND= option sets the precision of imputed values. For example, with a roundoff unit of 0.001, each value is rounded to the nearest multiple of 0.001. That is, each value has three significant digits after the decimal point. See Example 77.3 for an illustration of this option.

SEED=*number*
specifies a positive integer to start the pseudo-random number generator. The default is a value generated from reading the time of day from the computer's clock. However, in order to duplicate the results under identical situations, you must use the same value of the seed explicitly in subsequent runs of the MI procedure.

The seed information is displayed in the "Model Information" table so that the results can be reproduced by specifying this seed with the SEED= option. You need to specify the same seed number in the future to reproduce the results.

SIMPLE
displays simple descriptive univariate statistics and pairwise correlations from available cases. For a detailed description of these statistics, see the section "Descriptive Statistics" on page 6093.

SINGULAR=*p*
specifies the criterion for determining the singularity of a covariance matrix based on standardized variables, where $0 < p < 1$. The default is SINGULAR=1E–8.

Suppose that $\mathbf{S}$ is a covariance matrix and v is the number of variables in $\mathbf{S}$. Based on the spectral decomposition $\mathbf{S} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues λ_j, $j = 1, \ldots, v$, where $\lambda_i \geq \lambda_j$ when $i < j$, and $\mathbf{\Gamma}$ is a matrix with the corresponding orthonormal eigenvectors of $\mathbf{S}$ as columns, $\mathbf{S}$ is considered singular when an eigenvalue λ_j is less than $p\bar{\lambda}$, where the average $\bar{\lambda} = \sum_{k=1}^{v} \lambda_k / v$.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MI 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 MI 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 *SAS Visual Data Management and Utility Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables in the VAR statement. Classification variables can be either character or numeric. The CLASS statement must be used in conjunction with either an FCS or MONOTONE statement.

Classification levels are determined from the formatted values of the classification variables. See "The FORMAT Procedure" in the *Base SAS Procedures Guide* for details.

EM Statement

EM < *options* > **;**

The expectation-maximization (EM) algorithm is a technique for maximum likelihood estimation in parametric models for incomplete data. The EM statement uses the EM algorithm to compute the MLE for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the means and covariance matrix, of a multivariate normal distribution from the input data set with missing values. Either the means and covariances from complete cases or the means and standard deviations from available cases can be used as the initial estimates for the EM algorithm. You can also specify the correlations for the estimates from available cases.

You can also use the EM statement with the NIMPUTE=0 option in the PROC MI statement to compute the EM estimates without multiple imputation, as shown in Example 77.1.

The following seven options are available with the EM statement (in alphabetical order):

CONVERGE=*p*
XCONV=*p*
sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than *p* for each parameter—that is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E–4.

INITIAL=CC | AC | AC(R=*r***)**
sets the initial estimates for the EM algorithm. The INITIAL=CC option uses the means and covariances from complete cases; the INITIAL=AC option uses the means and standard deviations from available cases, and the correlations are set to zero; and the INITIAL=AC(R= *r*) option uses the means and standard deviations from available cases with correlation *r*, where $-1/(p-1) < r < 1$ and p is the number of variables to be analyzed. The default is INITIAL=AC.

ITPRINT
prints the iteration history in the EM algorithm.

MAXITER=*number*
specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

OUT=*SAS-data-set*
creates an output SAS data set that contains results from the EM algorithm. The data set contains all variables in the input data set, with missing values being replaced by the expected values from the EM algorithm. See the section "Output Data Sets" on page 6121 for a description of this data set.

OUTEM=*SAS-data-set*
creates an output SAS data set of TYPE=COV that contains the MLE of the parameter vector $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. These estimates are computed with the EM algorithm. See the section "Output Data Sets" on page 6121 for a description of this output data set.

OUTITER < (*options* **) > =***SAS-data-set*
creates an output SAS data set of TYPE=COV that contains parameters for each iteration. The data set includes a variable named _Iteration_ to identify the iteration number. The parameters in the output data set depend on the options specified. You can specify the MEAN and COV options to output the mean and covariance parameters. When no options are specified, the output data set contains the mean parameters for each iteration. See the section "Output Data Sets" on page 6121 for a description of this data set.

FCS Statement

FCS < *options* > **;**

The FCS statement specifies a multivariate imputation by fully conditional specification methods. If you specify an FCS statement, you must also specify a VAR statement.

Table 77.2 summarizes the options available for the FCS statement.

Table 77.2 Summary of Options in FCS

Option	Description
Imputation Details	
NBITER=	Specifies the number of burn-in iterations
Data Set	
OUTITER=	Outputs parameter estimates used in iterations
ODS Graphics Output	
PLOTS=TRACE	Displays trace plots
Imputation Methods	
DISCRIM	Specifies the discriminant function method
LOGISTIC	Specifies the logistic regression method
REG	Specifies the regression method
REGPMM	Specifies the predictive mean matching method

The following options are available for the FCS statement in addition to the imputation methods specified (in alphabetical order):

NBITER=*number*

specifies the number of burn-in iterations before each imputation. The default is NBITER=20.

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

creates an output SAS data set of TYPE=COV that contains parameters used in the imputation step for each iteration. The data set includes variables named _Imputation_ and _Iteration_ to identify the imputation number and iteration number.

The parameters in the output data set depend on the options specified. You can specify the options MEAN and STD to output parameters of means and standard deviations, respectively. When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. See the section "Output Data Sets" on page 6121 for a description of this data set.

PLOTS < (LOG) > < = TRACE < (*trace-options*) > >

requests statistical graphics of trace plots from iterations via the Output Delivery System (ODS).

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;
proc mi data=Fitness1 seed=501213 mu0=50 10 180;
   mcmc plots=(trace(mean(Oxygen)) acf(mean(Oxygen)));
   var Oxygen RunTime RunPulse;
run;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 615 in Chapter 21, "Statistical Graphics Using ODS."

The global plot option LOG requests that the logarithmic transformations of parameters be used. The default is PLOTS=TRACE(MEAN).

The available *trace-options* are as follows:

MEAN < (*variables*) >
displays plots of means for continuous variables in the list. When the MEAN option is specified without variables, all continuous variables are used.

STD < (*variables*) >
displays plots of standard deviations for continuous variables in the list. When the STD option is specified without variables, all continuous variables are used.

The discriminant function, logistic regression, regression, and predictive mean matching methods are available in the FCS statement. You specify each method with the syntax

method < (< *imputed* < = *effects* > > </ *options* >) >

That is, for each method, you can specify the imputed variables and, optionally, a set of effects to impute these variables. Each effect is a variable or a combination of variables in the VAR statement. The syntax for the specification of effects is the same as for the GLM procedure. See Chapter 48, "The GLM Procedure," for more information.

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 an FCS statement is used without specifying any methods, the regression method is used for all imputed continuous variables and the discriminant function method is used for all imputed classification variables. In this case, for each imputed continuous variable, all other variables in the VAR statement are used as the covariates, and for each imputed classification variable, all other continuous variables in the VAR statement are used as the covariates.

When a method for continuous variables is specified without imputed variables, the method is used for all continuous variables in the VAR statement that are not specified in other methods. Similarly, when a method for classification variables is specified without imputed variables, the method is used for all classification variables in the VAR statement that are not specified in other methods.

For each imputed variable that does not use the discriminant function method, if no covariates are specified, then all other variables in the VAR statement are used as the covariates. That is, each continuous variable is used as a regressor effect, and each classification variable is used as a main effect. For an imputed variable that uses the discriminant function method, if no covariates are specified, then all other variables in the VAR statement are used as the covariates with the CLASSEFFECTS=INCLUDE option, and all other continuous variables in the VAR statement are used as the covariates with the CLASSEFFECTS=EXCLUDE option (which is the default).

With an FCS statement, the variables are imputed sequentially in the order specified in the VAR statement. For a continuous variable, you can use a regression method or a regression predicted mean matching method to impute missing values. For a nominal classification variable, you can use either a discriminant function method or a logistic regression method (generalized logit model) to impute missing values without using the ordering of the class levels. For an ordinal classification variable, you can use a logistic regression method (cumulative logit model) to impute missing values by using the ordering of the class levels. For a binary classification variable, either a discriminant function method or a logistic regression method can be used. By default, a regression method is used for a continuous variable, and a discriminant function method is used for a classification variable.

Note that except for the regression method, all other methods impute values from the observed values. See the section "FCS Methods for Data Sets with Arbitrary Missing Patterns" on page 6108 for a detailed description of the FCS methods.

You can specify the following imputation methods in an FCS statement (in alphabetical order):

DISCRIM < (*imputed* < = *effects* > < / *options* >) >

specifies the discriminant function method of classification variables. The available options are as follows:

CLASSEFFECTS=EXCLUDE | INCLUDE

specifies whether the CLASS variables are used as covariate effects. The CLASSEFFECTS=EXCLUDE option excludes the CLASS variables from covariate effects and the CLASSEFFECTS=INCLUDE option includes the CLASS variables as covariate effects. The default is CLASSEFFECTS=EXCLUDE.

DETAILS

displays the group means and pooled covariance matrix used in each imputation.

PCOV=FIXED | POSTERIOR

specifies the pooled covariance used in the discriminant method. The PCOV=FIXED option uses the observed-data pooled covariance matrix for each imputation and the PCOV=POSTERIOR option draws a pooled covariance matrix from its posterior distribution. The default is PCOV=POSTERIOR.

PRIOR=EQUAL | JEFFREYS < =*c* > | PROPORTIONAL | RIDGE < =*d* >

specifies the prior probabilities of group membership. The PRIOR=EQUAL option sets the prior probabilities equal for all groups; the PRIOR=JEFFREYS < =*c* > option specifies a noninformative prior, $0 < c < 1$; the PRIOR=PROPORTIONAL option sets the prior probabilities proportion to the group sample sizes; and the PRIOR=RIDGE < =*d* > option specifies a ridge prior, $d > 0$. If the noninformative prior c is not specified, c=0.5 is used. If the ridge prior d is not specified, d=0.25 is used. The default is PRIOR=JEFFREYS.

See the section "Monotone and FCS Discriminant Function Methods" on page 6102 for a detailed description of the method.

LOGISTIC < (*imputed* < = *effects* > < / *options* >) >

specifies the logistic regression method for classification variables. The available options are as follows:

DESCENDING

reverses the sort order for the levels of the response variables.

DETAILS

displays the regression coefficients in the logistic regression model used in each imputation.

LIKELIHOOD=NOAUGMENT

LIKELIHOOD=AUGMENT < (WEIGHT=*w* | NPARM < (MULT=*m*) >) >

specifies whether to add new observations to the likelihood function in the computation of maximum likelihood estimates. The LIKELIHOOD=AUGMENT option adds observations in each response group to the likelihood function, and the LIKELIHOOD=NOAUGMENT option makes no adjustment to the likelihood function. By default, LIKELIHOOD=NOAUGMENT.

The LIKELIHOOD=AUGMENT option is useful when the maximum likelihood parameter estimates do not exist. When LIKELIHOOD=AUGMENT, each added observation contributes the same weight, and the WEIGHT= option specifies the total added weight:

WEIGHT=*w*
explicitly specifies the total added weight *w*.

WEIGHT=NPARM < (MULT=*m*) >
uses the number of parameters in the logistic regression model as the total added weight. For example, for a simple binary logistic regression model that consists only of p continuous effects, the added weight is $p+1$. The MULT=*m* option specifies the multiplier for the total added weight, $0 < m \leq 1$, and the resulting total added weight is *m* times the number of parameters in the model. By default, MULT=1.

BY default, WEIGHT=NPARM. You can specify either the MULT=*m* suboption in WEIGHT=NPARM or the WEIGHT=*w* option to use a different total added weight in the computation of maximum likelihood estimates. For example, if the ratio between the number of parameters and the number of available observations (before augmentation) is large, you can use either MULT=*m* or WEIGHT=*w* to reduce the weight for the added observations (that is, reduce the effect from the added observations in the computation of maximum likelihood estimates). For more information about the augmented data approach, see the section "Logistic Regression with Augmented Data" on page 6107.

LINK=GLOGIT | LOGIT
specifies the link function that links the response probabilities to the linear predictors. The LINK=LOGIT option (which is the default) uses the log odds function to fit the binary logit model when there are two response categories and to fit the cumulative logit model when there are more than two response categories. The LINK=GLOGIT option uses the generalized logit function to fit the generalized logit model, in which each nonreference category is contrasted with the last category.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sort order for the levels of the response variable. The ORDER=DATA sorts by the order of appearance in the input data set; the ORDER=FORMATTED sorts by their external formatted values; the ORDER=FREQ sorts by the descending frequency counts; and the ORDER=INTERNAL sorts by the unformatted values. The default is ORDER=FORMATTED.

See the section "Monotone and FCS Logistic Regression Methods" on page 6104 for a detailed description of the method.

REG | REGRESSION < (*imputed* < = *effects* > < / DETAILS >) >
specifies the regression method of continuous variables. The DETAILS option displays the regression coefficients in the regression model used in each imputation.

With a regression method, the MAXIMUM=, MINIMUM=, and ROUND= options can be used to make the imputed values more consistent with the observed variable values.

See the section "Monotone and FCS Regression Methods" on page 6100 for a detailed description of the method.

REGPMM < **(** *imputed* < **=** *effects* > < / *options* > **)** >

REGPREDMEANMATCH < **(** *imputed* < **=** *effects* > < / *options* > **)** >

specifies the predictive mean matching method for continuous variables. This method is similar to the regression method except that it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

The available options are DETAILS and K=. The DETAILS option displays the regression coefficients in the regression model used in each imputation. The K= option specifies the number of closest observations to be used in the selection. The default is K=5.

See the section "Monotone and FCS Predictive Mean Matching Methods" on page 6101 for a detailed description of the method.

With an FCS statement, the missing values of variables in the VAR statement are imputed. After the initial filled in, these variables with missing values are imputed sequentially in the order specified in the VAR statement in each iteration. For example, the following MI procedure statements use the regression method to impute variable y1 from effect y2, the regression method to impute variable y3 from effects y1 and y2, the logistic regression method to impute variable c1 from effects y1, y2, and y1 ∗ y2, and the default regression method for continuous variables to impute variable y2 from effects y1, y3, and c1:

```
proc mi;
   class c1;
   fcs reg(y1= y2);
   fcs reg(y3= y1 y2);
   fcs logistic(c1= y1 y2 y1*y2);
   var y1 y2 y3 c1;
run;
```

FREQ Statement

FREQ *variable* **;**

To run a procedure on an input data set that contains observations that occur multiple times, you can use a variable in the data set to represent how frequently observations occur and specify a FREQ statement with the name of that variable as its argument (*variable*) when you run the procedure.

When you specify a FREQ statement in other SAS procedures, they treat the data set as if each observation appeared *n* times, where *n* is the value of *variable* in the observation. However, PROC MI treats the data set differently because as PROC MI imputes each missing value in each observation, it generates only one imputed value for that missing value. That is, when you specify a FREQ *variable*, each imputed observation (with its imputed value in place of the missing value) is treated as if it appeared *n* times. In contrast, if an observation actually occurs *n* times in the data set, the missing value at each occurrence is imputed separately, and the resulting *n* observations are not identical.

PROC MI uses only the integer portion of each value of *variable*; if any value is less than 1, PROC MI does not use the corresponding observation in the analysis. When PROC MI calculates significance probabilities, it considers the total number of observations to be equal to the sum of the values of *variable*.

MCMC Statement

MCMC < *options* > **;**

The MCMC statement specifies the details of the MCMC method for imputation.

Table 77.3 summarizes the options available for the MCMC statement.

Table 77.3 Summary of Options in MCMC

Option	Description
Data Sets	
INEST=	Inputs parameter estimates for imputations
OUTEST=	Outputs parameter estimates used in imputations
OUTITER=	Outputs parameter estimates used in iterations
Imputation Details	
IMPUTE=	Specifies monotone or full imputation
CHAIN=	Specifies single or multiple chain
NBITER=	Specifies the number of burn-in iterations for each chain
NITER=	Specifies the number of iterations between imputations in a chain
INITIAL=	Specifies initial parameter estimates for MCMC
PRIOR=	Specifies the prior parameter information
START=	Specifies starting parameters
ODS Graphics Output	
PLOTS=TRACE	Displays trace plots
PLOTS=ACF	Displays autocorrelation plots
Printed Output	
WLF	Displays the worst linear function
DISPLAYINIT	Displays initial parameter values for MCMC

The following options are available for the MCMC statement (in alphabetical order).

CHAIN=SINGLE | MULTIPLE
specifies whether a single chain is used for all imputations or a separate chain is used for each imputation. The default is CHAIN=SINGLE.

DISPLAYINIT
displays initial parameter values in the MCMC method for each imputation.

IMPUTE=FULL | MONOTONE
specifies whether a full-data imputation is used for all missing values or a monotone-data imputation is used for a subset of missing values to make the imputed data sets have a monotone missing pattern. The default is IMPUTE=FULL. When IMPUTE=MONOTONE is specified, the order in the VAR statement is used to complete the monotone pattern.

INEST=*SAS-data-set*

names a SAS data set of TYPE=EST that contains parameter estimates for imputations. These estimates are used to impute values for observations in the DATA= data set. A detailed description of the data set is provided in the section "Input Data Sets" on page 6120.

INITIAL=EM <(*options***)>**

INITIAL=INPUT=*SAS-data-set*

specifies the initial mean and covariance estimates for the MCMC method. The default is INITIAL=EM.

You can specify INITIAL=INPUT=*SAS-data-set* to read the initial estimates of the mean and covariance matrix for each imputation from a SAS data set. See the section "Input Data Sets" on page 6120 for a description of this data set.

With INITIAL=EM, PROC MI derives parameter estimates for a posterior mode, the highest observed-data posterior density, from the EM algorithm. The MLE from the EM algorithm is used to start the EM algorithm for the posterior mode, and the resulting EM estimates are used to begin the MCMC method. The prior information specified in the PRIOR= option is also used in the process to compute the posterior mode.

The following four options are available with INITIAL=EM:

BOOTSTRAP <=*number* **>**

requests bootstrap resampling, which uses a simple random sample with replacement from the input data set for the initial estimate. You can explicitly specify the number of observations in the random sample. Alternatively, you can implicitly specify the number of observations in the random sample by specifying the proportion $p, 0 < p <= 1$, to request $[np]$ observations in the random sample, where n is the number of observations in the data set and $[np]$ is the integer part of np. This produces an overdispersed initial estimate that provides different starting values for the MCMC method. If you specify the BOOTSTRAP option without the number, $p = 0.75$ is used by default.

CONVERGE=*p*

XCONV=*p*

sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than *p* for each parameter—that is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E–4.

ITPRINT

prints the iteration history in the EM algorithm for the posterior mode.

MAXITER=*number*

specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

NBITER=*number*

specifies the number of burn-in iterations before the first imputation in each chain. The default is NBITER=200.

NITER=*number*
specifies the number of iterations between imputations in a single chain. The default is NITER=100.

OUTEST=*SAS-data-set*
creates an output SAS data set of TYPE=EST. The data set contains parameter estimates used in each imputation. The data set also includes a variable named _Imputation_ to identify the imputation number. See the section "Output Data Sets" on page 6121 for a description of this data set.

OUTITER < (*options*) > =*SAS-data-set*
creates an output SAS data set of TYPE=COV that contains parameters used in the imputation step for each iteration. The data set includes variables named _Imputation_ and _Iteration_ to identify the imputation number and iteration number.

The parameters in the output data set depend on the options specified. You can specify the options MEAN, STD, COV, LR, LR_POST, and WLF to output parameters of means, standard deviations, covariances, –2 log LR statistic, –2 log LR statistic of the posterior mode, and the worst linear function, respectively. When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. See the section "Output Data Sets" on page 6121 for a description of this data set.

PLOTS < (LOG) > < = *plot-request* >
PLOTS < (LOG) > < = (*plot-request* < . . . *plot-request* >) >
requests statistical graphics via the Output Delivery System (ODS). To request these graphs, ODS Graphics must be enabled and you must specify options in the MCMC statement. For more information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

The global plot option LOG requests that the logarithmic transformations of parameters be used. The plot request options include the following:

ACF < (*acf-options*) >
displays plots of the autocorrelation function of parameters from iterations. The default is ACF(MEAN).

ALL
produces all appropriate plots.

NONE
suppresses all plots.

TRACE < (*trace-options*) >
displays trace plots of parameters from iterations. The default is TRACE(MEAN).

The available *acf-options* are as follows:

NLAG=*n*
specifies the maximum lag of the series. The default is NLAG=20. The autocorrelations at each lag are displayed in the graph.

COV < (< *variables* > < *variable1*variable2* > . . .) >**
displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < (*variables*) >
displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF
displays the plot for the worst linear function.

The available *trace-options* are as follows:

COV <(< *variables* > < *variable1*variable2* > . . .)>

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < (*variables*) >
displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF
displays the plot of the worst linear function.

PRIOR=*name*

PRIOR=JEFFREYS | RIDGE=*number* **| INPUT=***SAS-data-set*
specifies the prior information for the means and covariances. The PRIOR=JEFFREYS option specifies a noninformative prior, the RIDGE=*number* option specifies a ridge prior, and the INPUT=*SAS-data-set* option specifies a data set that contains prior information.

For a detailed description of the prior information, see the section "Bayesian Estimation of the Mean Vector and Covariance Matrix" on page 6113 and the section "Posterior Step" on page 6113. If you do not specify the PRIOR= option, the default is PRIOR=JEFFREYS.

The PRIOR=INPUT= option specifies a TYPE=COV data set from which the prior information of the mean vector and the covariance matrix is read. See the section "Input Data Sets" on page 6120 for a description of this data set.

START=VALUE | DIST
specifies that the initial parameter estimates are used either as the starting value (START=VALUE) or as the starting distribution (START=DIST) in the first imputation step of each chain. If the IMPUTE=MONOTONE option is specified, then START=VALUE is used in the procedure. The default is START=VALUE.

WLF
displays the worst linear function of parameters. This scalar function of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is "worst" in the sense that its values from iterations converge most slowly among parameters. For a detailed description of this statistic, see the section "Worst Linear Function of Parameters" on page 6119.

MNAR Statement

MNAR *options* **;**

The MNAR statement imputes missing values by using the pattern-mixture model approach, assuming the missing data are missing not at random (MNAR), which is described in the section "Multiple Imputation with Pattern-Mixture Models" on page 6128. By comparing inferential results for these values to results for imputed values that are obtained under the missing at random (MAR) assumption, you can assess the sensitivity of the conclusions to the MAR assumption. The inference under MAR is questionable if it leads to results that are different from the results for a plausible MNAR scenario.

There are two main options in the MNAR statement, MODEL and ADJUST. You use the MODEL option to specify a subset of observations from which imputation models are to be derived for specified variables. You use the ADJUST option to specify an imputed variable and adjustment parameters (such as shift and scale) for adjusting the imputed variable values for a specified subset of observations.

The MNAR statement is applicable only if it is used along with a MONOTONE statement or an FCS statement. For a detailed explanation of the imputation process for the MNAR statement and how this process is implemented differently using the MONOTONE and FCS statements, see the section "Multiple Imputation with Pattern-Mixture Models" on page 6128.

MODEL(*imputed-variables* / *model-options* **)**

specifies a set of *imputed-variables* in the VAR statement and the subset of observations from which the imputation models for these variables are to be derived. You can specify multiple MODEL options in the MNAR statement, but only one MODEL option for each imputed variable.

When an imputed variable that is listed in the VAR statement is not specified as an *imputed-variable* in the MODEL option, all available observations are used to construct the imputation for that variable.

The following *model-options* provide various ways to specify the subset of observations:

MODELOBS=CCMV <(K= *k***)>**

MODELOBS=NCMV <(K= *k***)>**

MODELOBS=(*obs-variable***=***character-list***)**

identifies the subset of observations that are used to derive the imputation models.

When you use the MNAR statement along with an FCS statement, only the MODELOBS=(*obs-variable=character-list*) *model-option* is applicable. When you use the MNAR statement along with a MONOTONE statement, all three *model-options* are applicable.

MODELOBS=CCMV specifies the complete-case missing values method (Little 1993; Molenberghs and Kenward 2007, p. 35). This method derives the imputation model from the group of observations for which all the variables are observed.

MODELOBS=CCMV(K=*k*) uses the k groups of observations together with as many observed variables as possible to derive the imputation models. For a data set that has a monotone missing pattern and p variables, there are at most p groups of observations for which the same number of variables is observed. The default is K=1, which uses observations from the group for which all the variables in the VAR statement are observed (this corresponds to MODELOBS=CCMV).

MODELOBS=NCMV specifies the neighboring-case missing values method (Molenberghs and Kenward 2007, pp. 35–36). For an imputed variable Y_j, this method uses the observations for which Y_j is observed and Y_{j+1} is missing.

For an imputed variable Y_j, MODELOBS=NCMV(K=*k*) uses the k closest groups of observations for which Y_j is observed and for which Y_{j+k} is missing. The default is K=1, which corresponds to MODELOBS=NCMV.

MODELOBS=(*obs-variable*=*character-list*) identifies the subset of observations from which the imputation models are to be derived in terms of specified levels of the *obs-variable*. You must also specify the *obs-variable* in the CLASS statement. If you include the *obs-variable* in the VAR statement, it must be completely observed.

For a detailed description of the options for specifying the observations for deriving the imputation model. see the section "Specifying Sets of Observations for Imputation in Pattern-Mixture Models" on page 6130.

ADJUST(*imputed-variable* / *adjust-options* **)**

ADJUST(*imputed-variable* **(EVENT='***level***')** / *adjust-options* **)**

specifies an *imputed-variable* in the VAR statement and the subset of observations from which the imputed values for the variable are to be adjusted. If the *imputed-variable* is a classification variable, you must specify the EVENT= option to identify the response category to which the adjustments are applied. The *adjust-options* specify the subset of observations and the adjustment parameters.

You can specify multiple ADJUST options. Each ADJUST option adjusts the imputed values of an *imputed-variable* for the subset of observations that are specified in the option. The ADJUST option applies only to continuous *imputed-variables* whose values are imputed using the regression and predictive mean matching methods, and to classification *imputed-variables* whose values are imputed using the logistic regression method.

You can use the following *adjust-option* to specify the subset of observations to be adjusted:

ADJUSTOBS= (*obs-variable*=*character-list* **)**

identifies the subset of observations for which the imputed values of *imputed-variable* are to be adjusted in terms of specified levels of the *obs-variable*. You must also specify the *obs-variable* in the CLASS statement. If the *obs-variable* appears in the VAR statement, it must be completely observed.

If you do not specify the ADJUSTOBS= option, all the imputed values of *imputed-variable* are adjusted.

You can use the following *adjust-options* to explicitly specify adjustment parameters:

SCALE=*c*

specifies a scale parameter for adjusting imputed values of a continuous *imputed-variable*. The value of *c* must be positive. By default, *c*= 1 (no scale adjustment is made). The SCALE= option does not apply to adjusting imputed values of classification variables.

SHIFT | DELTA=δ

specifies the shift parameter for imputed values of *imputed-variable*. By default, $\delta = 0$ (no shift adjustment is made).

SIGMA=σ

specifies the sigma parameter for imputed values of *imputed-variable*, where $\sigma \geq 0$. For a specified $\sigma > 0$, a simulated shift parameter is generated from the normal distribution, with mean δ and standard deviation σ in each imputation. By default, $\sigma = 0$, which means that the same shift adjustment δ is made for imputed values of *imputed-variable*.

You can use the following *adjust-option* to adjust imputed values by using parameters that are stored in a data set:

PARMS(*parms-options* **)=***SAS-data-set*
names the SAS data set that contains the adjustment parameters at each imputation for imputed values of *imputed-variable*. You can specify the following *parms-options*:

SHIFT | DELTA=*variable*
identifies the *variable* for the shift parameter.

SCALE=*variable*
identifies the *variable* for the scale parameter of a continuous *imputed-variable*.

When the PARMS= data set does not contain a variable named _IMPUTATION_, the same adjustment parameters are used for each imputation. When the PARMS= data set contains a variable named _IMPUTATION_, whose values are 1, 2, ..., n, where n is the number of imputations, the adjustment parameters are used for the corresponding imputations.

For a classification *imputed-variable* whose values are imputed by using an ordinal logistic regression method, you cannot specify the SHIFT= and SIGMA= parameters for more than one EVENT= level if the imputed variable has more than two response levels. For a detailed description of imputed value adjustments, see the section "Adjusting Imputed Values in Pattern-Mixture Models" on page 6131.

MONOTONE Statement

MONOTONE < *method* < **(**< *imputed* < **=** *effects* > > < / *options* >**)** > >
< . . . *method* < **(**< *imputed* < **=** *effects* > > < / *options* >**)** > > **;**

The MONOTONE statement specifies imputation methods for data sets with monotone missingness. You must also specify a VAR statement, and the data set must have a monotone missing pattern with variables ordered in the VAR list.

Table 77.4 summarizes the options available for the MONOTONE statement.

Table 77.4 Summary of Imputation Methods in MONOTONE Statement

Option	Description
DISCRIM	Specifies the discriminant function method
LOGISTIC	Specifies the logistic regression method
PROPENSITY	Specifies the propensity scores method
REG	Specifies the regression method
REGPMM	Specifies the predictive mean matching method

For each method, you can specify the imputed variables and, optionally, a set of the effects to impute these variables. Each effect is a variable or a combination of variables preceding the imputed variable in the VAR statement. The syntax for specification of effects is the same as for the GLM procedure. See Chapter 48, "The GLM Procedure," for more information.

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 a MONOTONE statement is used without specifying any methods, the regression method is used for all imputed continuous variables and the discriminant function method is used for all imputed classification variables. In this case, for each imputed continuous variable, all preceding variables in the VAR statement are used as the covariates, and for each imputed classification variable, all preceding continuous variables in the VAR statement are used as the covariates.

When a method for continuous variables is specified without imputed variables, the method is used for all continuous variables in the VAR statement that are not specified in other methods. Similarly, when a method for classification variables is specified without imputed variables, the method is used for all classification variables in the VAR statement that are not specified in other methods.

For each imputed variable that does not use the discriminant function method, if no covariates are specified, then all preceding variables in the VAR statement are used as the covariates. That is, each preceding continuous variable is used as a regressor effect, and each preceding classification variable is used as a main effect. For an imputed variable that uses the discriminant function method, if no covariates are specified, then all preceding variables in the VAR statement are used as the covariates with the CLASSEFFECTS=INCLUDE option, and all preceding continuous variables in the VAR statement are used as the covariates with the CLASSEFFECTS=EXCLUDE option (which is the default).

With a MONOTONE statement, the variables are imputed sequentially in the order given by the VAR statement. For a continuous variable, you can use a regression method, a regression predicted mean matching method, or a propensity score method to impute missing values. For a nominal classification variable, you can use either a discriminant function method or a logistic regression method (generalized logit model) to impute missing values without using the ordering of the class levels. For an ordinal classification variable, you can use a logistic regression method (cumulative logit model) to impute missing values by using the ordering of the class levels. For a binary classification variable, either a discriminant function method or a logistic regression method can be used.

Note that except for the regression method, all other methods impute values from the observed observation values. You can specify the following methods in a MONOTONE statement.

DISCRIM < (*imputed* < = *effects* > < / *options* >) >

specifies the discriminant function method of classification variables. The available options are as follows:

CLASSEFFECTS=EXCLUDE | INCLUDE

specifies whether the CLASS variables are used as covariate effects. The CLASSEFFECTS=EXCLUDE option excludes the CLASS variables from covariate effects and the CLASSEFFECTS=INCLUDE option includes the CLASS variables as covariate effects. The default is CLASSEFFECTS=EXCLUDE.

DETAILS

displays the group means and pooled covariance matrix used in each imputation.

PCOV=FIXED | POSTERIOR
specifies the pooled covariance used in the discriminant method. The PCOV=FIXED option uses the observed-data pooled covariance matrix for each imputation and the PCOV=POSTERIOR option draws a pooled covariance matrix from its posterior distribution. The default is PCOV=POSTERIOR.

PRIOR=EQUAL | JEFFREYS <=*c*> | PROPORTIONAL | RIDGE <=*d*>
specifies the prior probabilities of group membership. The PRIOR=EQUAL option sets the prior probabilities equal for all groups; the PRIOR=JEFFREYS <=*c*> option specifies a noninformative prior, $0 < c < 1$; the PRIOR=PROPORTIONAL option sets the prior probabilities proportion to the group sample sizes; and the PRIOR=RIDGE <=*d*> option specifies a ridge prior, $d > 0$. If the noninformative prior c is not specified, c=0.5 is used. If the ridge prior d is not specified, d=0.25 is used. The default is PRIOR=JEFFREYS.

See the section "Monotone and FCS Discriminant Function Methods" on page 6102 for a detailed description of the method.

LOGISTIC <(*imputed* **<** = *effects*> **<**/ *options*> **)** >
specifies the logistic regression method for classification variables. The available options are as follows:

DESCENDING
reverses the sort order for the levels of the response variables.

DETAILS
displays the regression coefficients in the logistic regression model used in each imputation.

LIKELIHOOD=NOAUGMENT

LIKELIHOOD=AUGMENT <(WEIGHT=*w* | NPARM <(MULT=*m*)>)>
specifies whether to add new observations to the likelihood function in the computation of maximum likelihood estimates. The LIKELIHOOD=AUGMENT option adds observations in each response group to the likelihood function, and the LIKELIHOOD=NOAUGMENT option makes no adjustment to the likelihood function. By default, LIKELIHOOD=NOAUGMENT.

The LIKELIHOOD=AUGMENT option is useful when the maximum likelihood parameter estimates do not exist. When LIKELIHOOD=AUGMENT, each added observation contributes the same weight, and the WEIGHT= option specifies the total added weight:

WEIGHT=*w*
explicitly specifies the total added weight w.

WEIGHT=NPARM <(MULT=*m*)>
uses the number of parameters in the logistic regression model as the total added weight. For example, for a simple binary logistic regression model that consists only of p continuous effects, the added weight is p+1. The MULT=m option specifies the multiplier for the total added weight, $0 < m \leq 1$, and the resulting total added weight is m times the number of parameters in the model. By default, MULT=1.

By default, WEIGHT=NPARM. You can specify either the MULT=m suboption in WEIGHT=NPARM or the WEIGHT=w option to use a different total added weight in the computation of maximum likelihood estimates. For example, if the ratio between the number of parameters and the number of available observations (before augmentation) is large, you

can use either MULT=*m* or WEIGHT=*w* to reduce the weight for the added observations (that is, reduce the effect from the added observations in the computation of maximum likelihood estimates). For more information about the augmented data approach, see the section "Logistic Regression with Augmented Data" on page 6107.

LINK=GLOGIT | LOGIT
specifies the link function linking the response probabilities to the linear predictors. The default is LINK=LOGIT. The LINK=LOGIT option uses the log odds function to fit the binary logit model when there are two response categories and to fit the cumulative logit model when there are more than two response categories; and the LINK=GLOGIT option uses the generalized logit function to fit the generalized logit model where each nonreference category is contrasted with the last category.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sort order for the levels of the response variable. The ORDER=DATA sorts by the order of appearance in the input data set; the ORDER=FORMATTED sorts by their external formatted values; the ORDER=FREQ sorts by the descending frequency counts; and the ORDER=INTERNAL sorts by the unformatted values. The default is ORDER=FORMATTED.

See the section "Monotone and FCS Logistic Regression Methods" on page 6104 for a detailed description of the method.

PROPENSITY < (*imputed* < = *effects* > < / *options* >) >
specifies the propensity scores method of variables. Each variable is either a classification variable or a continuous variable. The available options are DETAILS and NGROUPS=. The DETAILS option displays the regression coefficients in the logistic regression model for propensity scores. The NGROUPS= option specifies the number of groups created based on propensity scores. The default is NGROUPS=5.

See the section "Monotone Propensity Score Method" on page 6108 for a detailed description of the method.

REG | REGRESSION < (*imputed* < = *effects* > < / DETAILS >) >
specifies the regression method of continuous variables. The DETAILS option displays the regression coefficients in the regression model used in each imputation.

With a regression method, the MAXIMUM=, MINIMUM=, and ROUND= options can be used to make the imputed values more consistent with the observed variable values.

See the section "Monotone and FCS Regression Methods" on page 6100 for a detailed description of the method.

REGPMM < (*imputed* < = *effects* > < / *options* >) >
REGPREDMEANMATCH < (*imputed* < = *effects* > < / *options* >) >
specifies the predictive mean matching method for continuous variables. This method is similar to the regression method except that it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

The available options are DETAILS and K=. The DETAILS option displays the regression coefficients in the regression model used in each imputation. The K= option specifies the number of closest observations to be used in the selection. The default is K=5.

See the section "Monotone and FCS Predictive Mean Matching Methods" on page 6101 for a detailed description of the method.

With a MONOTONE statement, the variables with missing values are imputed sequentially in the order specified in the VAR statement. For example, the following MI procedure statements use the default regression method for continuous variables to impute variable y2 from the effect y1, the logistic regression method to impute variable c1 from effects y1, y2, and y1 * y2, and the regression method to impute variable y3 from effects y1, y2, and c1:

```
proc mi;
   class c1;
   var y1 y2 c1 y3;
   monotone logistic(c1= y1 y2 y1*y2);
   monotone reg(y3= y1 y2 c1);
run;
```

The variable y1 is not imputed since it is the leading variable in the VAR statement.

TRANSFORM Statement

TRANSFORM *transform* **(***variables*< */ options* >**)**< *. . . transform* **(***variables*< */ options* >**)** > **;**

The TRANSFORM statement lists the transformations and their associated variables to be transformed. The options are transformation options that provide additional information for the transformation.

The MI procedure assumes that the data are from a multivariate normal distribution when either the regression method or the MCMC method is used. When some variables in a data set are clearly non-normal, it is useful to transform these variables to conform to the multivariate normality assumption. With a TRANSFORM statement, variables are transformed before the imputation process, and these transformed variable values are displayed in all of the results. When you specify an OUT= option, the variable values are back-transformed to create the imputed data set.

The following transformations can be used in the TRANSFORM statement:

BOXCOX
specifies the Box-Cox transformation of variables. The variable Y is transformed to $\frac{(Y+c)^{\lambda}-1}{\lambda}$, where c is a constant such that each value of $Y + c$ must be positive. If the specified constant $\lambda = 0$, the logarithmic transformation is used.

EXP
specifies the exponential transformation of variables. The variable Y is transformed to $e^{(Y+c)}$, where c is a constant.

LOG
specifies the logarithmic transformation of variables. The variable Y is transformed to $\log(Y + c)$, where c is a constant such that each value of $Y + c$ must be positive.

LOGIT
specifies the logit transformation of variables. The variable Y is transformed to $\log(\frac{Y/c}{1-Y/c})$, where the constant c>0 and the values of Y/c must be between 0 and 1.

POWER

specifies the power transformation of variables. The variable Y is transformed to $(Y + c)^{\lambda}$, where c is a constant such that each value of $Y + c$ must be positive and the constant $\lambda \neq 0$.

The following options provide the constant c and λ values in the transformations.

C=*number*

specifies the c value in the transformation. The default is $c = 1$ for logit transformation and $c = 0$ for other transformations.

LAMBDA=*number*

specifies the λ value in the power and Box-Cox transformations. You must specify the λ value for these two transformations.

For example, the following statement requests that variables $\log(\mathtt{y1})$, a logarithmic transformation for the variable y1, and $\sqrt{\mathtt{y2}+1}$, a power transformation for the variable y2, be used in the imputation:

```
transform log(y1) power(y2/c=1 lambda=.5);
```

If the MU0= option is used to specify a parameter value μ_0 for a transformed variable, the same transformation for the variable is also applied to its corresponding MU0= value in the *t* test. Otherwise, $\mu_0 = 0$ is used for the transformed variable. See Example 77.10 for a usage of the TRANSFORM statement.

VAR Statement

VAR *variables* **;**

The VAR statement lists the variables to be analyzed. The variables can be either character or numeric. If you omit the VAR statement, all continuous variables not mentioned in other statements are used. The VAR statement is required if you specify either an FCS statement, a MONOTONE statement, an IMPUTE=MONOTONE option in the MCMC statement, or more than one number in the MU0=, MAXIMUM=, MINIMUM=, or ROUND= option.

The classification variables in the VAR statement, which can be either character or numeric, are further specified in the CLASS statement.

Details: MI Procedure

Descriptive Statistics

Suppose $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)'$ is the $(n \times p)$ matrix of complete data, which might not be fully observed, n_0 is the number of observations fully observed, and n_j is the number of observations with observed values for variable Y_j.

With complete cases, the sample mean vector is

$$\bar{\mathbf{y}} = \frac{1}{n_0} \sum \mathbf{y}_i$$

and the CSSCP matrix is

$$\sum (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

where each summation is over the fully observed observations.

The sample covariance matrix is

$$\mathbf{S} = \frac{1}{n_0 - 1} \sum (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

and is an unbiased estimate of the covariance matrix.

The correlation matrix $\mathbf{R}$, which contains the Pearson product-moment correlations of the variables, is derived by scaling the corresponding covariance matrix:

$$\mathbf{R} = \mathbf{D}^{-1}\mathbf{S}\,\mathbf{D}^{-1}$$

where $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the square roots of the diagonal elements of $\mathbf{S}$.

With available cases, the corrected sum of squares for variable Y_j is

$$\sum (y_{ji} - \bar{y}_j)^2$$

where $\bar{y}_j = \frac{1}{n_j} \sum y_{ji}$ is the sample mean and each summation is over observations with observed values for variable Y_j.

The variance is

$$s_{jj}^2 = \frac{1}{n_j - 1} \sum (y_{ji} - \bar{y}_j)^2$$

The correlations for available cases contain pairwise correlations for each pair of variables. Each correlation is computed from all observations that have nonmissing values for the corresponding pair of variables.

EM Algorithm for Data with Missing Values

The EM algorithm (Dempster, Laird, and Rubin 1977) is a technique that finds maximum likelihood estimates in parametric models for incomplete data. For a detailed description and applications of the EM algorithm, see the books by Little and Rubin (2002); Schafer (1997); McLachlan and Krishnan (1997).

The EM algorithm is an iterative procedure that finds the MLE of the parameter vector by repeating the following steps:

1. The expectation E-step
Given a set of parameter estimates, such as a mean vector and covariance matrix for a multivariate normal distribution, the E-step calculates the conditional expectation of the complete-data log likelihood given the observed data and the parameter estimates.

2. The maximization M-step
Given a complete-data log likelihood, the M-step finds the parameter estimates to maximize the complete-data log likelihood from the E-step.

The two steps are iterated until the iterations converge.

In the EM process, the observed-data log likelihood is nondecreasing at each iteration. For multivariate normal data, suppose there are G groups with distinct missing patterns. Then the observed-data log likelihood being maximized can be expressed as

$$\log L(\theta|Y_{obs}) = \sum_{g=1}^{G} \log L_g(\boldsymbol{\theta}|Y_{obs})$$

where $\log L_g(\boldsymbol{\theta}|Y_{obs})$ is the observed-data log likelihood from the gth group, and

$$\log L_g(\boldsymbol{\theta}|Y_{obs}) = -\frac{n_g}{2}\log|\Sigma_g| - \frac{1}{2}\sum_{ig}(\mathbf{y}_{ig}-\boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g{}^{-1}(\mathbf{y}_{ig}-\boldsymbol{\mu}_g)$$

where n_g is the number of observations in the gth group, the summation is over observations in the gth group, $\mathbf{y}_{ig}$ is a vector of observed values corresponding to observed variables, $\boldsymbol{\mu}_g$ is the corresponding mean vector, and $\boldsymbol{\Sigma}_g$ is the associated covariance matrix.

A sample covariance matrix is computed at each step of the EM algorithm. If the covariance matrix is singular, the linearly dependent variables for the observed data are excluded from the likelihood function. That is, for each observation with linear dependency among its observed variables, the dependent variables are excluded from the likelihood function. Note that this can result in an unexpected change in the likelihood between iterations prior to the final convergence.

See Schafer (1997, pp. 163–181) for a detailed description of the EM algorithm for multivariate normal data.

By default, PROC MI uses the means and standard deviations from available cases as the initial estimates for the EM algorithm. The correlations are set to zero. These estimates provide a good starting value with positive definite covariance matrix. For a discussion of suggested starting values for the algorithm, see Schafer (1997, p. 169).

You can specify the convergence criterion with the CONVERGE= option in the EM statement. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. You can also specify the maximum number of iterations used in the EM algorithm with the MAXITER= option.

The MI procedure displays tables of the initial parameter estimates used to begin the EM process and the MLE parameter estimates derived from EM. You can also display the EM iteration history with the ITPRINT option. PROC MI lists the iteration number, the likelihood –2 log L, and the parameter values $\boldsymbol{\mu}$ at each iteration. You can also save the MLE derived from the EM algorithm in a SAS data set by specifying the OUTEM= option.

Statistical Assumptions for Multiple Imputation

The MI procedure assumes that the data are from a multivariate distribution and contain missing values that can occur for any of the variables. It also assumes that the data are from a multivariate normal distribution when either the regression method or the MCMC method is used.

Suppose $\mathbf{Y}$ is the $n \times p$ matrix of complete data, which is not fully observed, and denote the observed part of $\mathbf{Y}$ by $\mathbf{Y}_{obs}$ and the missing part by $\mathbf{Y}_{mis}$. The MI and MIANALYZE procedures assume that the missing data are missing at random (MAR); that is, the probability that an observation is missing can depend on $\mathbf{Y}_{obs}$, but not on $\mathbf{Y}_{mis}$ (Rubin 1976, 1987, p. 53).

To be more precise, suppose that $\mathbf{R}$ is the $n \times p$ matrix of response indicators whose elements are zero or one depending on whether the corresponding elements of Y are missing or observed. Then the MAR assumption is that the distribution of $\mathbf{R}$ can depend on Y_{obs} but not on Y_{mis}:

$$\mathrm{pr}(\mathbf{R}|Y_{obs}, Y_{mis}) = \mathrm{pr}(\mathbf{R}|Y_{obs})$$

For example, consider a trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 that has missing values. MAR assumes that the probability that Y_3 is missing for an individual can be related to the individual's values of variables Y_1 and Y_2, but not to its value of Y_3. On the other hand, if a complete case and an incomplete case for Y_3 with exactly the same values for variables Y_1 and Y_2 have systematically different values, then there exists a response bias for Y_3, and MAR is violated.

The MAR assumption is not the same as missing completely at random (MCAR), which is a special case of MAR. Under the MCAR assumption, the missing data values are a simple random sample of all data values; the missingness does not depend on the values of any variables in the data set.

Although the MAR assumption cannot be verified with the data and it can be questionable in some situations, the assumption becomes more plausible as more variables are included in the imputation model (Schafer 1997, pp. 27–28; Van Buuren, Boshuizen, and Knook 1999, p. 687).

Furthermore, the MI and MIANALYZE procedures assume that the parameters $\boldsymbol{\theta}$ of the data model and the parameters $\boldsymbol{\phi}$ of the model for the missing-data indicators are distinct. That is, knowing the values of $\boldsymbol{\theta}$ does not provide any additional information about $\boldsymbol{\phi}$, and vice versa. If both the MAR and distinctness assumptions are satisfied, the missing-data mechanism is said to be ignorable Rubin 1987, pp. 50–54; Schafer 1997, pp. 10–11).

Missing Data Patterns

The MI procedure sorts the data into groups based on whether the analysis variables are observed or missing. Note that the input data set does not need to be sorted in any order.

For example, with variables Y_1, Y_2, and Y_3 (in that order) in a data set, up to eight groups of observations can be formed from the data set. Figure 77.6 displays the eight groups of observations and an unique missing pattern for each group.

Figure 77.6 Missing Data Patterns

Missing Data Patterns

Group	Y1	Y2	Y3
1	X	X	X
2	X	X	.
3	X	.	X
4	X	.	.
5	.	X	X
6	.	X	.
7	.	.	X
8	.	.	.

An "X" in Figure 77.6 means that the variable is observed in the corresponding group and a "." means that the variable is missing. The MI procedure denotes variables that have missing values by "." or "O". The value "." means that the variable is missing and will be imputed, and the value "O" means that the variable is missing and will not be imputed.

The variable order is used to derive the order of the groups from the data set, and thus determines the order of missing values in the data to be imputed. If you specify a different order of variables in the VAR statement, then the results are different even if the other specifications remain the same.

A data set with variables $Y_1, Y_2, \ldots, Y_p$ (in that order) is said to have a *monotone missing pattern* when the event that a variable Y_j is missing for a particular individual implies that all subsequent variables $Y_k, k > j$, are missing for that individual. Alternatively, when a variable Y_j is observed for a particular individual, it is assumed that all previous variables $Y_k, k < j$, are also observed for that individual.

For example, Figure 77.7 displays a data set of three variables with a monotone missing pattern.

Figure 77.7 Monotone Missing Patterns

Monotone Missing Data Patterns

Group	Y1	Y2	Y3
1	X	X	X
2	X	X	.
3	X	.	.

Figure 77.8 displays a data set of three variables with a non-monotone missing pattern.

Figure 77.8 Non-monotone Missing Patterns

Non-monotone Missing Data Patterns

Group	Y1	Y2	Y3
1	X	X	X
2	X	.	X
3	.	X	.
4	.	.	X

A data set with an *arbitrary missing pattern* is a data set with either a monotone missing pattern or a non-monotone missing pattern.

Imputation Methods

This section describes the methods for multiple imputation that are available in the MI procedure. The method of choice depends on the pattern of missingness in the data and the type of the imputed variable, as summarized in Table 77.5.

Table 77.5 Imputation Methods in PROC MI

Pattern of Missingness	Type of Imputed Variable	Type of Covariates	Available Methods
Monotone	Continuous	Arbitrary	• Monotone regression • Monotone predicted mean matching • Monotone propensity score
Monotone	Classification (ordinal)	Arbitrary	• Monotone logistic regression
Monotone	Classification (nominal)	Arbitrary	• Monotone discriminant function • Monotone logistic regression
Arbitrary	Continuous	Continuous	• MCMC full-data imputation • MCMC monotone-data imputation
Arbitrary	Continuous	Arbitrary	• FCS regression • FCS predicted mean matching
Arbitrary	Classification (ordinal)	Arbitrary	• FCS logistic regression
Arbitrary	Classification (nominal)	Arbitrary	• FCS discriminant function • FCS logistic regression

To impute missing values for a continuous variable in data sets with monotone missing patterns, you should use either a parametric method that assumes multivariate normality or a nonparametric method that uses propensity scores Rubin 1987, pp. 124, 158; Lavori, Dawson, and Shera 1995). Parametric methods available include the regression method (Rubin 1987, pp. 166–167) and the predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996).

To impute missing values for a classification variable in data sets with monotone missing patterns, you should use the logistic regression method or the discriminant function method. Use the logistic regression method when the classification variable has a binary, nominal, or ordinal response, and use the discriminant function method when the classification variable has a binary or nominal response.

For data sets with arbitrary missing patterns, you can use either of the following methods to impute missing values: a Markov chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality, or a fully conditional specification (FCS) method (Van Buuren 2007; Brand 1999) that assumes the existence of a joint distribution for all variables.

For continuous variables in data sets with arbitrary missing patterns, you can use the MCMC method to impute either all the missing values or just enough missing values to make the imputed data sets have monotone missing patterns. With a monotone missing data pattern, you have greater flexibility in your choice of imputation models. In addition to the MCMC method, you can implement other methods, such as the regression method, that do not use Markov chains. You can also specify a different set of covariates for each imputed variable.

Although the regression and MCMC methods assume multivariate normality, inferences based on multiple imputation can be robust to departures from multivariate normality if the amount of missing information is not large, because the imputation model is effectively applied not to the entire data set but only to its missing part (Schafer 1997, pp.147–148).

To impute missing values for both continuous and classification variables in data sets with arbitrary missing patterns, you can use FCS methods to impute missing values for all variables assuming a joint distribution for these variables exists (Brand 1999; Van Buuren 2007). Similar to the methods of imputing missing values for variables in data sets with monotone missing patterns, you can use the regression and predictive mean matching methods to impute missing values for a continuous variable, the logistic regression method to impute missing values for a classification variable when the variable has a binary, nominal, or ordinal response, and the discriminant function method to impute missing values for a classification variable when the variable has a binary or nominal response.

You can also use a TRANSFORM statement to transform variables to conform to the multivariate normality assumption. Variables are transformed before the imputation process and then are reverse-transformed to create the imputed data set. All continuous variables are standardized before the imputation process and then are transformed back to the original scale after the imputation process.

Li (1988) presents a theoretical argument for convergence of the MCMC method in the continuous case and uses it to create imputations for incomplete multivariate continuous data. In practice, however, it is not easy to check the convergence of a Markov chain, especially for a large number of parameters. PROC MI generates statistics and plots that you can use to check for convergence of the MCMC method. The details are described in the section "Checking Convergence in MCMC" on page 6118.

Monotone Methods for Data Sets with Monotone Missing Patterns

For data sets with monotone missing data patterns, you can use monotone methods to impute missing values for the variables. A monotone method creates multiple imputations by imputing missing values sequentially over the variables taken one at a time.

For example, with variables Y_1, Y_2, ..., Y_p (in that order) in the VAR statement, a monotone method sequentially simulates a draw for missing values for variables Y_2, ..., Y_p. That is, the missing values are imputed by using the sequence

$$
\begin{aligned}
\boldsymbol{\theta}_2^{(*)} &\sim P(\boldsymbol{\theta}_2 \mid Y_{1(obs)}, Y_{2(obs)}) \\
Y_2^{(*)} &\sim P(Y_2 \mid \boldsymbol{\theta}_2^{(*)}) \\
&\cdots \\
&\cdots \\
\boldsymbol{\theta}_p^{(*)} &\sim P(\boldsymbol{\theta}_p \mid Y_{1(obs)}, \ldots, Y_{p(obs)}) \\
Y_p^{(*)} &\sim P(Y_p \mid \boldsymbol{\theta}_p^{(*)})
\end{aligned}
$$

where $Y_{j(obs)}$ is the set of observed Y_j values, $\boldsymbol{\theta}_j^{(*)}$ is the set of simulated parameters for the conditional distribution of Y_j given covariates constructed from variables Y_1, Y_2, …, Y_{j-1}, and $Y_j^{(*)}$ is the set of imputed Y_j values.

The missing values for the leading variable Y_1 are not imputed, and missing values for Y_2, …, Y_p are not imputed for those observations with missing Y_1 values. For each subsequent variable Y_j with missing values, the corresponding imputation method is used to fit a model with covariates constructed from its preceding variables $Y_1, Y_2, \ldots, Y_{j-1}$. The observed observations for Y_j, which include only observations with observed values for $Y_1, Y_2, \ldots, Y_{j-1}$, are used in the model fitting. With this resulting model, a new model is drawn and then used to impute missing values for Y_j.

You can specify a separate monotone method for each imputed variable. If a method is not specified for the variable, then the default method is used. That is, a regression method is used for a continuous variable and a discriminant function method is used for a classification variable. For each imputed variable, you can also specify a set of covariates that are constructed from its preceding variables. If a set of covariates is not specified for the variable, all preceding variables in the VAR list are used as covariates.

You can use a regression method, a predictive mean matching method, or a propensity score method to impute missing values for a continuous variable; a logistic regression method for a classification variable with a binary or ordinal response; and a discriminant function method for a classification variable with a binary or nominal response. See the sections "Monotone and FCS Regression Methods" on page 6100, "Monotone and FCS Predictive Mean Matching Methods" on page 6101, "Monotone Propensity Score Method" on page 6108, "Monotone and FCS Discriminant Function Methods" on page 6102, and "Monotone and FCS Logistic Regression Methods" on page 6104 for these methods.

Monotone and FCS Regression Methods

The regression method is the default imputation method in the MONOTONE and FCS statements for continuous variables.

In the regression method, a regression model is fitted for a continuous variable with the covariates constructed from a set of effects. Based on the fitted regression model, a new regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable (Rubin 1987, pp. 166–167). That is, for a continuous variable Y_j with missing values, a model

$$Y_j = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

is fitted using observations with observed values for the variable Y_j and its covariates X_1, X_2, …, X_k.

The fitted model includes the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$ and the associated covariance matrix $\hat{\sigma}_j^2 \mathbf{V}_j$, where $\mathbf{V}_j$ is the usual $\mathbf{X}'\mathbf{X}$ inverse matrix derived from the intercept and covariates $X_1, X_2, \ldots, X_k$.

The following steps are used to generate imputed values for each imputation:

1. New parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(k)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

   $$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - k - 1)/g$$

   where g is a $\chi^2_{n_j - k - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

   $$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

   where $\mathbf{V}_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj} \mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k + 1$ independent random normal variates.

2. The missing values are then replaced by

   $$\beta_{*0} + \beta_{*1} x_1 + \beta_{*2} x_2 + \ldots + \beta_{*(k)} x_k + z_i \sigma_{*j}$$

   where $x_1, x_2, \ldots, x_k$ are the values of the covariates and z_i is a simulated normal deviate.

Monotone and FCS Predictive Mean Matching Methods

The predictive mean matching method is also an imputation method available for continuous variables. It is similar to the regression method except that for each missing value, it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

Following the description of the model in the section "Monotone and FCS Regression Methods" on page 6100, the following steps are used to generate imputed values:

1. New parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(k)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

   $$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - k - 1)/g$$

   where g is a $\chi^2_{n_j - k - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

   $$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

   where $\mathbf{V}_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj} \mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k + 1$ independent random normal variates.

2. For each missing value, a predicted value

$$y_{i*} = \beta_{*0} + \beta_{*1}\, x_1 + \beta_{*2}\, x_2 + \ldots + \beta_{*(k)}\, x_k$$

is computed with the covariate values $x_1, x_2, \ldots, x_k$.

3. A set of k_0 observations whose corresponding predicted values are closest to y_{i*} is generated. You can specify k_0 with the K= option.

4. The missing value is then replaced by a value drawn randomly from these k_0 observed values.

The predictive mean matching method requires the number of closest observations to be specified. A smaller k_0 tends to increase the correlation among the multiple imputations for the missing observation and results in a higher variability of point estimators in repeated sampling. On the other hand, a larger k_0 tends to lessen the effect from the imputation model and results in biased estimators (Schenker and Taylor 1996, p. 430).

The predictive mean matching method ensures that imputed values are plausible; it might be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Monotone and FCS Discriminant Function Methods

The discriminant function method is the default imputation method in the MONOTONE and FCS statements for classification variables.

For a nominal classification variable Y_j with responses $1, \ldots, g$ and a set of effects from its preceding variables, if the covariates $X_1, X_2, \ldots, X_k$ associated with these effects within each group are approximately multivariate normal and the within-group covariance matrices are approximately equal, the discriminant function method (Brand 1999, pp. 95–96) can be used to impute missing values for the variable Y_j.

Denote the group-specific means for covariates $X_1, X_2, \ldots, X_k$ by

$$\overline{\mathbf{X}}_t = (\overline{X}_{t1}, \overline{X}_{t2}, \ldots, \overline{X}_{tk}),\ t = 1, 2, \ldots, g$$

then the pooled covariance matrix is computed as

$$\mathbf{S} = \frac{1}{n-g} \sum_{t=1}^{g} (n_t - 1)\mathbf{S}_t$$

where $\mathbf{S}_t$ is the within-group covariance matrix, n_t is the group-specific sample size, and $n = \sum_{t=1}^{g} n_t$ is the total sample size.

In each imputation, new parameters of the group-specific means ($\mathbf{m}_{*t}$), pooled covariance matrix ($\mathbf{S}_*$), and prior probabilities of group membership (q_{*t}) can be drawn from their corresponding posterior distributions (Schafer 1997, p. 356).

Pooled Covariance Matrix and Group-Specific Means

For each imputation, the MI procedure uses either the fixed observed pooled covariance matrix (PCOV=FIXED) or a drawn pooled covariance matrix (PCOV=POSTERIOR) from its posterior distribution with a noninformative prior. That is,

$$\mathbf{\Sigma}\,|\mathbf{X} \quad \sim \quad W^{-1}\,(n-g,\ (n-g)\mathbf{S})$$

where W^{-1} is an inverted Wishart distribution.

The group-specific means are then drawn from their posterior distributions with a noninformative prior

$$\boldsymbol{\mu}_t|(\boldsymbol{\Sigma}, \overline{\mathbf{X}}_t) \quad \sim \quad N\left(\overline{\mathbf{X}}_t, \frac{1}{n_t}\boldsymbol{\Sigma}\right)$$

See the section "Bayesian Estimation of the Mean Vector and Covariance Matrix" on page 6113 for a complete description of the inverted Wishart distribution and posterior distributions that use a noninformative prior.

Prior Probabilities of Group Membership

The prior probabilities are computed through the drawing of new group sample sizes. When the total sample size n is considered fixed, the group sample sizes $(n_1, n_2, \ldots, n_g)$ have a multinomial distribution. New multinomial parameters (group sample sizes) can be drawn from their posterior distribution by using a Dirichlet prior with parameters $(\alpha_1, \alpha_2, \ldots, \alpha_g)$.

After the new sample sizes are drawn from the posterior distribution of $(n_1, n_2, \ldots, n_g)$, the prior probabilities q_{*t} are computed proportionally to the drawn sample sizes.

See Schafer (1997, pp. 247–255) for a complete description of the Dirichlet prior.

Imputation Steps

The discriminant function method uses the following steps in each imputation to impute values for a nominal classification variable Y_j with g responses:

1. Draw a pooled covariance matrix $\mathbf{S}_*$ from its posterior distribution if the PCOV=POSTERIOR option is used.
2. For each group, draw group means $\mathbf{m}_{*t}$ from the observed group mean $\overline{\mathbf{X}}_t$ and either the observed pooled covariance matrix (PCOV=FIXED) or the drawn pooled covariance matrix $\mathbf{S}_*$ (PCOV=POSTERIOR).
3. For each group, compute or draw q_{*t}, prior probabilities of group membership, based on the PRIOR= option:
   - PRIOR=EQUAL, $q_{*t} = 1/g$, prior probabilities of group membership are all equal.
   - PRIOR=PROPORTIONAL, $q_{*t} = n_t/n$, prior probabilities are proportional to their group sample sizes.
   - PRIOR=JEFFREYS=*c*, a noninformative Dirichlet prior with $\alpha_t = c$ is used.
   - PRIOR=RIDGE=*d*, a ridge prior is used with $\alpha_t = d * n_t/n$ for $d \geq 1$ and $\alpha_t = d * n_t$ for $d < 1$.
4. With the group means $\mathbf{m}_{*t}$, the pooled covariance matrix $\mathbf{S}_*$, and the prior probabilities of group membership q_{*t}, the discriminant function method derives linear discriminant function and computes the posterior probabilities of an observation belonging to each group

$$p_t(\mathbf{x}) = \frac{\exp(-0.5D_t^2(\mathbf{x}))}{\sum_{u=1}^{g}\exp(-0.5D_u^2(\mathbf{x}))}$$

where $D_t^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_{*t})' \mathbf{S}_*^{-1} (\mathbf{x} - \mathbf{m}_{*t}) - 2\log(q_{*t})$ is the generalized squared distance from $\mathbf{x}$ to group t.

5. Draw a random uniform variate u, between 0 and 1, for each observation with missing group value. With the posterior probabilities, $p_1(\mathbf{x}) + p_2(\mathbf{x}) + \ldots, + p_g(\mathbf{x}) = 1$, the discriminant function method imputes $Y_j = 1$ if the value of u is less than $p_1(\mathbf{x})$, $Y_j = 2$ if the value is greater than or equal to $p_1(\mathbf{x})$ but less than $p_1(\mathbf{x}) + p_2(\mathbf{x})$, and so on.

Monotone and FCS Logistic Regression Methods

The logistic regression method is another imputation method available for classification variables. In the logistic regression method, a logistic regression model is fitted for a classification variable with a set of covariates constructed from the effects, where the classification variable is an ordinal response or a nominal response variable.

In the MI procedure, ordered values are assigned to response levels in ascending sorted order. If the response variable Y takes values in $\{1, \ldots, K\}$, then for ordinal response models, the cumulative model has the form

$$\text{logit}(\Pr(Y \le j|\mathbf{x})) = \log\left(\frac{\Pr(Y \le j|\mathbf{x})}{1 - \Pr(Y \le j|\mathbf{x})}\right) = \alpha_j + \boldsymbol{\beta}'\mathbf{x}, \quad j = 1, \ldots, K-1$$

where $\alpha_1, \ldots, \alpha_{K-1}$ are K-1 intercept parameters, and $\boldsymbol{\beta}$ is the vector of slope parameters.

For nominal response logistic models, where the K possible responses have no natural ordering, the generalized logit model has the form

$$\log\left(\frac{\Pr(Y = j \mid \mathbf{x})}{\Pr(Y = K \mid \mathbf{x})}\right) = \alpha_j + \boldsymbol{\beta}_j'\mathbf{x}, \quad j = 1, \ldots, K-1$$

where the $\alpha_1, \ldots, \alpha_{K-1}$ are K-1 intercept parameters, and the $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}$ are K-1 vectors of slope parameters.

Binary Response Logistic Regression

For a binary classification variable, based on the fitted regression model, a new logistic regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable (Rubin 1987, pp. 167–170).

For a binary variable Y with responses 1 and 2, a logistic regression model is fitted using observations with observed values for the imputed variable Y:

$$\text{logit}(p_1) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

where $X_1, X_2, \ldots, X_p$ are covariates for Y, $p_1 = \Pr(Y = 1|X_1, X_2, \ldots, X_p)$, and $\text{logit}(p_1) = \log(p_1/(1-p_1))$

The fitted model includes the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ and the associated covariance matrix $\mathbf{V}$.

The following steps are used to generate imputed values for a binary variable Y with responses 1 and 2:

1. New parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(p)})$ are drawn from the posterior predictive distribution of the parameters.

   $$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \mathbf{V}_h' \mathbf{Z}$$

   where $\mathbf{V}_h$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V} = \mathbf{V}_h' \mathbf{V}_h$, and $\mathbf{Z}$ is a vector of $p + 1$ independent random normal variates.

2. For an observation with missing Y_j and covariates $x_1, x_2, \ldots, x_p$, compute the predicted probability that Y= 1:

   $$p_1 = \frac{\exp(\mu_1)}{1 + \exp(\mu_1)}$$

   where $\mu_1 = \beta_{*0} + \beta_{*1}\, x_1 + \beta_{*2}\, x_2 + \ldots + \beta_{*(p)}\, x_p$.

3. Draw a random uniform variate, u, between 0 and 1. If the value of u is less than p_1, impute Y= 1; otherwise impute Y= 2.

The binary logistic regression imputation method can be extended to include the ordinal classification variables with more than two levels of responses, and the nominal classification variables. The LINK=LOGIT and LINK=GLOGIT options can be used to specify the cumulative logit model and the generalized logit model, respectively. The options ORDER= and DESCENDING can be used to specify the sort order for the levels of the imputed variables.

Ordinal Response Logistic Regression

For an ordinal classification variable, based on the fitted regression model, a new logistic regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable.

For a variable Y with ordinal responses 1, 2, …, K, a logistic regression model is fitted using observations with observed values for the imputed variable Y:

$$\text{logit}(p_j) = \alpha_j + \beta_1\, X_1 + \beta_2\, X_2 + \ldots + \beta_p\, X_p$$

where $X_1, X_2, \ldots, X_p$ are covariates for Y and $p_j = \Pr(Y \leq j | X_1, X_2, \ldots, X_k)$.

The fitted model includes the regression parameter estimates $\hat{\alpha} = (\hat{\alpha}_0, \ldots, \hat{\alpha}_{K-1})$ and $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, and their associated covariance matrix $\mathbf{V}$.

The following steps are used to generate imputed values for an ordinal classification variable Y with responses 1, 2, …, K:

1. New parameters γ_* are drawn from the posterior predictive distribution of the parameters.

   $$\gamma_* = \hat{\gamma} + \mathbf{V}_h' \mathbf{Z}$$

   where $\hat{\gamma} = (\hat{\alpha}, \hat{\boldsymbol{\beta}})$, $\mathbf{V}_h$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V} = \mathbf{V}_h' \mathbf{V}_h$, and $\mathbf{Z}$ is a vector of $p + K - 1$ independent random normal variates.

2. For an observation with missing Y and covariates $x_1, x_2, \ldots, x_k$, compute the predicted cumulative probability for $Y \leq j$:

$$p_j = \text{pr}(Y \leq j) = \frac{e^{\alpha_j + \mathbf{x}'\boldsymbol{\beta}}}{e^{\alpha_j + \mathbf{x}'\boldsymbol{\beta}} + 1}$$

3. Draw a random uniform variate, u, between 0 and 1, then impute

$$Y = \begin{cases} 1 & \text{if } u < p_1 \\ k & \text{if } p_{k-1} \leq u < p_k \\ K & \text{if } p_{K-1} \leq u \end{cases}$$

Nominal Response Logistic Regression

For a nominal classification variable, based on the fitted regression model, a new logistic regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable.

For a variable Y with nominal responses 1, 2, ..., K, a logistic regression model is fitted using observations with observed values for the imputed variable Y:

$$\log\left(\frac{p_j}{p_K}\right) = \alpha_j + \beta_{j1} X_1 + \beta_{j2} X_2 + \ldots + \beta_{jp} X_p$$

where $X_1, X_2, \ldots, X_p$ are covariates for Y and $p_j = \Pr(Y = j | X_1, X_2, \ldots, X_p)$.

The fitted model includes the regression parameter estimates $\hat{\alpha} = (\hat{\alpha}_0, \ldots, \hat{\alpha}_{K-1})$ and $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_0, \ldots, \hat{\beta}_{K-1})$, and their associated covariance matrix $\mathbf{V}$, where $\hat{\boldsymbol{\beta}}_j = (\hat{\beta}_{j0}, \hat{\beta}_{j1}, \ldots, \hat{\beta}_{jp})$,

The following steps are used to generate imputed values for a nominal classification variable Y with responses 1, 2, ..., K:

1. New parameters γ_* are drawn from the posterior predictive distribution of the parameters.

$$\gamma_* = \hat{\gamma} + \mathbf{V}_h' \mathbf{Z}$$

where $\hat{\gamma} = (\hat{\alpha}, \hat{\boldsymbol{\beta}})$, $\mathbf{V}_h$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V} = \mathbf{V}_h' \mathbf{V}_h$, and $\mathbf{Z}$ is a vector of $p + K - 1$ independent random normal variates.

2. For an observation with missing Y and covariates $x_1, x_2, \ldots, x_k$, compute the predicted probability for Y= j, j=1, 2, ..., K-1:

$$\text{pr}(Y = j) = \frac{e^{\alpha_j + \mathbf{x}'\boldsymbol{\beta}_j}}{\sum_{k=1}^{K-1} e^{\alpha_k + \mathbf{x}'\boldsymbol{\beta}_k} + 1}$$

and

$$\text{pr}(Y = K) = \frac{1}{\sum_{k=1}^{K-1} e^{\alpha_k + \mathbf{x}'\boldsymbol{\beta}_k} + 1}$$

3. Compute the cumulative probability for $\mathrm{Y} \leq j$:

$$P_j = \sum_{k=1}^{j} \mathrm{pr}(\mathrm{Y} = k)$$

4. Draw a random uniform variate, u, between 0 and 1, then impute

$$Y = \begin{cases} 1 & \text{if } u < p_1 \\ k & \text{if } p_{k-1} \leq u < p_k \\ K & \text{if } p_{K-1} \leq u \end{cases}$$

Logistic Regression with Augmented Data

In a logistic regression model, you might not be able to find the maximum likelihood estimates of the parameters if there is no overlap of the sample points from response groups—that is, if the data points have either a complete separation pattern or a quasi-complete separation pattern.

Complete separation of data points occurs when a linear combination of predictors correctly allocates all observations to their response groups. Quasi-complete separation occurs when a linear combination of predictors correctly allocates all observations to their response groups except for a subset of observations where the values of linear combinations of predictors are identical. For more information about complete separation patterns and quasi-complete separation patterns, see the section "Existence of Maximum Likelihood Estimates" on page 5565 in Chapter 74, "The LOGISTIC Procedure."

To address the separation issue in multiple imputation, White, Daniel, and Royston (2010) add observations to each response group and then use the augmented data to fit a weighted logistic regression. In each response group, $2p$ observations are added, where p is the number of predictors. More specifically, corresponding to each predictor, two observations are added: the first with the predictor mean minus the predictor standard deviation, and the second with the predictor mean plus the predictor standard deviation. In both observations, the values of other predictors are fixed at their corresponding means. Each additional observation contributes the same weight, and the total added weight is p+1. Each available observation in the data set (before augmentation) has a weight of 1. With this approach, there is an overlap of sample points, and maximum likelihood estimates can be obtained.

In the MONOTONE and FCS statements, the LIKELIHOOD=AUGMENT suboption in the LOGISTIC option requests maximum likelihood estimates based on augmented data. When LIKELIHOOD=AUGMENT, you can use the WEIGHT=*w* option to specify the total added weight *w* explicitly, or you can use the WEIGHT=NPARM option to specify the number of parameters as the total added weight. More specifically, for logistic regression models that consist only of p continuous effects, the added weight is p+1 for a simple binary logistic model, p+k–1 for an ordinal response model, and (p+1) (k–1) for a nominal response model, where k is the number of response levels.

If the ratio between the number of parameters and the number of available observations (before augmentation) is large, the effect from the added observations in the computation of maximum likelihood estimates can be significant. You can use the MULT=*m* suboption in the WEIGHT=NPARM option to reduce the total added weight, where the multiplier $0 < m \leq 1$. The resulting total added weight is then *m* times the number of parameters. Alternatively, you can use the WEIGHT=*w* option to specify a smaller total added weight *w* explicitly.

Monotone Propensity Score Method

The propensity score method is another imputation method available for continuous variables when the data set has a monotone missing pattern.

A propensity score is generally defined as the conditional probability of assignment to a particular treatment given a vector of observed covariates (Rosenbaum and Rubin 1983). In the propensity score method, for a variable with missing values, a propensity score is generated for each observation to estimate the probability that the observation is missing. The observations are then grouped based on these propensity scores, and an approximate Bayesian bootstrap imputation (Rubin 1987, p. 124) is applied to each group (Lavori, Dawson, and Shera 1995).

The propensity score method uses the following steps to impute values for variable Y_j with missing values:

1. Creates an indicator variable R_j with the value 0 for observations with missing Y_j and 1 otherwise.
2. Fits a logistic regression model

   $$\text{logit}(p_j) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

   where $X_1, X_2, \ldots, X_k$ are covariates for Y_j, $p_j = \Pr(R_j = 0 | X_1, X_2, \ldots, X_k)$, and $\text{logit}(p) = \log(p/(1-p))$.
3. Creates a propensity score for each observation to estimate the probability that it is missing.
4. Divides the observations into a fixed number of groups (typically assumed to be five) based on these propensity scores.
5. Applies an approximate Bayesian bootstrap imputation to each group. In group k, suppose that Y_{obs} denotes the n_1 observations with nonmissing Y_j values and Y_{mis} denotes the n_0 observations with missing Y_j. The approximate Bayesian bootstrap imputation first draws n_1 observations randomly with replacement from Y_{obs} to create a new data set Y^*_{obs}. This is a nonparametric analog of drawing parameters from the posterior predictive distribution of the parameters. The process then draws the n_0 values for Y_{mis} randomly with replacement from Y^*_{obs}.

Steps 1 through 5 are repeated sequentially for each variable with missing values.

The propensity score method was originally designed for a randomized experiment with repeated measures on the response variables. The goal was to impute the missing values on the response variables. The method uses only the covariate information that is associated with whether the imputed variable values are missing; it does not use correlations among variables. It is effective for inferences about the distributions of individual imputed variables, such as a univariate analysis, but it is not appropriate for analyses that involve relationship among variables, such as a regression analysis (Schafer 1999, p. 11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

FCS Methods for Data Sets with Arbitrary Missing Patterns

For a data set with an arbitrary missing data pattern, you can use FCS methods to impute missing values for all variables, assuming the existence of a joint distribution for these variables (Brand 1999; Van Buuren

2007). FCS method involves two phases in each imputation: the preliminary filled-in phase followed by the imputation phase.

At the filled-in phase, the missing values for all variables are filled in sequentially over the variables taken one at a time. The missing values for each variable are filled in using the specified method, or the default method for the variable if a method is not specified, with preceding variables serving as the covariates. These filled-in values provide starting values for these missing values at the imputation phase.

At the imputation phase, the missing values for each variable are imputed using the specified method and covariates at each iteration. The default method for the variable is used if a method is not specified, and the remaining variables are used as covariates if the set of covariates is not specified. After a number of iterations, as specified with the NBITER= option, the imputed values in each variable are used for the imputation. At each iteration, the missing values are imputed sequentially over the variables taken one at a time.

The MI procedure orders the variables as they are ordered in the VAR statement. For example, if the order of the p variables in the VAR statement is $Y_1, Y_2, \ldots, Y_p$, then $Y_1, Y_2, \ldots, Y_p$ (in that order) are used in the filled-in and imputation phases.

The filled-in phase replaces missing values with filled-in values for each variable. That is, with p variables $Y_1, Y_2, \ldots, Y_p$ (in that order), the missing values are filled in by using the sequence,

$$
\begin{aligned}
\boldsymbol{\theta}_1^{(0)} &\sim P(\boldsymbol{\theta}_1 \mid Y_{1(obs)}) \\
Y_{1(*)}^{(0)} &\sim P(Y_1 \mid \boldsymbol{\theta}_1^{(0)}) \\
Y_1^{(0)} &= (Y_{1(obs)}, Y_{1(*)}^{(0)}) \\
&\ldots \\
&\ldots \\
\boldsymbol{\theta}_p^{(0)} &\sim P(\boldsymbol{\theta}_p \mid Y_1^{(0)}, \ldots, Y_{p-1}^{(0)}, Y_{p(obs)}) \\
Y_{p(*)}^{(0)} &\sim P(Y_p \mid \boldsymbol{\theta}_p^{(0)}) \\
Y_p^{(0)} &= (Y_{p(obs)}, Y_{p(*)}^{(0)})
\end{aligned}
$$

where $Y_{j(obs)}$ is the set of observed Y_j values, $Y_{j(*)}^{(0)}$ is the set of filled-in Y_j values, $Y_j^{(0)}$ is the set of both observed and filled-in Y_j values, and $\boldsymbol{\theta}_j^{(0)}$ is the set of simulated parameters for the conditional distribution of Y_j given variables $Y_1, Y_2, \ldots, Y_{j-1}$.

For each variable Y_j with missing values, the corresponding imputation method is used to fit the model with covariates $Y_1, Y_2, \ldots, Y_{j-1}$. The observed observations for Y_j, which might include observations with filled-in values for $Y_1, Y_2, \ldots, Y_{j-1}$, are used in the model fitting. With this resulting model, a new model is drawn and then used to impute missing values for Y_j.

The imputation phase replaces these filled-in values $Y_{j(*)}^{(0)}$ with imputed values for each variable sequentially at each iteration. That is, with p variables $Y_1, Y_2, \ldots, Y_p$ (in that order), the missing values are imputed with the sequence at iteration $t + 1$,

$$
\begin{aligned}
\boldsymbol{\theta}_1^{(t+1)} &\sim P(\boldsymbol{\theta}_1 \mid Y_{1(obs)}, Y_2^{(t)}, \ldots, Y_p^{(t)}) \\
Y_{1(*)}^{(t+1)} &\sim P(Y_1 \mid \boldsymbol{\theta}_1^{(t+1)}) \\
Y_1^{(t+1)} &= (Y_{1(obs)}, Y_{1(*)}^{(t+1)}) \\
&\ldots \\
&\ldots \\
\boldsymbol{\theta}_p^{(t+1)} &\sim P(\boldsymbol{\theta}_p \mid Y_1^{(t+1)}, \ldots, Y_{p-1}^{(t+1)}, Y_{p(obs)}) \\
Y_{p(*)}^{(t+1)} &\sim P(Y_p \mid \boldsymbol{\theta}_p^{(t+1)}) \\
Y_p^{(t+1)} &= (Y_{p(obs)}, Y_{p(*)}^{(t+1)})
\end{aligned}
$$

where $Y_{j(obs)}$ is the set of observed Y_j values, $Y_{j(*)}^{(t+1)}$ is the set of imputed Y_j values at iteration $t+1$, $Y_{j(*)}^{(t)}$ is the set of filled-in Y_j values ($t = 0$) or the set of imputed Y_j values at iteration t ($t > 0$), $Y_j^{(t+1)}$ is the set of both observed and imputed Y_j values at iteration $t+1$, and $\boldsymbol{\theta}_j^{(t+1)}$ is the set of simulated parameters for the conditional distribution of Y_j given covariates constructed from $Y_1, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_p$.

At each iteration, a specified model is fitted for each variable with missing values by using observed observations for that variable, which might include observations with imputed values for other variables. With this resulting model, a new model is drawn and then used to impute missing values for the imputed variable.

The steps are iterated long enough for the results to reliably simulate an approximately independent draw of the missing values for an imputed data set.

The imputation methods used in the filled-in and imputation phases are similar to the corresponding monotone methods for monotone missing data. You can use a regression method or a predictive mean matching method to impute missing values for a continuous variable, a logistic regression method for a classification variable with a binary or ordinal response, and a discriminant function method for a classification variable with a binary or nominal response. See the sections "Monotone and FCS Regression Methods" on page 6100, "Monotone and FCS Predictive Mean Matching Methods" on page 6101, "Monotone and FCS Discriminant Function Methods" on page 6102, and "Monotone and FCS Logistic Regression Methods" on page 6104 for these methods.

The FCS method requires fewer iterations than the MCMC method (Van Buuren 2007). Often, as few as five or 10 iterations are enough to produce satisfactory results (Van Buuren 2007; Brand 1999).

Checking Convergence in FCS Methods

The parameters used in the imputation step at each iteration can be saved in an output data set with the OUTITER= option. These include the means and standard deviations. You can then monitor the convergence by displaying trace plots for those parameter values with the PLOTS=TRACE option.

A trace plot for a parameter ξ is a scatter plot of successive parameter estimates ξ_i against the iteration number i. The plot provides a simple way to examine the convergence behavior of the estimation algorithm

for ξ. Long-term trends in the plot indicate that successive iterations are highly correlated and that the series of iterations has not converged.

You can display trace plots for the variable means and standard deviations. You can also request logarithmic transformations for positive parameters in the plots with the LOG option. With the LOG option, if a parameter value is less than or equal to zero, then the value is not displayed in the corresponding plot.

See Example 77.8 for a usage of the trace plot.

MCMC Method for Arbitrary Missing Multivariate Normal Data

The Markov chain Monte Carlo (MCMC) method originated in physics as a tool for exploring equilibrium distributions of interacting molecules. In statistical applications, it is used to generate pseudorandom draws from multidimensional and otherwise intractable probability distributions via Markov chains. A Markov chain is a sequence of random variables in which the distribution of each element depends only on the value of the previous element.

In MCMC simulation, you constructs a Markov chain long enough for the distribution of the elements to stabilize to a stationary distribution, which is the distribution of interest. Repeatedly simulating steps of the chain simulates draws from the distribution of interest. See Schafer (1997) for a detailed discussion of this method.

In Bayesian inference, information about unknown parameters is expressed in the form of a posterior probability distribution. This posterior distribution is computed using Bayes' theorem,

$$p(\boldsymbol{\theta}|y) = \frac{p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

MCMC has been applied as a method for exploring posterior distributions in Bayesian inference. That is, through MCMC, you can simulate the entire joint posterior distribution of the unknown quantities and obtain simulation-based estimates of posterior parameters that are of interest.

In many incomplete-data problems, the observed-data posterior $p(\boldsymbol{\theta}|Y_{obs})$ is intractable and cannot easily be simulated. However, when Y_{obs} is augmented by an estimated or simulated value of the missing data Y_{mis}, the complete-data posterior $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis})$ is much easier to simulate. Assuming that the data are from a multivariate normal distribution, data augmentation can be applied to Bayesian inference with missing data by repeating the following steps:

1. The imputation I-step
Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. That is, if you denote the variables with missing values for observation i by $Y_{i(mis)}$ and the variables with observed values by $Y_{i(obs)}$, then the I-step draws values for $Y_{i(mis)}$ from a conditional distribution for $Y_{i(mis)}$ given $Y_{i(obs)}$.

2. The posterior P-step
Given a complete sample, the P-step simulates the posterior population mean vector and covariance matrix. These new estimates are then used in the next I-step. Without prior information about the parameters, a noninformative prior is used. You can also use other informative priors. For example, a prior information about the covariance matrix can help to stabilize the inference about the mean vector for a near singular covariance matrix.

That is, with a current parameter estimate $\boldsymbol{\theta}^{(t)}$ at the tth iteration, the I-step draws $Y_{mis}^{(t+1)}$ from $p(Y_{mis}|Y_{obs}, \boldsymbol{\theta}^{(t)})$ and the P-step draws $\boldsymbol{\theta}^{(t+1)}$ from $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis}^{(t+1)})$. The two steps are iterated long enough for the results to reliably simulate an approximately independent draw of the missing values for a multiply imputed data set (Schafer 1997).

This creates a Markov chain $(Y_{mis}^{(1)}, \boldsymbol{\theta}^{(1)})$, $(Y_{mis}^{(2)}, \boldsymbol{\theta}^{(2)})$, ..., which converges in distribution to $p(Y_{mis}, \boldsymbol{\theta}|Y_{obs})$. Assuming the iterates converge to a stationary distribution, the goal is to simulate an approximately independent draw of the missing values from this distribution.

To validate the imputation results, you should repeat the process with different random number generators and starting values based on different initial parameter estimates.

The next three sections provide details for the imputation step, Bayesian estimation of the mean vector and covariance matrix, and the posterior step.

Imputation Step

In each iteration, starting with a given mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the imputation step draws values for the missing data from the conditional distribution Y_{mis} given Y_{obs}.

Suppose $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1', \boldsymbol{\mu}_2'\right]'$ is the partitioned mean vector of two sets of variables, Y_{obs} and Y_{mis}, where $\boldsymbol{\mu}_1$ is the mean vector for variables Y_{obs} and $\boldsymbol{\mu}_2$ is the mean vector for variables Y_{mis}.

Also suppose

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}' & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

is the partitioned covariance matrix for these variables, where $\boldsymbol{\Sigma}_{11}$ is the covariance matrix for variables Y_{obs}, $\boldsymbol{\Sigma}_{22}$ is the covariance matrix for variables Y_{mis}, and $\boldsymbol{\Sigma}_{12}$ is the covariance matrix between variables Y_{obs} and variables Y_{mis}.

By using the sweep operator (Goodnight 1979) on the pivots of the $\boldsymbol{\Sigma}_{11}$ submatrix, the matrix becomes

$$\begin{bmatrix} \boldsymbol{\Sigma}_{11}^{-1} & \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12} \\ -\boldsymbol{\Sigma}_{12}'\boldsymbol{\Sigma}_{11}^{-1} & \boldsymbol{\Sigma}_{22.1} \end{bmatrix}$$

where $\boldsymbol{\Sigma}_{22.1} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}'\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$ can be used to compute the conditional covariance matrix of $\mathbf{Y}_{mis}$ after controlling for $\mathbf{Y}_{obs}$.

For an observation with the preceding missing pattern, the conditional distribution of Y_{mis} given $Y_{obs} = \mathbf{y}_1$ is a multivariate normal distribution with the mean vector

$$\boldsymbol{\mu}_{2.1} = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{12}'\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1)$$

and the conditional covariance matrix

$$\boldsymbol{\Sigma}_{22.1} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}'\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$$

Bayesian Estimation of the Mean Vector and Covariance Matrix

Suppose that $\mathbf{Y} = (\mathbf{y}_1', \mathbf{y}_2', \ldots, \mathbf{y}_n')'$ is an $(n \times p)$ matrix made up of n $(p \times 1)$ independent vectors $\mathbf{y}_i$, each of which has a multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Lambda}$. Then the SSCP matrix

$$\mathbf{A} = \mathbf{Y}'\mathbf{Y} = \sum_i \mathbf{y}_i \mathbf{y}_i'$$

has a Wishart distribution $W(n, \mathbf{\Lambda})$.

When each observation $\mathbf{y}_i$ is distributed with a multivariate normal distribution with an unknown mean $\boldsymbol{\mu}$, then the CSSCP matrix

$$\mathbf{A} = \sum_i (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

has a Wishart distribution $W(n-1, \mathbf{\Lambda})$.

If $\mathbf{A}$ has a Wishart distribution $W(n, \mathbf{\Lambda})$, then $\mathbf{B} = \mathbf{A}^{-1}$ has an inverted Wishart distribution $W^{-1}(n, \mathbf{\Psi})$, where n is the degrees of freedom and $\mathbf{\Psi} = \mathbf{\Lambda}^{-1}$ is the precision matrix (Anderson 1984).

Note that, instead of using the parameter $\mathbf{\Psi} = \mathbf{\Lambda}^{-1}$ for the inverted Wishart distribution, Schafer (1997) uses the parameter $\mathbf{\Lambda}$.

Suppose that each observation in the data matrix $\mathbf{Y}$ has a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$. Then with a prior inverted Wishart distribution for $\mathbf{\Sigma}$ and a prior normal distribution for $\boldsymbol{\mu}$

$$\begin{aligned} \mathbf{\Sigma} &\sim W^{-1}(m, \mathbf{\Psi}) \\ \boldsymbol{\mu}|\mathbf{\Sigma} &\sim N\left(\boldsymbol{\mu}_0, \frac{1}{\tau}\mathbf{\Sigma}\right) \end{aligned}$$

where $\tau > 0$ is a fixed number.

The posterior distribution (Anderson 1984, p. 270; Schafer 1997, p. 152) is

$$\begin{aligned} \mathbf{\Sigma}|\mathbf{Y} &\sim W^{-1}\left(n+m,\ (n-1)\mathbf{S} + \mathbf{\Psi} + \frac{n\tau}{n+\tau}(\bar{\mathbf{y}} - \boldsymbol{\mu}_0)(\bar{\mathbf{y}} - \boldsymbol{\mu}_0)'\right) \\ \boldsymbol{\mu}|(\mathbf{\Sigma}, \mathbf{Y}) &\sim N\left(\frac{1}{n+\tau}(n\bar{\mathbf{y}} + \tau\boldsymbol{\mu}_0),\ \frac{1}{n+\tau}\mathbf{\Sigma}\right) \end{aligned}$$

where $(n-1)\mathbf{S}$ is the CSSCP matrix.

Posterior Step

In each iteration, the posterior step simulates the posterior population mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ from prior information for $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, and the complete sample estimates.

You can specify the prior parameter information by using one of the following methods:

- PRIOR=JEFFREYS, which uses a noninformative prior

- PRIOR=INPUT=, which provides a prior information for $\mathbf{\Sigma}$ in the data set. Optionally, it also provides a prior information for $\boldsymbol{\mu}$ in the data set.
- PRIOR=RIDGE=, which uses a ridge prior

The next four subsections provide details of the posterior step for different prior distributions.

1. A Noninformative Prior

Without prior information about the mean and covariance estimates, you can use a noninformative prior by specifying the PRIOR=JEFFREYS option. The posterior distributions (Schafer 1997, p. 154) are

$$\begin{aligned} \mathbf{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n-1,\ (n-1)\mathbf{S}\right) \\ \boldsymbol{\mu}^{(t+1)}|(\mathbf{\Sigma}^{(t+1)},\mathbf{Y}) &\sim N\left(\bar{\mathbf{y}},\ \frac{1}{n}\mathbf{\Sigma}^{(t+1)}\right) \end{aligned}$$

2. An Informative Prior for μ and Σ

When prior information is available for the parameters $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, you can provide it with a SAS data set that you specify with the PRIOR=INPUT= option:

$$\begin{aligned} \mathbf{\Sigma} &\sim W^{-1}\left(d^*,\ d^*\mathbf{S}^*\right) \\ \boldsymbol{\mu}|\mathbf{\Sigma} &\sim N\left(\boldsymbol{\mu}_0,\ \frac{1}{n_0}\mathbf{\Sigma}\right) \end{aligned}$$

To obtain the prior distribution for $\mathbf{\Sigma}$, PROC MI reads the matrix $\mathbf{S}^*$ from observations in the data set with _TYPE_='COV', and it reads $n^* = d^* + 1$ from observations with _TYPE_='N'.

To obtain the prior distribution for $\boldsymbol{\mu}$, PROC MI reads the mean vector $\boldsymbol{\mu}_0$ from observations with _TYPE_='MEAN', and it reads n_0 from observations with _TYPE_='N_MEAN'. When there are no observations with _TYPE_='N_MEAN', PROC MI reads n_0 from observations with _TYPE_='N'.

The resulting posterior distribution, as described in the section "Bayesian Estimation of the Mean Vector and Covariance Matrix" on page 6113, is given by

$$\begin{aligned} \mathbf{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n+d^*,\ (n-1)\mathbf{S}+d^*\mathbf{S}^*+\mathbf{S}_m\right) \\ \boldsymbol{\mu}^{(t+1)}\ |\ \left(\mathbf{\Sigma}^{(t+1)},\mathbf{Y}\right) &\sim N\left(\frac{1}{n+n_0}(n\bar{\mathbf{y}}+n_0\boldsymbol{\mu}_0),\ \frac{1}{n+n_0}\mathbf{\Sigma}^{(t+1)}\right) \end{aligned}$$

where

$$\mathbf{S}_m = \frac{nn_0}{n+n_0}(\bar{\mathbf{y}}-\boldsymbol{\mu}_0)(\bar{\mathbf{y}}-\boldsymbol{\mu}_0)'$$

3. An Informative Prior for Σ

When the sample covariance matrix $\mathbf{S}$ is singular or near singular, prior information about $\boldsymbol{\Sigma}$ can also be used without prior information about $\boldsymbol{\mu}$ to stabilize the inference about $\boldsymbol{\mu}$. You can provide it with a SAS data set that you specify with the PRIOR=INPUT= option.

To obtain the prior distribution for $\boldsymbol{\Sigma}$, PROC MI reads the matrix $\mathbf{S}^*$ from observations in the data set with _TYPE_='COV', and it reads n^* from observations with _TYPE_='N'.

The resulting posterior distribution for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (Schafer 1997, p. 156) is

$$\begin{aligned} \boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n+d^*,\ (n-1)\mathbf{S}+d^*\mathbf{S}^*\right) \\ \boldsymbol{\mu}^{(t+1)} \mid \left(\boldsymbol{\Sigma}^{(t+1)}, \mathbf{Y}\right) &\sim N\left(\bar{\mathbf{y}},\ \frac{1}{n}\boldsymbol{\Sigma}^{(t+1)}\right) \end{aligned}$$

Note that if the PRIOR=INPUT= data set also contains observations with _TYPE_='MEAN', then a complete informative prior for both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ will be used.

4. A Ridge Prior

A special case of the preceding adjustment is a ridge prior with $\mathbf{S}^* = \text{Diag}(\mathbf{S})$ (Schafer 1997, p. 156). That is, $\mathbf{S}^*$ is a diagonal matrix with diagonal elements equal to the corresponding elements in $\mathbf{S}$.

You can request a ridge prior by using the PRIOR=RIDGE= option. You can explicitly specify the number $d^* \geq 1$ in the PRIOR=RIDGE=d^* option. Or you can implicitly specify the number by specifying the proportion p in the PRIOR=RIDGE=p option to request $d^* = (n-1)\,p$.

The posterior is then given by

$$\begin{aligned} \boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n+d^*,\ (n-1)\mathbf{S}+d^*\text{Diag}(\mathbf{S})\right) \\ \boldsymbol{\mu}^{(t+1)} \,\Big|\, \left(\boldsymbol{\Sigma}^{(t+1)}, \mathbf{Y}\right) &\sim N\left(\bar{\mathbf{y}},\ \frac{1}{n}\boldsymbol{\Sigma}^{(t+1)}\right) \end{aligned}$$

Producing Monotone Missingness with the MCMC Method

The monotone data MCMC method was first proposed by Li (1988) and Liu (1993) described the algorithm. The method is useful especially when a data set is close to having a monotone missing pattern. In this case, the method needs to impute only a few missing values to the data set to have a monotone missing pattern in the imputed data set. Compared to a full data imputation that imputes all missing values, the monotone data MCMC method imputes fewer missing values in each iteration and achieves approximate stationarity in fewer iterations (Schafer 1997, p. 227).

You can request the monotone MCMC method by specifying the option IMPUTE=MONOTONE in the MCMC statement. The "Missing Data Patterns" table now denotes the variables with missing values by "." or "O". The value "." means that the variable is missing and will be imputed, and the value "O" means that the variable is missing and will not be imputed. The "Variance Information" and "Parameter Estimates" tables are not created.

You must specify the variables in the VAR statement. The variable order in the list determines the monotone missing pattern in the imputed data set. With a different order in the VAR list, the results will be different because the monotone missing pattern to be constructed will be different.

Assuming that the data are from a multivariate normal distribution, then like the MCMC method, the monotone MCMC method repeats the following steps:

1. The imputation I-step
Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. Only a subset of missing values are simulated to achieve a monotone pattern of missingness.

2. The posterior P-step
Given a new sample with a monotone pattern of missingness, the P-step simulates the posterior population mean vector and covariance matrix with a noninformative Jeffreys prior. These new estimates are then used in the next I-step.

Imputation Step

The I-step is almost identical to the I-step described in the section "MCMC Method for Arbitrary Missing Multivariate Normal Data" on page 6111 except that only a subset of missing values need to be simulated. To state this precisely, denote the variables with observed values for observation i by $Y_{i(obs)}$ and the variables with missing values by $Y_{i(mis)} = (Y_{i(m1)}, Y_{i(m2)})$, where $Y_{i(m1)}$ is a subset of the missing variables that will cause a monotone missingness when their values are imputed. Then the I-step draws values for $Y_{i(m1)}$ from a conditional distribution for $Y_{i(m1)}$ given $Y_{i(obs)}$.

Posterior Step

The P-step is different from the P-step described in the section "MCMC Method for Arbitrary Missing Multivariate Normal Data" on page 6111. Instead of simulating the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters from the full imputed data set, this P-step simulates the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters through simulated regression coefficients from regression models based on the imputed data set with a monotone pattern of missingness. The step is similar to the process described in the section "Monotone and FCS Regression Methods" on page 6100.

That is, for the variable Y_j, a model

$$Y_j = \beta_0 + \beta_1 Y_1 + \beta_2 Y_2 + \ldots + \beta_{j-1} Y_{j-1}$$

is fitted using n_j nonmissing observations for variable Y_j in the imputed data sets.

The fitted model consists of the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$ and the associated covariance matrix $\hat{\sigma}_j^2 \mathbf{V}_j$, where $\mathbf{V}_j$ is the usual $\mathbf{X}'\mathbf{X}$ inverse matrix from the intercept and variables $Y_1, Y_2, \ldots, Y_{j-1}$.

For each imputation, new parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(j-1)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

$$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - j)/g$$

where g is a $\chi^2_{n_j - p + j - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

$$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

where $\mathbf{V}_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj}\mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of j independent random normal variates.

These simulated values of $\boldsymbol{\beta}_*$ and σ^2_{*j} are then used to re-create the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For a detailed description of how to produce monotone missingness with the MCMC method for a multivariate normal data, see Schafer (1997, pp. 226–235).

MCMC Method Specifications

With the MCMC method, you can impute either all missing values (IMPUTE=FULL) or just enough missing values to make the imputed data set have a monotone missing pattern (IMPUTE=MONOTONE). In the process, either a single chain for all imputations (CHAIN=SINGLE) or a separate chain for each imputation (CHAIN=MULTIPLE) is used. The single chain might be somewhat more precise for estimating a single quantity such as a posterior mean (Schafer 1997, p. 138). See Schafer (1997, pp. 137–138) for a discussion of single versus multiple chains.

You can specify the number of initial burn-in iterations before the first imputation with the NBITER= option. This number is also used for subsequent chains for multiple chains. For a single chain, you can also specify the number of iterations between imputations with the NITER= option.

You can explicitly specify initial parameter values for the MCMC method with the INITIAL=INPUT= data set option. Alternatively, you can use the EM algorithm to derive a set of initial parameter values for MCMC with the option INITIAL=EM. These estimates are used as either the starting value (START=VALUE) or the starting distribution (START=DIST) for the MCMC method. For multiple chains, these estimates are used again as either the starting value (START=VALUE) or the starting distribution (START=DIST) for the subsequent chains.

You can specify the prior parameter information in the PRIOR= option. You can use a noninformative prior (PRIOR=JEFFREYS), a ridge prior (PRIOR=RIDGE), or an informative prior specified in a data set (PRIOR=INPUT).

The parameter estimates used to generate imputed values in each imputation can be saved in a data set with the OUTEST= option. Later, this data set can be read with the INEST= option to provide the reference distribution for imputing missing values for a new data set.

By default, the MCMC method uses a single chain to produce five imputations. It completes 200 burn-in iterations before the first imputation and 100 iterations between imputations. The posterior mode computed from the EM algorithm with a noninformative prior is used as the starting values for the MCMC method.

INITIAL=EM Specifications

The EM algorithm is used to find the maximum likelihood estimates for incomplete data in the EM statement. You can also use the EM algorithm to find a posterior mode, the parameter estimates that maximize the observed-data posterior density. The resulting posterior mode provides a good starting value for the MCMC method.

With the INITIAL=EM option, PROC MI uses the MLE of the parameter vector as the initial estimates in the EM algorithm for the posterior mode. You can use the ITPRINT option within the INITIAL=EM option to display the iteration history for the EM algorithm.

You can use the CONVERGE= option to specify the convergence criterion in deriving the EM posterior mode. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. By default, CONVERGE=1E–4.

You can also use the MAXITER= option to specify the maximum number of iterations of the EM algorithm. By default, MAXITER=200.

With the BOOTSTRAP option, you can use overdispersed starting values for the MCMC method. In this case, PROC MI applies the EM algorithm to a bootstrap sample, a simple random sample with replacement from the input data set, to derive the initial estimates for each chain (Schafer 1997, p. 128).

Checking Convergence in MCMC

The theoretical convergence of the MCMC method has been explored under various conditions, as described in Schafer (1997, p. 70). However, in practice, verification of convergence is not a simple matter.

The parameters used in the imputation step for each iteration can be saved in an output data set with the OUTITER= option. These include the means, standard deviations, covariances, worst linear function, and observed-data LR statistics. You can then monitor the convergence in a single chain by displaying trace plots and autocorrelations for those parameter values (Schafer 1997, p. 120). The trace and autocorrelation function plots for parameters such as variable means, covariances, and the worst linear function can be displayed by specifying the PLOTS=TRACE and PLOTS=ACF options, respectively.

You can apply the EM algorithm to a bootstrap sample to obtain overdispersed starting values for multiple chains (Gelman and Rubin 1992). This provides a conservative estimate of the number of iterations needed before each imputation.

The next four subsections describe useful statistics and plots that can be used to check the convergence of the MCMC method.

LR Statistics

You can save the observed-data likelihood ratio (LR) statistic in each iteration with the LR option in the OUTITER= data set. The statistic is based on the observed-data likelihood with parameter values used in the iteration and the observed-data maximum likelihood derived from the EM algorithm.

In each iteration, the LR statistic is given by

$$-2\log\left(\frac{f(\hat{\boldsymbol{\theta}}_i)}{f(\hat{\boldsymbol{\theta}})}\right)$$

where $f(\hat{\boldsymbol{\theta}})$ is the observed-data maximum likelihood derived from the EM algorithm and $f(\hat{\boldsymbol{\theta}}_i)$ is the observed-data likelihood for $\hat{\boldsymbol{\theta}}_i$ used in the iteration.

Similarly, you can also save the observed-data LR posterior mode statistic for each iteration with the LR_POST option. This statistic is based on the observed-data posterior density with parameter values used in each iteration and the observed-data posterior mode derived from the EM algorithm for posterior mode.

For large samples, these LR statistics tends to be approximately χ^2 distributed with degrees of freedom equal to the dimension of $\boldsymbol{\theta}$ (Schafer 1997, p. 131). For example, with a large number of iterations, if the values of the LR statistic do not behave like a random sample from the described χ^2 distribution, then there is evidence that the MCMC method has not converged.

Worst Linear Function of Parameters

The worst linear function (WLF) of parameters (Schafer 1997, pp. 129–131) is a scalar function of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that is "worst" in the sense that its function values converge most slowly among parameters in the MCMC method. The convergence of this function is evidence that other parameters are likely to converge as well.

For linear functions of parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, a worst linear function of $\boldsymbol{\theta}$ has the highest asymptotic rate of missing information. The function can be derived from the iterative values of $\boldsymbol{\theta}$ near the posterior mode in the EM algorithm. That is, an estimated worst linear function of $\boldsymbol{\theta}$ is

$$w(\boldsymbol{\theta}) = \mathbf{v}'(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

where $\hat{\boldsymbol{\theta}}$ is the posterior mode and the coefficients $\mathbf{v} = \hat{\boldsymbol{\theta}}_{(-1)} - \hat{\boldsymbol{\theta}}$ are the difference between the estimated value of $\boldsymbol{\theta}$ one step prior to convergence and the converged value $\hat{\boldsymbol{\theta}}$.

You can display the coefficients of the worst linear function, $\mathbf{v}$, by specifying the WLF option in the MCMC statement. You can save the function value from each iteration in an OUTITER= data set by specifying the WLF option within the OUTITER option. You can also display the worst linear function values from iterations in an autocorrelation plot or a trace plot by specifying PLOTS=ACF(WLF) or PLOTS=TRACE(WLF), respectively.

Note that when the observed-data posterior is nearly normal, the WLF is one of the slowest functions to approach stationarity. When the posterior is not close to normal, other functions might take much longer than the WLF to converge, as described in Schafer (1997, p.130).

Trace Plot

A trace plot for a parameter ξ is a scatter plot of successive parameter estimates ξ_i against the iteration number i. The plot provides a simple way to examine the convergence behavior of the estimation algorithm for ξ. Long-term trends in the plot indicate that successive iterations are highly correlated and that the series of iterations has not converged.

You can display trace plots for worst linear function, variable means, variable variances, and covariances of variables. You can also request logarithmic transformations for positive parameters in the plots with the LOG option. When a parameter value is less than or equal to zero, the value is not displayed in the corresponding plot. See Example 77.11 for a usage of the trace plot.

Autocorrelation Function Plot

To examine relationships of successive parameter estimates ξ, the autocorrelation function (ACF) can be used. For a stationary series, $\xi_i, i \geq 1$, in trace data, the autocorrelation function at lag k is

$$\rho_k = \frac{\mathrm{Cov}(\xi_i, \xi_{i+k})}{\mathrm{Var}(\xi_i)}$$

The sample kth order autocorrelation is computed as

$$r_k = \frac{\sum_{i=1}^{n-k}(\xi_i - \bar{\xi})(\xi_{i+k} - \bar{\xi})}{\sum_{i=1}^{n}(\xi_i - \bar{\xi})^2}$$

You can display autocorrelation function plots for the worst linear function, variable means, variable variances, and covariances of variables. You can also request logarithmic transformations for parameters in the plots with the LOG option. When a parameter has values less than or equal to zero, the corresponding plot is not created.

You specify the maximum number of lags of the series with the NLAG= option. The autocorrelations at each lag less than or equal to the specified lag are displayed in the graph. In addition, the plot also displays approximate 95% confidence limits for the autocorrelations. At lag k, the confidence limits indicate a set of approximate 95% critical values for testing the hypothesis $\rho_j = 0, j \geq k$. See Example 77.11 for an illustration of the autocorrelation function plot.

Input Data Sets

You can specify the input data set with missing values by using the DATA= option in the PROC MI statement. When an MCMC method is used, you can specify the data set that contains the reference distribution information for imputation with the INEST= option, the data set that contains initial parameter estimates for the MCMC method with the INITIAL=INPUT= option, and the data set that contains information for the prior distribution with the PRIOR=INPUT= option in the MCMC statement.

When the ADJUST option is specified in the MNAR statement, you can use the PARMS= option to specify the data set that contains adjustment parameters for the sensitivity analysis.

DATA=*SAS-data-set*

The input DATA= data set is an ordinary SAS data set that contains multivariate data with missing values.

INEST=*SAS-data-set*

The input INEST= data set is a TYPE=EST data set and contains a variable _Imputation_ to identify the imputation number. For each imputation, PROC MI reads the point estimate from the observations with _TYPE_='PARM' or _TYPE_='PARMS' and the associated covariances from the observations with _TYPE_='COV' or _TYPE_='COVB'. These estimates are used as the reference distribution to impute values for observations in the DATA= data set. When the input INEST= data set also contains observations with _TYPE_='SEED', PROC MI reads the seed information for the random number generator from these observations. Otherwise, the SEED= option provides the seed information.

INITIAL=INPUT=*SAS-data-set*

The input INITIAL=INPUT= data set is a TYPE=COV or CORR data set and provides initial parameter estimates for the MCMC method. The covariances derived from the TYPE=COV/CORR data set are divided by the number of observations to get the correct covariance matrix for the point estimate (sample mean).

If TYPE=COV, PROC MI reads the number of observations from the observations with _TYPE_='N', the point estimate from the observations with _TYPE_='MEAN', and the covariances from the observations with _TYPE_='COV'.

If TYPE=CORR, PROC MI reads the number of observations from the observations with _TYPE_='N', the point estimate from the observations with _TYPE_='MEAN', the correlations from the observations with _TYPE_='CORR', and the standard deviations from the observations with _TYPE_='STD'.

PARMS= *SAS-data-set*

The input PARMS= data set is an ordinary SAS data set that contains adjustment parameters for imputed values of the specified imputed variables.

The PARMS= data set contains variables _Imputation_ for the imputation number, the SHIFT= or DELTA= variable for the shift parameter, and the SCALE= variable for the scale parameter. Either the shift or scale variable must be included in the data set.

PRIOR=INPUT=*SAS-data-set*

The input PRIOR=INPUT= data set is a TYPE=COV data set that provides information for the prior distribution. You can use the data set to specify a prior distribution for $\boldsymbol{\Sigma}$ of the form

$$\boldsymbol{\Sigma} \sim W^{-1}\left(d^*,\, d^*\mathbf{S}^*\right)$$

where $d^* = n^* - 1$ is the degrees of freedom. PROC MI reads the matrix $\mathbf{S}^*$ from observations with _TYPE_='COV' and reads n^* from observations with _TYPE_='N'.

You can also use this data set to specify a prior distribution for $\boldsymbol{\mu}$ of the form

$$\boldsymbol{\mu} \sim N\left(\boldsymbol{\mu}_0,\, \frac{1}{n_0}\boldsymbol{\Sigma}\right)$$

PROC MI reads the mean vector $\boldsymbol{\mu}_0$ from observations with _TYPE_='MEAN' and reads n_0 from observations with _TYPE_='N_MEAN'. When there are no observations with _TYPE_='N_MEAN', PROC MI reads n_0 from observations with _TYPE_='N'.

Output Data Sets

You can specify the output data set of imputed values with the OUT= option in the PROC MI statement. When an EM statement is used, you can specify the data set that contains the original data set with missing values being replaced by the expected values from the EM algorithm by using the OUT= option in the EM statement. You can also specify the data set that contains MLE computed with the EM algorithm by using the OUTEM= option.

When an MCMC method is used, you can specify the data set that contains parameter estimates used in each imputation with the OUTEST= option in the MCMC statement, and you can specify the data set that contains parameters used in the imputation step for each iteration with the OUTITER option in the MCMC statement.

OUT=*SAS-data-set* in the PROC MI statement

The OUT= data set contains all the variables in the original data set and a new variable named _Imputation_ that identifies the imputation. For each imputation, the data set contains all variables in the input DATA= data set with missing values being replaced by imputed values. Note that when the NIMPUTE=1 option is specified, the variable _Imputation_ is not created.

OUT=*SAS-data-set* in an EM statement

The OUT= data set contains the original data set with missing values being replaced by expected values from the EM algorithm.

OUTEM=*SAS-data-set*

The OUTEM= data set is a TYPE=COV data set and contains the MLE computed with the EM algorithm. The observations with _TYPE_='MEAN' contain the estimated mean and the observations with _TYPE_='COV' contain the estimated covariances.

OUTEST=*SAS-data-set*

The OUTEST= data set is a TYPE=EST data set and contains parameter estimates used in each imputation in the MCMC method. It also includes an index variable named _Imputation_, which identifies the imputation.

The observations with _TYPE_='SEED' contain the seed information for the random number generator. The observations with _TYPE_='PARM' or _TYPE_='PARMS' contain the point estimate, and the observations with _TYPE_='COV' or _TYPE_='COVB' contain the associated covariances. These estimates are used as the parameters of the reference distribution to impute values for observations in the DATA= dataset.

Note that these estimates are the values used in the I-step before each imputation. These are not the parameter values simulated from the P-step in the same iteration. See Example 77.12 for a usage of this option.

OUTITER < (*options*) > =*SAS-data-set* in an EM statement

The OUTITER= data set in an EM statement is a TYPE=COV data set and contains parameters for each iteration. It also includes a variable _Iteration_ that provides the iteration number.

The parameters in the output data set depend on the options specified. You can specify the MEAN and COV options for OUTITER. With the MEAN option, the output data set contains the mean parameters in observations with the variable _TYPE_='MEAN'. Similarly, with the COV option, the output data set contains the covariance parameters in observations with the variable _TYPE_='COV'. When no options are specified, the output data set contains the mean parameters for each iteration.

OUTITER < (*options*) > =*SAS-data-set* in an FCS statement

The OUTITER= data set in an FCS statement is a TYPE=COV data set and contains parameters for each iteration. It also includes variables named _Imputation_ and _Iteration_, which provide the imputation number and iteration number.

The parameters in the output data set depend on the options specified. You can specify the MEAN and STD options for OUTITER. With the MEAN option, the output data set contains the mean parameters used in the imputation in observations with the variable _TYPE_='MEAN'. Similarly, with the STD option, the output data set contains the standard deviation parameters used in the imputation in observations with the variable _TYPE_='STD'. When no options are specified, the output data set contains the mean parameters for each iteration.

OUTITER <(*options*)> =*SAS-data-set* in an MCMC statement

The OUTITER= data set in an MCMC statement is a TYPE=COV data set and contains parameters used in the imputation step for each iteration. It also includes variables named _Imputation_ and _Iteration_, which provide the imputation number and iteration number.

The parameters in the output data set depend on the options specified. Table 77.6 summarizes the options available for OUTITER and the corresponding values for the output variable _TYPE_.

Table 77.6 Summary of Options for OUTITER in an MCMC statement

Option	Output Parameters	_TYPE_
MEAN	mean parameters	MEAN
STD	standard deviations	STD
COV	covariances	COV
LR	–2 log LR statistic	LOG_LR
LR_POST	–2 log LR statistic of the posterior mode	LOG_POST
WLF	worst linear function	WLF

When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. For a detailed description of the worst linear function and LR statistics, see the section "Checking Convergence in MCMC" on page 6118.

Combining Inferences from Multiply Imputed Data Sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Suppose $\hat{Q}_i$ and $\hat{W}_i$ are the point and variance estimates 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 $\overline{W}$ 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 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 $\gamma = (1+m^{-1})B/T$.

Note that the MI procedure uses the adjusted degrees of freedom, v_m^*, for inference.

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):

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

where $m$is the number of imputations and λ is the fraction of missing information.

Table 77.7 shows relative efficiencies with different values of m and λ.

Table 77.7 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 if the fraction of missing information is modest, only a small number of imputations are needed. For example, if $\lambda = 0.3$, only three imputations are needed to have a 91% efficiency and five imputations are needed to have a 94% efficiency.

Number of Imputations

In multiple imputation, the number of imputations, *m*, must be specified in advance. The classic recommendation of a small *m* is based on a relative efficiency argument. For example, with 30% missing information, only three imputations are needed for 91% efficiency and five imputations are needed for 94% efficiency. Thus, often as few as three to five imputations are adequate in multiple imputation (Rubin 1996, p. 480). For more information about the relative efficiency of an estimator based on *m* imputations, see the section "Multiple Imputation Efficiency" on page 6124.

Recent studies by Graham, Olchowski, and Gilreath (2007); Bodner (2008); and Von Hippel (2009, p. 278) note that a small number of imputations that suffice for high relative efficiency might not be adequate for other inferential goals, such as confidence intervals and *p*-values. These studies recommend much larger numbers of imputations.

Graham, Olchowski, and Gilreath (2007) use simulations to examine the effect of *m* on the power of a hypothesis test, and they recommend the use of many more imputations than the classic recommendation of three to five imputations. For example, with a 1% power falloff tolerance in multiple imputation, as compared to an infinite number of imputations, multiple imputation requires 20 imputations for 30% missing information and 40 imputations for 50% missing information (Graham, Olchowski, and Gilreath 2007, p. 212).

Bodner (2008) points out that with small *m* there is substantial imprecision in important statistics such as *p*-values and widths of confidence intervals. That is, different conclusions might be drawn for the same test in separate multiple imputation runs. Bodner (2008) uses simulation to compute the minimum number of imputations that are needed for an estimated 95% confidence interval width to achieve a specified precision. For example, for a 95% confidence interval width to be within 10% of its true value 95% of the time, multiple imputation requires 24 imputations if $\lambda = 0.3$ and 59 imputations if $\lambda = 0.5$ (Bodner 2008, p. 668), where λ is the fraction of missing information. Because λ is unknown, a conservative estimate of λ is the proportion of cases with missing values (Bodner 2008, p. 670).

Von Hippel (2009, p. 278) shows that with a small number of imputations, only the point estimates are reliable. That is, the point estimates will not change much if the missing values are imputed again. For other statistics (such as standard error and *p*-value) to be reliable, the rule of thumb is to use the percentages of

cases with missing values as the number of imputations. White, Royston, and Wood (2011, pp. 387–388) also suggest the use of this rule of thumb, at least when $\lambda \leq 0.5$, and the resulting number of imputations often provides an adequate level to reproduce the results if the missing values are imputed again.

These recent studies suggest the use of a much larger number of imputations than the classic recommendation of three to five imputations. Thus, the default number of imputations in PROC MI has been changed from NIMPUTE=5 to NIMPUTE=25 in SAS/STAT 14.1. Alternatively, you can specify NIMPUTE=PCTMISSING to use the percentage of cases with missing values as the number of imputations.

In practice, you can verify the needed number of imputations informally by replicating sets of *m* imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

Imputer's Model Versus Analyst's Model

Multiple imputation inference assumes that the model you used to analyze the multiply imputed data (the analyst's model) is the same as the model used to impute missing values in multiple imputation (the imputer's model). But in practice, the two models might not be the same (Schafer 1997, p. 139).

Schafer (1997, pp. 139–143) provides comprehensive coverage of this topic, and the following example is based on his work.

Consider a trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 with missing values. An imputer creates multiple imputations with the model $Y_3 = Y_1\ Y_2$. However, the analyst can later use the simpler model $Y_3 = Y_1$. In this case, the analyst assumes more than the imputer. That is, the analyst assumes there is no relationship between variables Y_3 and Y_2.

The effect of the discrepancy between the models depends on whether the analyst's additional assumption is true. If the assumption is true, the imputer's model still applies. The inferences derived from multiple imputations will still be valid, although they might be somewhat conservative because they reflect the additional uncertainty of estimating the relationship between Y_3 and Y_2.

On the other hand, suppose that the analyst models $Y_3 = Y_1$, and there is a relationship between variables Y_3 and Y_2. Then the model $Y_3 = Y_1$ will be biased and is inappropriate. Appropriate results can be generated only from appropriate analyst models.

Another type of discrepancy occurs when the imputer assumes more than the analyst. For example, suppose that an imputer creates multiple imputations with the model $Y_3 = Y_1$, but the analyst later fits a model $Y_3 = Y_1\ Y_2$. When the assumption is true, the imputer's model is a correct model and the inferences still hold.

On the other hand, suppose there is a relationship between Y_3 and Y_2. Imputations created under the incorrect assumption that there is no relationship between Y_3 and Y_2 will make the analyst's estimate of the relationship biased toward zero. Multiple imputations created under an incorrect model can lead to incorrect conclusions.

Thus, generally you should include as many variables as you can when doing multiple imputation. The precision you lose with included unimportant predictors is usually a relatively small price to pay for the general validity of analyses of the resultant multiply imputed data set (Rubin 1996). But at the same time, you need to keep the model building and fitting feasible (Barnard and Meng 1999, pp. 19–20).

To produce high-quality imputations for a particular variable, the imputation model should also include variables that are potentially related to the imputed variable and variables that are potentially related to the missingness of the imputed variable (Schafer 1997, p. 143).

Similar suggestions were also given by Van Buuren, Boshuizen, and Knook (1999, p. 687). They recommend that the imputation model include three sets of covariates: variables in the analyst's model, variables associated with the missingness of the imputed variable, and variables correlated with the imputed variable. They also recommend the removal of the covariates not in the analyst's model if they have too many missing values for observations with missing imputed variables.

Note that it is good practice to include a description of the imputer's model with the multiply imputed data set (Rubin 1996, p. 479). That way, the analysts will have information about the variables involved in the imputation and which relationships among the variables have been implicitly set to zero.

Parameter Simulation versus Multiple Imputation

As an alternative to multiple imputation, parameter simulation can also be used to analyze the data for many incomplete-data problems. Although the MI procedure does not offer parameter simulation, the trade-offs between the two methods (Schafer 1997, pp. 89–90, 135–136) are examined in this section.

The parameter simulation method simulates random values of parameters from the observed-data posterior distribution and makes simple inferences about these parameters (Schafer 1997, p. 89). When a set of well-defined population parameters $\boldsymbol{\theta}$ are of interest, parameter simulation can be used to directly examine and summarize simulated values of $\boldsymbol{\theta}$. This usually requires a large number of iterations, and involves calculating appropriate summaries of the resulting dependent sample of the iterates of the $\boldsymbol{\theta}$. If only a small set of parameters are involved, parameter simulation is suitable (Schafer 1997).

Multiple imputation requires only a small number of imputations. Generating and storing a few imputations can be more efficient than generating and storing a large number of iterations for parameter simulation.

When fractions of missing information are low, methods that average over simulated values of the missing data, as in multiple imputation, can be much more efficient than methods that average over simulated values of $\boldsymbol{\theta}$ as in parameter simulation (Schafer 1997).

Sensitivity Analysis for the MAR Assumption

Multiple imputation usually assumes that the data are missing at random (MAR). Suppose the data set contains variables $Y = (Y_{obs}, Y_{mis})$, where Y_{obs} are fully observed variables and Y_{mis} is a variable that contains missing observations. Also suppose $\mathbf{R}$ is a response indicator whose element is 0 or 1, depending on whether Y_{mis} is missing or observed. Then the MAR assumption is that the probability that a Y_{mis} observation is missing can depend on Y_{obs} but not on Y_{mis}. That is,

$$\mathrm{pr}(\,\mathbf{R} \mid Y_{obs}, Y_{mis}\,) = \mathrm{pr}(\,\mathbf{R} \mid Y_{obs}\,)$$

The MAR assumption cannot be verified, because the missing values are not observed. In clinical trials, for a study that assumes MAR, the sensitivity of inferences to departures from the MAR assumption should be examined, as recommended by the National Research Council (2010, p. 111):

> Recommendation 15: Sensitivity analysis should be part of the primary reporting of findings from clinical trials. Examining sensitivity to the assumptions about the missing data mechanism should be a mandatory component of reporting.

If it is plausible that the missing data are not MAR, you can perform sensitivity analysis under the missing not at random (MNAR) assumption. You can generate inferences for various scenarios under MNAR and then examine the results. If the results under MNAR differ from the results under MAR, then the conclusion under MAR is in question.

Based on the factorization of the joint distribution $\mathrm{pr}(Y, \mathbf{R})$, there are two common strategies for sensitivity analysis under MNAR: the pattern-mixture model approach and the selection model approach. The pattern-mixture model approach is implemented in the MI procedure because it is natural and straightforward.

Pattern-Mixture Model Approach

In the pattern-mixture model approach (Little 1993; Molenberghs and Kenward 2007, pp. 30, 34–37; National Research Council 2010, pp. 88–89), the joint distribution is factorized as

$$\mathrm{pr}(Y, \mathbf{R}) = \mathrm{pr}(Y \mid \mathbf{R})\ \mathrm{pr}(\mathbf{R})$$

This allows for different distributions for missing values and for observed values. For example,

$$\mathrm{pr}(Y, \mathbf{R}) = \mathrm{pr}(Y \mid \mathbf{R})\ \mathrm{pr}(\mathbf{R}) = \mathrm{pr}(Y \mid \mathbf{R} = \mathbf{1})\ \mathrm{pr}(\mathbf{R} = \mathbf{1}) + \mathrm{pr}(Y \mid \mathbf{R} = \mathbf{0})\ \mathrm{pr}(\mathbf{R} = \mathbf{0})$$

which is a mixture of distributions for two different patterns. Here, the "pattern" refers to a group of observations that have the same distribution; the term is not used in the same sense as "missing data pattern."

In the pattern-mixture model approach, the joint distribution is factored as

$$\mathrm{pr}(Y_{obs}, Y_{mis}, \mathbf{R}) = \mathrm{pr}(Y_{mis} \mid Y_{obs}, \mathbf{R})\ \mathrm{pr}(Y_{obs}, \mathbf{R})$$

and under the MNAR assumption,

$$\mathrm{pr}(Y_{mis} \mid Y_{obs}, \mathbf{R} = 0) \neq \mathrm{pr}(Y_{mis} \mid Y_{obs}, \mathbf{R} = 1)$$

It is straightforward to create imputations by using pattern-mixture models. The next three sections provide details for this approach.

Selection Model Approach

In the selection model approach (Rubin 1987, p. 207; Little and Rubin 2002, pp. 313–314; Molenberghs and Kenward 2007, p. 30), the joint distribution is factorized as

$$\mathrm{pr}(Y, \mathbf{R}) = \mathrm{pr}(\mathbf{R} \mid Y)\ \mathrm{pr}(Y)$$

where $Y = (Y_{obs}, Y_{mis})$, $\mathrm{pr}(Y)$ is the marginal distribution of Y, and $\mathrm{pr}(\mathbf{R} \mid Y)$ is the conditional distribution of the missing mechanism R given Y. The term "selection" comes from the specification of R that selects individuals to be observed in the conditional distribution $\mathrm{pr}(\mathbf{R} \mid Y)$. Both distributions, $\mathrm{pr}(Y)$ and $\mathrm{pr}(\mathbf{R} \mid Y)$, must be specified for the analysis. The MI procedure does not provide this approach.

Multiple Imputation with Pattern-Mixture Models

For $Y = (Y_{obs}, Y_{mis})$, the joint distribution of Y and $\mathbf{R}$ can be expressed as

$$\mathrm{pr}(Y_{obs}, Y_{mis}, \mathbf{R}) = \mathrm{pr}(Y_{mis} \mid Y_{obs}, \mathbf{R})\ \mathrm{pr}(Y_{obs}, \mathbf{R})$$

Under the MAR assumption,

$$\mathrm{pr}(\,\mathbf{R} \mid Y_{obs}, Y_{mis}) = \mathrm{pr}(\,\mathbf{R} \mid Y_{obs})$$

and it can be shown that

$$\mathrm{pr}(\,Y_{mis} \mid Y_{obs}, \mathbf{R}\,) = \mathrm{pr}(\,Y_{mis} \mid Y_{obs}\,)$$

That is,

$$\mathrm{pr}(\,Y_{mis} \mid Y_{obs}, \mathbf{R} = 0\,) = \mathrm{pr}(\,Y_{mis} \mid Y_{obs}, \mathbf{R} = 1\,)$$

Thus the posterior distribution $\mathrm{pr}(\,Y_{mis} \mid Y_{obs}, \mathbf{R} = 1\,)$ can be used to create imputations for missing data.

Under the MNAR assumption, each pattern that has missing Y_{mis} values might have a different distribution than the corresponding pattern that has observed Y_{mis} values. For example, in a clinical trial, suppose the data set contains an indicator variable Trt, with a value of 1 for patients in the treatment group and a value of 0 for patients in the placebo control group, a variable Y_0 for the baseline efficacy score, and a variable Y for the efficacy score at a follow-up visit. Assume that Trt and Y_0 are fully observed and Y is not fully observed. The indicator variable $\mathbf{R}$ is 0 or 1, depending on whether Y is missing or observed.

Then, under the MAR assumption,

$$\mathrm{pr}(\,Y \mid \mathsf{Trt} = 0, Y_0, \mathbf{R} = 0\,) = \mathrm{pr}(\,Y \mid \mathsf{Trt} = 0, Y_0, \mathbf{R} = 1\,)$$

and

$$\mathrm{pr}(\,Y \mid \mathsf{Trt} = 1, Y_0, \mathbf{R} = 0\,) = \mathrm{pr}(\,Y \mid \mathsf{Trt} = 1, Y_0, \mathbf{R} = 1\,)$$

Under the MNAR assumption,

$$\mathrm{pr}(\,Y \mid \mathsf{Trt} = 0, Y_0, \mathbf{R} = 0\,) \neq \mathrm{pr}(\,Y \mid \mathsf{Trt} = 0, Y_0, \mathbf{R} = 1\,)$$

or

$$\mathrm{pr}(\,Y \mid \mathsf{Trt} = 1, Y_0, \mathbf{R} = 0\,) \neq \mathrm{pr}(\,Y \mid \mathsf{Trt} = 1, Y_0, \mathbf{R} = 1\,)$$

Thus, under MNAR, missing Y values in the treatment group can be imputed from a posterior distribution generated from observations in the control group, and the imputed values can be adjusted to reflect the systematic difference between the distributions for missing and observed Y values.

Multiple imputation inference, under either the MAR or MNAR assumption, 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 by using other SAS procedures.
3. The results from the *m* complete data sets are combined for the inference.

For sensitivity analysis, you must specify the MNAR statement together with a MONOTONE statement or an FCS statement. When you specify a MONOTONE statement, the variables that have missing values are imputed sequentially in each imputation. When you specify an FCS statement, each imputation is carried out in two phases: the preliminary filled-in phase, followed by the imputation phase. The variables that have missing values are imputed sequentially for a number of burn-in iterations before the imputation.

Under the MNAR assumption, the following steps are used to impute missing values for each imputed variable in each imputation (when you specify a MONOTONE statement) or in each iteration (when you specify an FCS statement):

1. For each imputed variable, a conditional model, such as a regression model for continuous variables, is fitted using either all applicable observations or a specified subset of observations.
2. A new model is simulated from the posterior predictive distribution of the fitted model.
3. Missing values of the variable are imputed based on the new model, and the imputed values for a specified subset of observations can be adjusted using specified shift and scale parameters.

The next two sections provide details for specifying subsets of observations for imputation models and for adjusting imputed values.

Specifying Sets of Observations for Imputation in Pattern-Mixture Models

By default, all available observations are used to derive the imputation model. By using the MODEL option in the MNAR statement, you can specify the set of observations that are used to derive the model. You specify a classification variable (*obs-variable*) by using the option MODELOBS= (*obs-variable*= '*level1*' < '*level2*' **...** >). The MI procedure uses the group of observations for which *obs-variable* equals one of the specified classification levels.

When you use the MNAR statement together with a MONOTONE statement, you can also use the MODELOBS=CCMV and MODELOBS=NCMV options to specify the set of observations for deriving the imputation model. For a monotone missing pattern data set that contains the variables $Y_1, Y_2, \ldots, Y_p$ (in that order), there are at most p groups of observations such that the same number of variables is observed for observations in each group. The complete-case missing values (CCMV) method (Little 1993; Molenberghs and Kenward 2007, p. 35) uses the group of observations for which all variables are observed (complete cases) to derive the imputation model. The neighboring-case missing values (NCMV) method (Molenberghs and Kenward 2007, pp. 35–36) uses only the neighboring group of observations (that is, for Y_j, the group of observations with Y_j observed and Y_{j+1} missing).

In PROC MI, the option MODELOBS=CCMV(K=*k*) uses the k groups of observations together with as many observed variables as possible to derive the imputation model. For instance, specifying K=1 (which is the default) uses observations from the group that has all variables observed (complete cases). Specifying K=2 uses observations from the two groups that have the most variables observed (the group of observations that has all variables observed and the group of observations that has Y_{p-1} observed but Y_p missing).

For an imputed variable Y_j, the option MODELOBS=NCMV(K=*k*) uses the k closest groups of observations that have observed Y_j but have as few observed variables as possible to derive the imputation model. For instance, specifying K=1 (which is the default) uses the group of observations that has Y_j observed but Y_{j+1} missing (neighboring cases). Specifying K=2 uses observations from the two closest groups that have Y_j observed (the group of observations that has Y_j observed but Y_{j+1} missing, and the group of observations that has Y_{j+1} observed and Y_{j+2} missing).

When you use the MNAR statement together with an FCS statement, the MODEL option applies only after the preliminary filled-in phase in each of the imputations.

For an illustration of the MODEL option, see Example 77.15.

Adjusting Imputed Values in Pattern-Mixture Models

It is straightforward to specify pattern-mixture models under the MNAR assumption. When you impute continuous variables by using the regression and predictive mean matching methods, you can adjust the imputed values directly (Carpenter and Kenward 2013, pp. 237–239; Van Buuren 2012, pp. 88–89). When you impute classification variables by using the logistic regression method, you can adjust the imputed classification levels by modifying the log odds ratios for the classification levels (Carpenter and Kenward 2013, pp. 240–241; Van Buuren 2012, pp. 88–89). By modifying the log odds ratios, you modify the predicted probabilities for the classification levels.

For each imputed variable, you can use the ADJUST option to do the following:

- specify a subset of observations for which imputed values are adjusted. Otherwise, all imputed values are adjusted.
- adjust imputed continuous variable values by using the SHIFT=, SCALE=, and SIGMA= options. These options add a constant, multiply by a constant factor, and add a simulated value to the imputed values, respectively.
- adjust imputed classification variable levels by adjusting predicted probabilities for the classification levels by using the SHIFT= and SIGMA= options. These options add a constant and add a simulated constant value, respectively, to the log odds ratios for the classification levels.

In addition, you can provide the shift and scale parameters for each imputation by using a PARMS= data set.

When you use the MNAR statement together with a MONOTONE statement, the variables are imputed sequentially. For each imputed variable, the values can be adjusted using the ADJUST option, and these adjusted values are used to impute values for subsequent variables.

When you use the MNAR statement together with an FCS statement, there are two phases in each imputation: the preliminary filled-in phase, followed by the imputation phase. For each imputed variable, the values can be adjusted using the ADJUST option in the imputation phase in each of the imputations. These adjusted values are used to impute values for other variables in the imputation phase.

For illustrations of adjusting imputed continuous values, adjusting log odds ratio for imputed classification levels, and adjusting imputed continuous values by using parameters that are stored in an input data set, see Example 77.16, Example 77.17, and Example 77.18, respectively.

Specifying the Imputed Values to Be Adjusted

By default, all available imputed values are adjusted. You can specify a subset of imputed values to be adjusted by using the ADJUSTOBS= suboption in the ADJUST option.

You can specify a classification variable to identify the subset of imputed values to be adjusted by using the ADJUSTOBS= (*obs-variable*= '*level1*' < '*level2*' **...** >) option. This subset consists of the imputed values in the set of observations for which *obs-variable* equals one of the specified levels.

Adjusting Imputed Continuous Variables

For an imputed continuous variable, the SCALE=*c* option specifies the scale parameter, $c > 0$, for imputed values; the SHIFT=δ option specifies the shift parameter, δ, for imputed values; and the SIGMA=σ option specifies the sigma parameter, $\sigma > 0$, for imputed values.

When the sigma parameter is not specified, the adjusted value for each imputed value y is given by

$$y^* = c\, y + \delta$$

where c is the scale parameter and δ is the shift parameter.

When you specify a sigma parameter σ, a simulated shift parameter is generated from the normal distribution that has mean δ and standard deviation σ in each imputation

$$\delta^* \sim N\left(\delta,\ \sigma^2\right)$$

The adjusted value is then given by

$$y^* = c\, y + \delta^*$$

Adjusting Imputed Classification Variables

For an imputed classification variable, you can specify adjustment parameters for the response level. The SHIFT=δ option specifies the shift parameter δ, the SIGMA=σ option specifies the sigma parameter $\sigma > 0$, and the EVENT='*level*' option identifies the response level.

When the sigma parameter is not specified, the shift parameter δ is used in all imputations. When you specify a sigma parameter σ, a simulated shift parameter is generated from the normal distribution that has mean δ and standard deviation σ for each imputation

$$\delta^* \sim N\left(\delta,\ \sigma^2\right)$$

The next three sections provide details for adjusting imputed binary, ordinal, and nominal response variables.

Adjusting Imputed Binary Response Variables

For an imputed binary classification variable Y, the shift parameter δ is applied to the logit function values for the corresponding response level.

For instance, if Y has binary responses 1 and 2, a simulated logit model

$$\text{logit}(\,\text{pr}(Y = 1 \mid \mathbf{x})\,) = \alpha + \mathbf{x}'\boldsymbol{\beta}$$

is used to impute the missing response values. For a detailed description of this simulated logit model, see the section "Binary Response Logistic Regression" on page 6104.

For an observation that has missing Y and covariates $\mathbf{x_0}$, the predicted probabilities that Y=1 and Y=2 are then given by

$$\text{pr}(Y = 1) = \frac{e^{\alpha + \mathbf{x_0}'\boldsymbol{\beta}}}{e^{\alpha + \mathbf{x_0}'\boldsymbol{\beta}} + 1} = \frac{e^{d_1}}{e^{d_1} + e^{d_2}}$$

$$\text{pr}(Y = 2) = \frac{1}{e^{\alpha + \mathbf{x_0}'\boldsymbol{\beta}} + 1} = \frac{e^{d_2}}{e^{d_1} + e^{d_2}}$$

where $d_1 = \alpha + \mathbf{x_0}'\boldsymbol{\beta}$ and $d_2 = 0$.

When you provide the shift parameters δ_1 for the response Y=1 and δ_2 for the response Y=2, the predicted probabilities are

$$\text{pr}(Y = 1) = \frac{e^{d_1^*}}{e^{d_1^*} + e^{d_2^*}}$$

$$\text{pr}(Y = 2) = \frac{e^{d_2^*}}{e^{d_1^*} + e^{d_2^*}}$$

where $d_1^* = d_1 + \delta_1$ and $d_2^* = d_2 + \delta_2 = \delta_2$.

For example, the following statement specifies the shift parameters $\delta_1 = 0.8$ and $\delta_2 = 1.6$:

```
mnar adjust( y(event='1') / shift=0.8)
     adjust( y(event='2') / shift=1.6);
```

The statement

```
mnar adjust( y(event='1') / shift=0.8 sigma=0.2);
```

simulates a shift parameter δ_1 from

$$\delta \sim N\left(0.8,\ 0.2^2\right)$$

in each imputation. Because an adjustment is not specified for Y=2, the corresponding shift parameter is $\delta_2 = 0$.

Adjusting Imputed Ordinal Response Variables

For an imputed ordinal classification variable Y, the shift parameter δ is applied to the cumulative logit function values for the corresponding response level.

For instance, if Y has ordinal responses 1, 2, ..., K, a simulated cumulative logit model that has covariates $\mathbf{x}$,

$$\text{logit}(\,\text{pr}(Y \le k \mid \mathbf{x})\,) = \alpha_k + \mathbf{x}'\boldsymbol{\beta}$$

is used to impute the missing response values, where k = 1, 2, ..., K–1. For a detailed description of this model, see the section "Ordinal Response Logistic Regression" on page 6105.

For an observation that has missing Y and covariates $\mathbf{x_0}$, the predicted cumulative probability for $Y \le j$, j = 1, 2, ..., K–1, is then given by

$$\text{pr}(Y \le j) = \frac{e^{\alpha_j + \mathbf{x_0}'\boldsymbol{\beta}}}{e^{\alpha_j + \mathbf{x_0}'\boldsymbol{\beta}} + 1} = \frac{e^{d_j}}{e^{d_j} + e^{d_K}}$$

where $d_j = \alpha_j + \mathbf{x_0}'\boldsymbol{\beta}$ and $d_K = 0$.

The predicted probabilities for $Y = k$ are

$$\text{pr}(Y = k) = \begin{cases} \frac{e^{d_1}}{e^{d_1} + e^{d_K}} & \text{if } k = 1 \\ \frac{e^{d_k}}{e^{d_k} + e^{d_K}} - \frac{e^{d_{(k-1)}}}{e^{d_{(k-1)}} + e^{d_K}} & \text{if } 1 < k < K \\ \frac{e^{d_K}}{e^{d_{(K-1)}} + e^{d_K}} & \text{if } k = K \end{cases}$$

For an ordinal logistic regression method that has two response levels, the section "Adjusting Imputed Binary Response Variables" on page 6132 explains how the predicted probabilities are adjusted using shift parameters.

For an ordinal logistic regression method that has more than two response levels, only one classification level can be adjusted. When you provide the shift parameter δ for the response level $Y = k$, the predicted probability for $Y = k$ is then given by

$$\mathrm{pr}(Y=k) = \begin{cases} \frac{e^{d_1^*}}{e^{d_1^*}+e^{d_K}} & \text{if } k = 1 \\ \frac{e^{d_k^*}}{e^{d_k^*}+e^{d_K}} - \frac{e^{d_{(k-1)}}}{e^{d_{(k-1)}}+e^{d_K}} & \text{if } 1 < k < K \\ \frac{e^{d_K^*}}{e^{d_{(K-1)}}+e^{d_K^*}} & \text{if } k = K \end{cases}$$

where $d_k^* = d_k + \delta$.

The predicted probabilities for the remaining $Y \neq k$ are then adjusted proportionally. When the shift parameter δ is less than 0, the value d_k^* can be less than d_{k-1} for $1 < k < K$. In this case, $\mathrm{pr}(Y = k)$ is set to 0.

Adjusting Imputed Nominal Response Variables

For an imputed nominal classification variable Y, the shift parameter δ is applied to the generalized logit model function values for the corresponding response level.

For instance, if
VariableY has nominal responses 1, 2, ..., K, a simulated generalized logit model

$$\log\left(\frac{\mathrm{pr}(Y=k \mid \mathbf{x})}{\mathrm{pr}(Y=K \mid \mathbf{x})}\right) = \alpha_k + \mathbf{x}'\boldsymbol{\beta}_k$$

is used to impute the missing response values, where k=1, 2, ..., K–1. For a detailed description of this model, see the section "Nominal Response Logistic Regression" on page 6106.

For an observation with missing Y and covariates $\mathbf{x_0}$, the predicted probability for Y = j, $j < K$, is then given by

$$\mathrm{pr}(Y=j) = \frac{e^{\alpha_j + \mathbf{x_0}'\boldsymbol{\beta}_j}}{\sum_{k=1}^{K-1} e^{\alpha_k + \mathbf{x_0}'\boldsymbol{\beta}_k} + 1} = \frac{e^{d_j}}{\sum_{k=1}^{K} e^{d_k}}$$

and

$$\mathrm{pr}(Y=K) = \frac{1}{\sum_{k=1}^{K-1} e^{\alpha_k + \mathbf{x_0}'\boldsymbol{\beta}_k} + 1} = \frac{e^{d_K}}{\sum_{k=1}^{K} e^{d_k}}$$

where $d_k = \alpha_k + \mathbf{x}'\boldsymbol{\beta}_k$ for $k < K$ and $d_K = 0$.

When you use the shift parameters δ_k for $Y = k, k = 1, 2, \ldots, K$, the predicted probabilities are

$$\mathrm{pr}(Y=j) = \frac{e^{d_j^*}}{\sum_{k=1}^{K} e^{d_k^*}}$$

where $d_k^* = d_k + \delta_k$.

Summary of Issues in Multiple Imputation

This section summarizes issues that are encountered in applications of the MI procedure.

The MAR Assumption

Multiple imputation usually assumes that the data are missing at random (MAR). But the assumption cannot be verified, because the missing values are not observed. Although the MAR assumption becomes more plausible as more variables are included in the imputation model (Schafer 1997, pp. 27–28; Van Buuren, Boshuizen, and Knook 1999, p. 687), it is important to examine the sensitivity of inferences to departures from the MAR assumption.

Number of Imputations

Based on estimator efficiency, only a small number of imputations are needed for data that have modest missing information (Rubin 1987, p. 114). However, based on other inferential aspects, such as confidence intervals and *p*-values, a larger number of imputations are needed for reliable results (Allison 2012; Van Buuren 2012, pp. 49–50). You can informally verify the number of imputations by replicating sets of *m* imputations and checking whether the estimates are stable (Horton and Lipsitz 2001, p. 246).

Imputation Model

Generally you should include as many variables as you can in the imputation model (Rubin 1996), At the same time, however, it is important to keep the number of variables in control, as discussed by Barnard and Meng (1999, pp. 19–20). For the imputation of a particular variable, the model should include variables in the complete-data model, variables that are correlated with the imputed variable, and variables that are associated with the missingness of the imputed variable Schafer 1997, p. 143; Van Buuren, Boshuizen, and Knook 1999, p. 687).

Multivariate Normality Assumption

Although the regression and MCMC methods assume multivariate normality, inferences based on multiple imputation can be robust to departures from the multivariate normality if the amount of missing information is not large (Schafer 1997, pp. 147–148).

You can use variable transformations to make the normality assumption more tenable. Variables are transformed before the imputation process and then back-transformed to create imputed values.

Monotone Regression Method

With the multivariate normality assumption, either the regression method or the predictive mean matching method can be used to impute continuous variables in data sets with monotone missing patterns.

The predictive mean matching method ensures that imputed values are plausible and might be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Monotone Propensity Score Method

The propensity score method can also be used to impute continuous variables in data sets with monotone missing patterns.

The propensity score method does not use correlations among variables and is not appropriate for analyses involving relationship among variables, such as a regression analysis (Schafer 1999, p. 11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

MCMC Monotone-Data Imputation

The MCMC method is used to impute continuous variables in data sets with arbitrary missing patterns, assuming a multivariate normal distribution for the data. It can also be used to impute just enough missing values to make the imputed data sets have a monotone missing pattern. Then, a more flexible monotone imputation method can be used for the remaining missing values.

Checking Convergence in MCMC

In an MCMC method, parameters are drawn after the MCMC is run long enough to converge to its stationary distribution. In practice, however, it is not simple to verify the convergence of the process, especially for a large number of parameters.

You can check for convergence by examining the observed-data likelihood ratio statistic and worst linear function of the parameters in each iteration. You can also check for convergence by examining a plot of autocorrelation function, as well as a trace plot of parameters (Schafer 1997, p. 120).

EM Estimates

The EM algorithm can be used to compute the MLE of the mean vector and covariance matrix of the data with missing values, assuming a multivariate normal distribution for the data. However, the covariance matrix associated with the estimate of the mean vector cannot be derived from the EM algorithm.

In the MI procedure, you can use the EM algorithm to compute the posterior mode, which provides a good starting value for the MCMC method (Schafer 1997, p. 169).

Sensitivity Analysis

Multiple imputation inference often assumes that the data are missing at random (MAR). But the MAR assumption cannot be verified, because the missing values are not observed. For a study that assumes MAR, the sensitivity of inferences to departures from the MAR assumption should be examined.

In the MI procedure, you can use the MNAR statement to imputes missing values for scenarios under the MNAR assumption. You can then generate inferences and examine the results. If the results under MNAR differ from the results under MAR, then the conclusion under MAR is in question.

Plot Options Superseded by ODS Graphics

You can select one of two types of graphics in PROC MI: ODS and traditional. When ODS Graphics is enabled, you can use the PLOTS= option in the MCMC statement to create plots by using ODS Graphics. When ODS Graphics is not enabled, you can use the TIMEPLOT and ACFPLOT options in the MCMC statement to create traditional graphics. ODS Graphics is the preferred method of creating graphs, superseding

traditional graphics. For more information about ODS Graphics options see the PLOTS= option in the section "MCMC Statement" on page 6082. This section describes the options that are available in the MCMC statement for traditional graphics.

Table 77.8 summarizes the options available for traditional graphics in the MCMC statement.

Table 77.8 Traditional Graphics Options in the MCMC Statement

Option	Description
TIMEPLOT	Displays trace plots
ACFPLOT	Displays autocorrelation plots
GOUT=	Specifies the graphics catalog name for saving traditional graphics output

The following options are available in the MCMC statement for traditional graphics (in alphabetical order).

ACFPLOT < **(***options*< */ display-options* >**)** >
displays the traditional autocorrelation function plots of parameters from iterations. The ACFPLOT option is applicable only if ODS Graphics is not enabled.

The available options are as follows.

COV < **(** < *variables* > < *variable1*variable2* > < *. . . variable1*variable2* > **)** >
displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < **(** *variables* **)** >
displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF
displays the plot for the worst linear function.

When the ACFPLOT is specified without the preceding options, the procedure displays plots of means for all variables that are used.

The display options provide additional information for the autocorrelation function plots. By default, the MI procedure uses the star (*) as the plot symbol to display the points with a height of one (percentage screen unit) in the plot, a solid line to display the reference line of zero autocorrelation, vertical line segments to connect autocorrelations to the reference line, and a pair of dashed lines to display approximately 95% confidence limits for the autocorrelations.

You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol, respectively, and the CNEEDLES= and WNEEDLES= options to change the color and width of the needles, respectively. You can also use the LREF=, CREF=, and WREF= options to change the line type, color, and width of the reference line, respectively. Similarly, you can use the LCONF=, CCONF=, and WCONF= options to change the line type, color, and width of the confidence limits, respectively.

By default, the plot title "Autocorrelation Plot" is displayed in a autocorrelation function plot. You can request another title by using the TITLE= option within the ACFPLOT option. When another title is

also specified in a TITLE statement, this title is displayed as the main title and the plot title is displayed as a subtitle in the plot.

You can use options in the GOPTIONS statement to change the color and height of the title. See the chapter "The SAS/GRAPH Statements" in *SAS/GRAPH: Help* for a description of title options.

The available display options are as follows:

CCONF=*color*
specifies the color of the displayed confidence limits. The default is CCONF=BLACK.

CFRAME=*color*
specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled.

CNEEDLES=*color*
specifies the color of the vertical line segments (needles) that connect autocorrelations to the reference line. The default is CNEEDLES=BLACK.

CREF=*color*
specifies the color of the displayed reference line. The default is CREF=BLACK.

CSYMBOL=*color*
specifies the color of the displayed data points. The default is CSYMBOL=BLACK.

HSYMBOL=*number*
specifies the height of data points in percentage screen units. The default is HSYMBOL=1.

LCONF=*linetype*
specifies the line type for the displayed confidence limits. The default is LCONF=1, a solid line.

LOG
requests that the logarithmic transformations of parameters be used to compute the autocorrelations; it is generally used for the variances of variables. When a parameter has values less than or equal to zero, the corresponding plot is not created.

LREF=*linetype*
specifies the line type for the displayed reference line. The default is LREF=3, a dashed line.

NAME=*'string'*
specifies a descriptive name, up to eight characters, that appears in the name field of the PROC GREPLAY master menu. The default is NAME='MI'.

NLAG=*number*
specifies the maximum lag of the series. The default is NLAG=20. The autocorrelations at each lag are displayed in the graph.

SYMBOL=*value*
specifies the symbol for data points in percentage screen units. The default is SYMBOL=STAR.

TITLE=*'string'*
specifies the title to be displayed in the autocorrelation function plots. The default is TITLE='Autocorrelation Plot'.

WCONF=*number*
specifies the width of the displayed confidence limits in percentage screen units. If you specify the WCONF=0 option, the confidence limits are not displayed. The default is WCONF=1.

WNEEDLES=*number*
specifies the width of the displayed needles that connect autocorrelations to the reference line, in percentage screen units. If you specify the WNEEDLES=0 option, the needles are not displayed. The default is WNEEDLES=1.

WREF=*number*
specifies the width of the displayed reference line in percentage screen units. If you specify the WREF=0 option, the reference line is not displayed. The default is WREF=1.

For example, the following statement requests autocorrelation function plots for the means and variances of the variable `y1`, respectively:

```
acfplot( mean( y1) cov(y1) /log);
```

Logarithmic transformations of both the means and variances are used in the plots. For a detailed description of the autocorrelation function plot, see the section "Autocorrelation Function Plot" on page 6119; see also Schafer (1997, pp. 120–126) and the *SAS/ETS User's Guide*.

GOUT=*graphics-catalog*
specifies the graphics catalog for saving graphics output from PROC MI. The default is WORK.GSEG. For more information, see "The GREPLAY Procedure" in *SAS/GRAPH: Help*.

TIMEPLOT < *(options* < */ display-options* > *)* >
displays the traditional trace (time series) plots of parameters from iterations. The TIMEPLOT option is applicable only if ODS Graphics is not enabled.

The available options are as follows:

COV <(< *variables* > < *variable1*variable2* > . . . **) >**

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < **(***variables***)** >
displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF
displays the plot of the worst linear function.

When the TIMEPLOT is specified without the preceding options, the procedure displays plots of means for all variables that are used.

The display options provide additional information for the trace plots. By default, the MI procedure uses solid line segments to connect data points in a trace plot. You can use the CCONNECT=, LCONNECT=, and WCONNECT= options to change the color, line type, and width of the line segments, respectively. When WCONNECT=0 is specified, the data points are not connected, and the procedure uses the plus sign (+) as the plot symbol to display the points with a height of one (percentage screen unit) in a trace plot. You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol, respectively.

By default, the plot title "Trace Plot" is displayed in a trace plot. You can request another title by using the TITLE= option in the TIMEPLOT option. When another title is also specified in a TITLE statement, this title is displayed as the main title and the plot title is displayed as a subtitle in the plot.

You can use options in the GOPTIONS statement to change the color and height of the title. See the chapter "The SAS/GRAPH Statements" in *SAS/GRAPH: Help* for an illustration of title options.

The available display options are as follows:

CCONNECT=*color*
specifies the color of the line segments that connect data points in the trace plots. The default is CCONNECT=BLACK.

CFRAME=*color*
specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled.

CSYMBOL=*color*
specifies the color of the data points to be displayed in the trace plots. The default is CSYMBOL=BLACK.

HSYMBOL=*number*
specifies the height of data points in percentage screen units. The default is HSYMBOL=1.

LCONNECT=*linetype*
specifies the line type for the line segments that connect data points in the trace plots. The default is LCONNECT=1, a solid line.

LOG
requests that the logarithmic transformations of parameters be used; it is generally used for the variances of variables. When a parameter value is less than or equal to zero, the value is not displayed in the corresponding plot.

NAME=*'string'*
specifies a descriptive name, up to eight characters, that appears in the name field of the PROC GREPLAY master menu. The default is NAME='MI'.

SYMBOL=*value*
specifies the symbol for data points in percentage screen units. The default is SYMBOL=PLUS.

TITLE=*'string'*
specifies the title to be displayed in the trace plots. The default is TITLE='Trace Plot'.

WCONNECT=*number*
specifies the width of the line segments that connect data points in the trace plots, in percentage screen units. If you specify the WCONNECT=0 option, the data points are not connected. The default is WCONNECT=1.

For a detailed description of the trace plot, see the section "Trace Plot" on page 6119 and Schafer (1997, pp. 120–126).

ODS Table Names

PROC MI 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 77.9. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 77.9 ODS Tables Produced by PROC MI

ODS Table Name	Description	Statement	Option
Corr	Pairwise correlations		SIMPLE
EMEstimates	EM (MLE) estimates	EM	
EMInitEstimates	EM initial estimates	EM	
EMIterHistory	EM (MLE) iteration history	EM	ITPRINT
EMPostEstimates	EM (posterior mode) estimates	MCMC	INITIAL=EM
EMPostIterHistory	EM (posterior mode) iteration history	MCMC	INITIAL=EM (ITPRINT)
EMWLF	Worst linear function	MCMC	WLF
FCSDiscrim	Discriminant model group means	FCS	DISCRIM (/DETAILS)
FCSLogistic	Logistic model	FCS	LOGISTIC (/DETAILS)
FCSModel	FCS models	FCS	
FCSReg	Regression model	FCS	REG (/DETAILS)
FCSRegPMM	Predicted mean matching model	FCS	REGPMM (/DETAILS)
MCMCInitEstimates	MCMC initial estimates	MCMC	DISPLAYINIT
MissPattern	Missing data patterns		
MNARModel	Observations that are used for imputation model under MNAR	MNAR	MODEL
MNARAdjust	Adjustment parameters and imputed values to be adjusted under MNAR	MNAR	ADJUST
ModelInfo	Model information		
MonoDiscrim	Discriminant model group means	MONOTONE	DISCRIM (/DETAILS)
MonoLogistic	Logistic model	MONOTONE	LOGISTIC (/DETAILS)
MonoModel	Monotone models	MONOTONE	
MonoPropensity	Propensity score model	MONOTONE	PROPENSITY (/DETAILS)

Table 77.9 *continued*

ODS Table Name	Description	Statement	Option
	logistic function		
MonoReg	Regression model	MONOTONE	REG (/DETAILS)
MonoRegPMM	Predicted mean matching model	MONOTONE	REGPMM (/DETAILS)
ParameterEstimates	Parameter estimates		
Transform	Variable transformations	TRANSFORM	
Univariate	Univariate statistics		SIMPLE
VarianceInfo	Between, within, and total variances		

ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS."

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 615 in Chapter 21, "Statistical Graphics Using ODS."

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section "A Primer on ODS Statistical Graphics" on page 614 in Chapter 21, "Statistical Graphics Using ODS."

PROC MI assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. To request these graphs, ODS Graphics must be enabled and you must specify the options indicated in Table 77.10.

Table 77.10 Graphs Produced by PROC MI

ODS Graph Name	Description	Statement	Option
ACFPlot	ACF plot	MCMC	PLOTS=ACF
TracePlot	Trace plot	MCMC	PLOTS= TRACE
		FCS	PLOTS= TRACE

Examples: MI Procedure

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. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail (Length1), from the nose to the notch of its tail (Length2), and from the nose to the end of its tail (Length3). See Chapter 111, "The STEPDISC Procedure," for more information.

The Fish1 data set is constructed from the Fish data set and contains only one species of the fish and the three length measurements. Some values have been set to missing, and the resulting data set has a monotone missing pattern in the variables Length1, Length2, and Length3. The Fish1 data set is used in Example 77.2 with the propensity score method and in Example 77.3 with the regression method.

The Fish2 data set is also 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 Fish2 data set is used in Example 77.4 with the logistic regression method and in Example 77.5 with the discriminant function method. Note that some values of the variable Species have also been altered in the data set.

The Fish3 data set is similar to the data set Fish2 except some additional values have been set to missing and the resulting data set has an arbitrary missing pattern. The Fish3 data set is used in Example 77.7 and in Example 77.8.

The Fitness1 data set created in the section "Getting Started: MI Procedure" on page 6067 is used in other examples.

The following statements create the Fish1 data set:

```
*----------------------------Fish1 Data-----------------------------*
| The data set contains one species of the fish (Bream) and          |
| three measurements: Length1, Length2, Length3.                    |
| Some values have been set to missing, and the resulting data set   |
| has a monotone missing pattern in the variables                    |
| Length1, Length2, and Length3.                                     |
*--------------------------------------------------------------------*;
data Fish1;
   title 'Fish Measurement Data';
   input Length1 Length2 Length3 @@;
   datalines;
23.2 25.4 30.0    24.0 26.3 31.2    23.9 26.5 31.1
26.3 29.0 33.5    26.5 29.0   .     26.8 29.7 34.7
26.8   .    .     27.6 30.0 35.0    27.6 30.0 35.1
28.5 30.7 36.2    28.4 31.0 36.2    28.7   .    .
29.1 31.5   .     29.5 32.0 37.3    29.4 32.0 37.2
29.4 32.0 37.2    30.4 33.0 38.3    30.4 33.0 38.5
30.9 33.5 38.6    31.0 33.5 38.7    31.3 34.0 39.5
31.4 34.0 39.2    31.5 34.5   .     31.8 35.0 40.6
31.9 35.0 40.5    31.8 35.0 40.9    32.0 35.0 40.6
32.7 36.0 41.5    32.8 36.0 41.6    33.5 37.0 42.6
35.0 38.5 44.1    35.0 38.5 44.0    36.2 39.5 45.3
37.4 41.0 45.9    38.0 41.0 46.5
;
```

The Fish2 data set contains two of the seven species in the Fish data set. For each of the two species (Bream and Pike), the length from the nose of the fish to the end of its tail and the width of each fish are measured.

The following statements create the Fish2 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 create the Fish3 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
;
```

Example 77.1: EM Algorithm for MLE

This example uses the EM algorithm to compute the maximum likelihood estimates for parameters of multivariate normally distributed data with missing values. The following statements invoke the MI procedure and request the EM algorithm to compute the MLE for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of a multivariate normal distribution from the input data set Fitness1:

```
proc mi data=Fitness1 seed=1518971 simple nimpute=0;
   em itprint outem=outem;
   var Oxygen RunTime RunPulse;
run;
```

Note that when you specify the NIMPUTE=0 option, the missing values are not imputed.

The "Model Information" table in Output 77.1.1 describes the method and options used in the procedure if a positive number is specified in the NIMPUTE= option.

Output 77.1.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	0
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	1518971

The "Missing Data Patterns" table in Output 77.1.2 lists distinct missing data patterns with corresponding frequencies and percentages. Here, a value of "X" means that the variable is observed in the corresponding group and a value of "." means that the variable is missing. The table also displays group-specific variable means.

Output 77.1.2 Missing Data Patterns

Missing Data Patterns

						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	21	67.74	46.353810	10.809524	171.666667
2	X	X	.	4	12.90	47.109500	10.137500	.
3	X	.	.	3	9.68	52.461667	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	.	2	6.45	.	9.885000	.

With the SIMPLE option, the procedure displays simple descriptive univariate statistics for available cases in the "Univariate Statistics" table in Output 77.1.3 and correlations from pairwise available cases in the "Pairwise Correlations" table in Output 77.1.4.

Output 77.1.3 Univariate Statistics

Univariate Statistics

						Missing Values	
Variable	**N**	**Mean**	**Std Dev**	**Minimum**	**Maximum**	**Count**	**Percent**
Oxygen	28	47.11618	5.41305	37.38800	60.05500	3	9.68
RunTime	28	10.68821	1.37988	8.63000	14.03000	3	9.68
RunPulse	22	171.86364	10.14324	148.00000	186.00000	9	29.03

Output 77.1.4 Pairwise Correlations

Pairwise Correlations			
	Oxygen	**RunTime**	**RunPulse**
Oxygen	1.000000000	-0.849118562	-0.343961742
RunTime	-0.849118562	1.000000000	0.247258191
RunPulse	-0.343961742	0.247258191	1.000000000

When you use the EM statement, the MI procedure displays the initial parameter estimates for the EM algorithm in the "Initial Parameter Estimates for EM" table in Output 77.1.5.

Output 77.1.5 Initial Parameter Estimates for EM

Initial Parameter Estimates for EM				
TYPE	**_NAME_**	**Oxygen**	**RunTime**	**RunPulse**
MEAN		47.116179	10.688214	171.863636
COV	Oxygen	29.301078	0	0
COV	RunTime	0	1.904067	0
COV	RunPulse	0	0	102.885281

When you use the ITPRINT option in the EM statement, the "EM (MLE) Iteration History" table in Output 77.1.6 displays the iteration history for the EM algorithm.

Output 77.1.6 EM (MLE) Iteration History

EM (MLE) Iteration History				
Iteration	**-2 Log L**	**Oxygen**	**RunTime**	**RunPulse**
0	289.544782	47.116179	10.688214	171.863636
1	263.549489	47.116179	10.688214	171.863636
2	255.851312	47.139089	10.603506	171.538203
3	254.616428	47.122353	10.571685	171.426790
4	254.494971	47.111080	10.560585	171.398296
5	254.483973	47.106523	10.556768	171.389208
6	254.482920	47.104899	10.555485	171.385257
7	254.482813	47.104348	10.555062	171.383345
8	254.482801	47.104165	10.554923	171.382424
9	254.482800	47.104105	10.554878	171.381992
10	254.482800	47.104086	10.554864	171.381796
11	254.482800	47.104079	10.554859	171.381708
12	254.482800	47.104077	10.554858	171.381669

The "EM (MLE) Parameter Estimates" table in Output 77.1.7 displays the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of a multivariate normal distribution from the data set Fitness1.

Output 77.1.7 EM (MLE) Parameter Estimates

EM (MLE) Parameter Estimates				
TYPE	_NAME_	Oxygen	RunTime	RunPulse
MEAN		47.104077	10.554858	171.381669
COV	Oxygen	27.797931	-6.457975	-18.031298
COV	RunTime	-6.457975	2.015514	3.516287
COV	RunPulse	-18.031298	3.516287	97.766857

You can also output the EM (MLE) parameter estimates to an output data set with the OUTEM= option. The following statements list the observations in the output data set Outem:

```
proc print data=outem;
   title 'EM Estimates';
run;
```

The output data set Outem in Output 77.1.8 is a TYPE=COV data set. The observation with _TYPE_='MEAN' contains the MLE for the parameter μ, and the observations with _TYPE_='COV' contain the MLE for the parameter Σ of a multivariate normal distribution from the data set Fitness1.

Output 77.1.8 EM Estimates

EM Estimates

Obs	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	MEAN		47.1041	10.5549	171.382
2	COV	Oxygen	27.7979	-6.4580	-18.031
3	COV	RunTime	-6.4580	2.0155	3.516
4	COV	RunPulse	-18.0313	3.5163	97.767

Example 77.2: Monotone Propensity Score Method

This example uses the propensity score method to impute missing values for variables in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the propensity score method. The resulting data set is named Outex2.

```
proc mi data=Fish1 seed=899603 out=outex2;
   monotone propensity;
   var Length1 Length2 Length3;
run;
```

Note that the VAR statement is required and the data set must have a monotone missing pattern with variables as ordered in the VAR statement.

The "Model Information" table in Output 77.2.1 describes the method and options used in the multiple imputation process. By default, 25 imputations are created for the missing data.

Output 77.2.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FISH1
Method	Monotone
Number of Imputations	25
Seed for random number generator	899603

When monotone methods are used in the imputation, MONOTONE is displayed as the method. The "Monotone Model Specification" table in Output 77.2.2 displays the detailed model specification. By default, the observations are sorted into five groups based on their propensity scores.

Output 77.2.2 Monotone Model Specification

Monotone Model Specification	
Method	**Imputed Variables**
Propensity(Groups= 5)	Length2 Length3

Without covariates specified for imputed variables Length2 and Length3, the variable Length1 is used as the covariate for Length2, and the variables Length1 and Length2 are used as covariates for Length3.

The "Missing Data Patterns" table in Output 77.2.3 lists distinct missing data patterns with corresponding frequencies and percentages. Here, values of "X" and "." indicate that the variable is observed or missing, respectively, in the corresponding group. The table confirms a monotone missing pattern for these three variables.

Output 77.2.3 Missing Data Patterns

Missing Data Patterns								
						Group Means		
Group	**Length1**	**Length2**	**Length3**	**Freq**	**Percent**	**Length1**	**Length2**	**Length3**
1	X	X	X	30	85.71	30.603333	33.436667	38.720000
2	X	X	.	3	8.57	29.033333	31.666667	.
3	X	.	.	2	5.71	27.750000	.	.

For the imputation process, first, missing values of Length2 in group 3 are imputed using observed values of Length1. Then the missing values of Length3 in group 2 are imputed using observed values of Length1 and Length2. And finally, the missing values of Length3 in group 3 are imputed using observed values of Length1 and imputed values of Length2.

After the completion of m imputations, the "Variance Information" table in Output 77.2.4 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. A detailed description of these statistics is provided in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Output 77.2.4 Variance Information

Variance Information (25 Imputations)

Variable	Variance Between	Variance Within	Variance Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Length2	0.003228	0.457211	0.460568	31.925	0.007343	0.007294	0.999708
Length3	0.037617	0.542147	0.581269	29.829	0.072161	0.067656	0.997301

The "Parameter Estimates" table in Output 77.2.5 displays the estimated mean and standard error of the mean for each variable. The inferences are based on the *t* distributions. For each variable, the table also displays a 95% mean confidence interval and a *t* statistic with the associated *p*-value for the hypothesis that the population mean is equal to the value specified in the MU0= option, which is 0 by default.

Output 77.2.5 Parameter Estimates

Parameter Estimates (25 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Length2	33.040343	0.678652	31.65785	34.42284	31.925	32.957143	33.122857	0	48.69	<.0001
Length3	38.325143	0.762410	36.76772	39.88257	29.829	37.808571	38.614286	0	50.27	<.0001

The following statements list the first 10 observations of the data set Outex2, as shown in Output 77.2.6. The missing values are imputed from observed values with similar propensity scores.

```
proc print data=outex2(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.2.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Length1	Length2	Length3
1	1	23.2	25.4	30.0
2	1	24.0	26.3	31.2
3	1	23.9	26.5	31.1
4	1	26.3	29.0	33.5
5	1	26.5	29.0	38.6
6	1	26.8	29.7	34.7
7	1	26.8	29.0	35.0
8	1	27.6	30.0	35.0
9	1	27.6	30.0	35.1
10	1	28.5	30.7	36.2

Example 77.3: Monotone Regression Method

This example uses the regression method to impute missing values for all variables in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the regression

method for the variable Length2 and the predictive mean matching method for variable Length3. The resulting data set is named Outex3.

```
proc mi data=Fish1 round=.1  mu0= 0 35 45
        seed=13951639 nimpute=8 out=outex3;
   monotone reg(Length2/ details)
            regpmm(Length3= Length1 Length2 Length1*Length2/ details);
   var Length1 Length2 Length3;
run;
```

The ROUND= option is used to round the imputed values to the same precision as observed values. The values specified with the ROUND= option are matched with the variables Length1, Length2, and Length3 in the order listed in the VAR statement. The MU0= option requests *t* tests for the hypotheses that the population means corresponding to the variables in the VAR statement are Length2=35 and Length3=45.

The "Missing Data Patterns" table lists distinct missing data patterns with corresponding frequencies and percentages. It is identical to the table in Output 77.2.3 in Example 77.2.

The "Monotone Model Specification" table in Output 77.3.1 displays the model specification.

Output 77.3.1 Monotone Model Specification

The MI Procedure

Monotone Model Specification

Method	Imputed Variables
Regression	Length2
Regression-PMM(K= 5)	Length3

When you use the DETAILS option, the parameters estimated from the observed data and the parameters used in each imputation are displayed in Output 77.3.2 and Output 77.3.3.

Output 77.3.2 Regression Model

Regression Models for Monotone Method

			Imputation							
Imputed Variable	**Effect**	**Obs-Data**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Length2	Intercept	-0.04249	-0.049184	-0.055470	-0.051346	-0.064193	-0.030719	-0.030694	-0.050964	-0.017976
Length2	Length1	0.98587	1.001934	0.995275	0.992294	0.983122	0.995883	0.989193	0.968480	0.977476

Output 77.3.3 Regression Predicted Mean Matching Model

Regression Models for Monotone Predicted Mean Matching Method

			Imputation							
Imputed Variable	**Effect**	**Obs Data**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Length3	Intercept	-0.01304	0.004134	-0.011417	-0.034177	-0.010532	0.004685	-0.013917	0.012658	-0.020144
Length3	Length1	-0.01332	0.025320	-0.037494	0.308765	0.156606	-0.147118	0.097745	-0.254054	0.086950
Length3	Length2	0.98918	0.955510	1.025741	0.673374	0.828384	1.146440	0.860564	1.226015	0.879854
Length3	Length1*Length2	-0.02521	-0.034964	-0.022017	-0.017919	-0.029335	-0.034671	-0.023384	-0.023829	-0.026785

After the completion of eight imputations (NIMPUTE=8), the "Variance Information" table in Output 77.3.4 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. These statistics are described in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Output 77.3.4 Variance Information

Variance Information (8 Imputations)

	Variance						
Variable	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Length2	0.000089650	0.439208	0.439309	32.155	0.000230	0.000230	0.999971
Length3	0.000433	0.487356	0.487842	32.13	0.000998	0.000998	0.999875

The "Parameter Estimates" table in Output 77.3.5 displays a 95% mean confidence interval and a *t* statistic with its associated *p*-value for each of the hypotheses requested with the MU0= option.

Output 77.3.5 Parameter Estimates

Parameter Estimates (8 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Length2	33.106071	0.662804	31.75624	34.45590	32.155	33.088571	33.117143	35.000000	-2.86	0.0074
Length3	38.416786	0.698457	36.99430	39.83927	32.13	38.382857	38.445714	45.000000	-9.43	<.0001

The following statements list the first 10 observations of the data set Outex3 in Output 77.3.6. Note that the imputed values of Length2 are rounded to the same precision as the observed values.

```
proc print data=outex3(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.3.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Length1	Length2	Length3
1	1	23.2	25.4	30.0
2	1	24.0	26.3	31.2
3	1	23.9	26.5	31.1
4	1	26.3	29.0	33.5
5	1	26.5	29.0	34.7
6	1	26.8	29.7	34.7
7	1	26.8	28.8	34.7
8	1	27.6	30.0	35.0
9	1	27.6	30.0	35.1
10	1	28.5	30.7	36.2

Example 77.4: Monotone Logistic Regression Method for CLASS Variables

This example uses logistic regression method to impute values for a binary variable in a data set with a monotone missing pattern.

In the following statements, the logistic regression method is used for the binary CLASS variable Species:

```
proc mi data=Fish2 seed=1305417 nimpute=15 out=outex4;
   class Species;
   monotone reg( Width/ details)
            logistic( Species= Length Width Length*Width/ details);
   var Length Width Species;
run;
```

The "Model Information" table in Output 77.4.1 describes the method and options used in the multiple imputation process.

Output 77.4.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FISH2
Method	Monotone
Number of Imputations	15
Seed for random number generator	1305417

The "Monotone Model Specification" table in Output 77.4.2 describes methods and imputed variables in the imputation model. The procedure uses the logistic regression method to impute the variable Species in the model. Missing values in other variables are not imputed.

Output 77.4.2 Monotone Model Specification

Monotone Model Specification	
Method	**Imputed Variables**
Regression	Width
Logistic Regression	Species

The "Missing Data Patterns" table in Output 77.4.3 lists distinct missing data patterns with corresponding frequencies and percentages. The table confirms a monotone missing pattern for these variables.

Output 77.4.3 Missing Data Patterns

Missing Data Patterns							
						Group Means	
Group	**Length**	**Width**	**Species**	**Freq**	**Percent**	**Length**	**Width**
1	X	X	X	49	73.13	28.595918	4.482518
2	X	X	.	9	13.43	27.533333	4.444844
3	X	.	.	9	13.43	28.633333	.

When you use the DETAILS option, parameters estimated from the observed data and the parameters used in each imputation are displayed in the "Logistic Models for Monotone Method" table in Output 77.4.4.

Output 77.4.4 Regression Model

Regression Models for Monotone Method

			Imputation								
Imputed Variable	**Effect**	**Obs-Data**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Width	Intercept	0.00284	-0.029987	0.049363	-0.015273	-0.064915	0.059375	0.018049	-0.028171	-0.016050	-0.012890
Width	Length	0.96212	0.981287	0.906104	0.962814	0.978103	0.952034	0.920482	0.908541	0.962650	0.949542

Regression Models for Monotone Method

		Imputation					
Imputed Variable	**Effect**	**10**	**11**	**12**	**13**	**14**	**15**
Width	Intercept	-0.056099	-0.013302	0.031642	0.051256	-0.032029	0.030396
Width	Length	0.962843	0.921873	0.947360	1.003013	0.950239	0.955749

Output 77.4.5 Logistic Regression Model

Logistic Models for Monotone Method

			Imputation							
Imputed Variable	**Effect**	**Obs-Data**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Species	Intercept	-3.93577	-5.016163	-3.422209	-4.706398	-2.049090	-4.568278	-4.336259	-4.250352	-2.843154
Species	Length	10.41940	16.262215	6.082966	9.832246	4.992717	11.886805	5.789312	7.662947	5.757570
Species	Width	-14.56630	-21.856472	-8.653119	-15.534802	-7.401465	-15.621272	-12.855797	-14.816308	-8.792538
Species	Length*Width	-0.48936	-0.208880	0.795883	-0.011135	-0.461227	0.080406	-2.586760	-2.604478	-0.317211

Logistic Models for Monotone Method

		Imputation						
Imputed Variable	**Effect**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Species	Intercept	-1.055153	-3.501466	-2.140199	-3.629155	-4.020008	-2.615227	-4.964532
Species	Length	0.572346	10.900700	10.743223	10.504352	12.346335	5.432583	11.926760
Species	Width	-1.775130	-15.547003	-12.353169	-14.555215	-16.481415	-11.694606	-18.401905
Species	Length*Width	0.027353	-0.809353	-0.060720	-0.544245	-0.507705	-2.484002	-2.094407

The following statements list the first 10 observations of the data set Outex4 in Output 77.4.6:

```
proc print data=outex4(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.4.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length	Width
1	1	Parkki	16.5	2.32650
2	1	Parkki	17.4	2.31420
3	1	Parkki	19.8	2.20482
4	1	Parkki	21.3	2.91810
5	1	Parkki	22.4	3.29280
6	1	Perch	23.2	3.29440
7	1	Parkki	23.2	3.41040
8	1	Parkki	24.1	3.15710
9	1	Perch	25.8	3.66360
10	1	Parkki	28.0	4.14400

Note that a missing value of the variable Species is not imputed if the corresponding covariates are missing and not imputed, as shown by observation 4 in the table.

Example 77.5: Monotone Discriminant Function Method for CLASS Variables

This example uses discriminant monotone methods to impute values of a CLASS variable from the observed observation values in a data set with a monotone missing pattern.

The following statements impute the continuous variables Height and Width with the regression method and the classification variable Species with the discriminant function method:

```
proc mi data=Fish2 seed=7545417 out=outex5;
   class Species;
   monotone discrim( Species= Length Width/ details);
   var Length Width Species;
run;
```

The "Model Information" table in Output 77.5.1 describes the method and options used in the multiple imputation process.

Output 77.5.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FISH2
Method	Monotone
Number of Imputations	25
Seed for random number generator	7545417

The "Monotone Model Specification" table in Output 77.5.2 describes methods and imputed variables in the imputation model. The procedure uses the regression method to impute the variables Height and Width, and uses the logistic regression method to impute the variable Species in the model.

Output 77.5.2 Monotone Model Specification

Monotone Model Specification	
Method	**Imputed Variables**
Regression	Width
Discriminant Function	Species

The "Missing Data Patterns" table in Output 77.5.3 lists distinct missing data patterns with corresponding frequencies and percentages. The table confirms a monotone missing pattern for these variables.

Output 77.5.3 Missing Data Patterns

Missing Data Patterns							
						Group Means	
Group	**Length**	**Width**	**Species**	**Freq**	**Percent**	**Length**	**Width**
1	X	X	X	49	73.13	28.595918	4.482518
2	X	X	.	9	13.43	27.533333	4.444844
3	X	.	.	9	13.43	28.633333	.

When you use the DETAILS option, the parameters estimated from the observed data and the parameters used in each imputation are displayed in Output 77.5.4.

Output 77.5.4 Discriminant Model

Group Means for Monotone Discriminant Method											
						Imputation					
Species	**Variable**	**Obs-Data**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Parkki	**Length**	-0.62249	-0.917467	-0.909076	-0.146825	-0.682080	-1.187056	-0.850499	-0.697413	-0.440116	-1.018781
Parkki	**Width**	-0.71787	-0.921200	-1.036075	-0.343058	-0.844507	-1.353333	-0.881096	-0.830821	-0.478435	-1.131564
Perch	**Length**	0.13937	0.042471	0.219096	0.079881	0.080076	0.190100	0.298206	-0.061338	0.145498	0.134497
Perch	**Width**	0.14408	0.047041	0.197736	0.082832	0.118336	0.193865	0.295603	-0.051199	0.147394	0.159755

Group Means for Monotone Discriminant Method											
						Imputation					
Species	**Variable**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Parkki	**Length**	-0.651330	-0.706491	-0.247232	-0.852547	-0.846080	-0.294569	-0.456750	-0.286640	-0.300657	-0.574438
Parkki	**Width**	-0.785122	-0.666500	-0.390243	-0.960644	-0.803615	-0.342767	-0.603262	-0.481397	-0.516386	-0.685527
Perch	**Length**	0.073197	-0.007668	0.402478	0.271589	0.166251	0.239710	0.135714	0.107864	0.138797	0.571539
Perch	**Width**	0.090466	0.069233	0.419146	0.239319	0.182764	0.216010	0.167861	0.085148	0.127604	0.572531

Group Means for Monotone Discriminant Method							
				Imputation			
Species	**Variable**	**20**	**21**	**22**	**23**	**24**	**25**
Parkki	**Length**	-0.165544	-0.123289	-0.557388	-0.667641	-0.332219	-0.242661
Parkki	**Width**	-0.239261	-0.159966	-0.650667	-0.815406	-0.449938	-0.246102
Perch	**Length**	0.045337	-0.085437	0.306678	-0.003056	0.115950	0.304774
Perch	**Width**	0.051245	-0.013271	0.294392	0.026570	0.164894	0.317910

The following statements list the first 10 observations of the data set Outex5 in Output 77.5.5. Note that all missing values of the variables Width and Species are imputed.

```
proc print data=outex5(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.5.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length	Width
1	1	Parkki	16.5	2.32650
2	1	Parkki	17.4	2.31420
3	1	Perch	19.8	3.03975
4	1	Parkki	21.3	2.91810
5	1	Parkki	22.4	3.29280
6	1	Perch	23.2	3.29440
7	1	Parkki	23.2	3.41040
8	1	Parkki	24.1	3.15710
9	1	Perch	25.8	3.66360
10	1	Parkki	28.0	4.14400

Example 77.6: FCS Methods for Continuous Variables

This example uses FCS regression methods to impute values for all continuous variables in a data set that has an arbitrary missing pattern.

The following statements invoke the MI procedure and impute missing values for the Fitness1 data set:

```
proc mi data=Fitness1 seed=1213 nimpute=pctmissing(min=5 max=20)
        mu0=50 10 180 out=outex6;
   fcs nbiter=20 reg(Oxygen/details);
   var Oxygen RunTime RunPulse;
run;
```

The NIMPUTE=PCTMISSING option uses the percentage of the incomplete cases as the number of imputations. The MIN=5 (which is the default) and MAX=20 options restrict the number of imputations to be in the range of 5 to 20. That is, 5 imputations are generated if the percentage of the incomplete cases is less than 5, and 20 imputations are generated if this percentage is greater than 20.

The FCS statement requests multivariate imputations by FCS methods, and the NBITER=20 option (which is the default) specifies the number of burn-in iterations before each imputation.

The "Model Information" table in Output 77.6.1 describes the method and options used in the multiple imputation process.

Output 77.6.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	FCS
Number of Imputations	20
Number of Burn-in Iterations	20
Seed for random number generator	1213

The "FCS Model Specification" table in Output 77.6.2 describes methods and imputed variables in the imputation model. With the REG(OXYGEN) option in the FCS statement, the procedure uses the regression method to impute variable Oxygen. By default, the regression method is also used to impute variables RunTime and RunPulse.

Output 77.6.2 FCS Model Specification

FCS Model Specification	
Method	**Imputed Variables**
Regression	Oxygen RunTime RunPulse

The "Missing Data Patterns" table in Output 77.6.3 lists distinct missing data patterns with corresponding frequencies and percentages.

Output 77.6.3 Missing Data Patterns

Missing Data Patterns								
						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	21	67.74	46.353810	10.809524	171.666667
2	X	X	.	4	12.90	47.109500	10.137500	.
3	X	.	.	3	9.68	52.461667	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	.	2	6.45	.	9.885000	.

For the NIMPUTE=PCTMISSING option, the percentage of the incomplete cases, 10/31 = 32.3%, is used as the number of imputations. But the number 33 (after rounding up) is greater than 20 (as specified in the MAX= option), so only 20 imputations are generated.

When you specify the DETAILS option in REG(OXYGEN/DETAILS), the parameters that are used in each imputation for Oxygen are displayed in Output 77.6.4.

Output 77.6.4 FCS Regression Model for Oxygen

Regression Models for FCS Method										
					Imputation					
Imputed Variable	Effect	1	2	3	4	5	6	7	8	9
Oxygen	Intercept	-0.132359	0.093555	0.078587	0.063256	-0.073869	-0.070292	-0.242377	-0.176468	-0.105706
Oxygen	RunTime	-0.908663	-0.753423	-1.125549	-0.634844	-0.569809	-0.797221	-0.498457	-0.922488	-0.790878
Oxygen	RunPulse	-0.134745	0.052640	-0.135864	-0.158692	-0.319878	-0.277367	-0.510742	-0.035716	-0.169551

Regression Models for FCS Method										
					Imputation					
Imputed Variable	Effect	10	11	12	13	14	15	16	17	18
Oxygen	Intercept	-0.100698	-0.046309	-0.267186	0.074579	-0.121640	-0.041454	-0.085027	-0.025642	-0.016753
Oxygen	RunTime	-0.748476	-0.833819	-0.745716	-0.612349	-0.747333	-0.744806	-0.864134	-0.720153	-0.615441
Oxygen	RunPulse	-0.086702	-0.158535	-0.006667	-0.175998	0.030089	-0.135503	-0.120457	-0.213634	-0.065866

Regression Models for FCS Method			
		Imputation	
Imputed Variable	Effect	19	20
Oxygen	Intercept	-0.148376	-0.165002
Oxygen	RunTime	-0.658032	-0.774503
Oxygen	RunPulse	-0.227149	-0.041462

The following statements list the first 10 observations of the data set Outex6 in Output 77.6.5. Note that all missing values of all variables are imputed.

```
proc print data=outex6(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.6.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	RunPulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	10.1985	185.842
5	1	49.8740	9.2200	173.379
6	1	44.8110	11.6300	176.000
7	1	44.6299	11.9500	176.000
8	1	47.4258	10.8500	183.926
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

After the completion of the specified four imputations, the "Variance Information" table in Output 77.6.6 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. These statistics are described in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Output 77.6.6 Variance Information

Variance Information (20 Imputations)

	Variance						
Variable	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.026152	0.927386	0.954846	27.339	0.029610	0.028843	0.998560
RunTime	0.002608	0.067218	0.069957	27.02	0.040736	0.039296	0.998039
RunPulse	2.430764	3.542847	6.095149	14.23	0.720410	0.429183	0.978992

The "Parameter Estimates" table in Output 77.6.7 displays a 95% mean confidence interval and a *t* statistic with its associated *p*-value for each of the hypotheses requested with the MU0= option.

Output 77.6.7 Parameter Estimates

Parameter Estimates (20 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Oxygen	47.080908	0.977162	45.0771	49.0847	27.339	46.758087	47.512585	50.000000	-2.99	0.0059
RunTime	10.564752	0.264493	10.0221	11.1074	27.02	10.511826	10.706529	10.000000	2.14	0.0420
RunPulse	171.412215	2.468836	166.1251	176.6993	14.23	168.633931	174.275234	180.000000	-3.48	0.0036

Example 77.7: FCS Method for CLASS Variables

This example uses FCS methods to impute missing values in both continuous and CLASS variables in a data set with an arbitrary missing pattern. The following statements invoke the MI procedure and impute missing values for the Fish3 data set:

```
proc mi data=Fish3 seed=1305417 nimpute=15 out=outex7;
   class Species;
   fcs nbiter=10 discrim(Species/details) reg(Width/details);
   var Species Length Width;
run;
```

The DISCRIM option uses the discriminant function method to impute the classification variable Species, and the REG option uses the regression method to impute the continuous variable Height. By default, the regression method is also used to impute other continuous variables, Length and Width.

The "Model Information" table in Output 77.7.1 describes the method and options used in the multiple imputation process.

Output 77.7.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FISH3
Method	FCS
Number of Imputations	15
Number of Burn-in Iterations	10
Seed for random number generator	1305417

The "FCS Model Specification" table in Output 77.7.2 describes methods and imputed variables in the imputation model. The procedure uses the discriminant function method to impute the variable Species, and the regression method to impute other variables.

Output 77.7.2 FCS Model Specification

FCS Model Specification	
Method	**Imputed Variables**
Regression	Length Width
Discriminant Function	Species

The "Missing Data Patterns" table in Output 77.7.3 lists distinct missing data patterns with corresponding frequencies and percentages.

Output 77.7.3 Missing Data Patterns

Missing Data Patterns							
						Group Means	
Group	**Species**	**Length**	**Width**	**Freq**	**Percent**	**Length**	**Width**
1	X	X	X	67	77.01	27.910448	4.361860
2	X	X	.	5	5.75	24.620000	.
3	X	.	X	6	6.90	.	4.167667
4	.	X	X	6	6.90	26.683333	4.136233
5	.	X	.	2	2.30	31.500000	.
6	.	.	X	1	1.15	.	3.663600

With the specified DETAILS option for variables Species and Height, parameters used in each imputation for these two variables are displayed in the "Group Means for FCS Discriminant Method" table in Output 77.7.4 and in the "Regression Models for FCS Method" table in Output 77.7.5.

Output 77.7.4 FCS Discrim Model for Species

Group Means for FCS Discriminant Method

		Imputation									
Species	**Variable**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Parkki	**Length**	-0.268298	-0.611484	-0.430752	-0.508489	-1.096890	-0.691899	-1.097177	-0.605319	-0.349771	-0.717117
Parkki	**Width**	-0.374514	-0.920031	-0.695627	-0.444730	-1.183297	-0.684968	-1.230322	-0.630649	-0.599680	-0.735174
Perch	**Length**	0.073272	0.281238	0.135766	0.105996	0.280959	-0.003145	0.356800	0.347392	0.055848	0.448266
Perch	**Width**	0.104187	0.345404	0.211220	0.109806	0.365960	0.116315	0.397011	0.404836	0.125563	0.524498
Roach	**Length**	-0.293847	-0.296757	-0.485885	0.094638	-0.028394	-0.258734	-0.391279	-0.426078	-0.616262	-0.305244
Roach	**Width**	-0.507327	-0.352964	-0.626142	-0.033285	-0.243456	-0.459994	-0.429801	-0.433738	-0.668304	-0.419368

Group Means for FCS Discriminant Method

		Imputation				
Species	**Variable**	**11**	**12**	**13**	**14**	**15**
Parkki	**Length**	-0.636064	-0.501049	-0.384597	-0.640036	-1.019628
Parkki	**Width**	-0.834453	-0.520368	-0.569160	-0.741567	-1.064032
Perch	**Length**	0.371622	0.231022	0.150028	0.237300	0.341506
Perch	**Width**	0.482840	0.286552	0.260146	0.287303	0.341318
Roach	**Length**	-0.451065	-0.205316	-0.125811	-0.320887	-0.420115
Roach	**Width**	-0.466063	-0.317969	-0.238243	-0.467192	-0.520025

Output 77.7.5 FCS Regression Model for Height

Regression Models for FCS Method

			Imputation								
Imputed Variable	**Effect**	**Species**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Width	Intercept		-0.080952	-0.008262	-0.040466	-0.083230	-0.047121	-0.028571	-0.055278	-0.051276	-0.095069
Width	Species	Parkki	-0.100521	-0.096675	-0.022778	-0.160418	-0.092341	0.006502	-0.022182	-0.111298	-0.101806
Width	Species	Perch	0.150457	0.119791	0.108795	0.132785	0.152929	0.056631	0.114786	0.159393	0.118097
Width	Length		0.928032	0.939600	1.039315	0.975903	0.961029	0.976143	0.977985	0.973528	0.973916

Regression Models for FCS Method

		Imputation					
Imputed Variable	**Effect**	**10**	**11**	**12**	**13**	**14**	**15**
Width	Intercept	-0.075439	-0.029298	-0.033294	-0.059441	-0.117950	-0.090120
Width	Species	-0.025496	0.015175	-0.035755	-0.082546	-0.214051	-0.076865
Width	Species	0.114437	0.091675	0.068180	0.120584	0.238069	0.125656
Width	Length	1.013839	0.943677	0.956911	0.995960	0.924327	0.919800

The following statements list the first 10 observations of the data set Outex7 in Output 77.7.6:

```
proc print data=outex7(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.7.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length	Width
1	1	Roach	16.2000	2.26800
2	1	Roach	20.3000	2.82170
3	1	Roach	21.2000	2.40895
4	1	Roach	18.6497	3.17460
5	1	Roach	22.2000	3.57420
6	1	Roach	22.8000	3.35160
7	1	Roach	23.1000	3.39570
8	1	Perch	23.7000	3.88340
9	1	Roach	24.7000	3.75440
10	1	Roach	24.3000	3.54780

After the completion of five imputations by default, the "Variance Information" table in Output 77.7.7 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences for continuous variables. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. These statistics are described in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Output 77.7.7 Variance Information

Variance Information (15 Imputations)

	Variance						
Variable	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Length	0.004567	0.812129	0.817001	83.548	0.005999	0.005968	0.999602
Width	0.000162	0.029149	0.029321	83.555	0.005920	0.005890	0.999607

The "Parameter Estimates" table in Output 77.7.8 displays a 95% mean confidence interval and a t statistic with its associated p-value for each of the hypotheses requested with the default MU0=0 option.

Output 77.7.8 Parameter Estimates

Parameter Estimates (15 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Length	27.584713	0.903881	25.78710	29.38232	83.548	27.447764	27.716689	0	30.52	<.0001
Width	4.298698	0.171235	3.95815	4.63924	83.555	4.275600	4.320615	0	25.10	<.0001

Example 77.8: FCS Method with Trace Plot

This example uses FCS methods to impute missing values in both continuous and classification variables in a data set with an arbitrary missing pattern. The following statements use a logistic regression method to impute values of the classification variable Species:

```
ods graphics on;
proc mi data=Fish3 seed=1305417 nimpute=5 out=outex8;
   class Species;
   fcs plots=trace
       logistic(Species= Length Width Length*Width /details link=glogit);
   var Species Length Width;
run;
```

The "Model Information" table in Output 77.8.1 describes the method and options used in the multiple imputation process. By default, a regression method is used to impute missing values in each continuous variable.

Output 77.8.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FISH3
Method	FCS
Number of Imputations	5
Number of Burn-in Iterations	20
Seed for random number generator	1305417

The "FCS Model Specification" table in Output 77.8.2 describes methods and imputed variables in the imputation model. The procedure uses the logistic regression method to impute the variable Species, and the regression method to impute variables Height and Width.

Output 77.8.2 FCS Model Specification

FCS Model Specification	
Method	**Imputed Variables**
Regression	Length Width
Logistic Regression	Species

The "Missing Data Patterns" table in Output 77.8.3 lists distinct missing data patterns with corresponding frequencies and percentages.

Output 77.8.3 Missing Data Patterns

Missing Data Patterns

Group	Species	Length	Width	Freq	Percent	Group Means Length	Group Means Width
1	X	X	X	67	77.01	27.910448	4.361860
2	X	X	.	5	5.75	24.620000	.
3	X	.	X	6	6.90	.	4.167667
4	.	X	X	6	6.90	26.683333	4.136233
5	.	X	.	2	2.30	31.500000	.
6	.	.	X	1	1.15	.	3.663600

When you use the DETAILS keyword in the LOGISTIC option, parameters estimated from the observed data and the parameters used in each imputation are displayed in the "Logistic Models for FCS Method" table in Output 77.8.4.

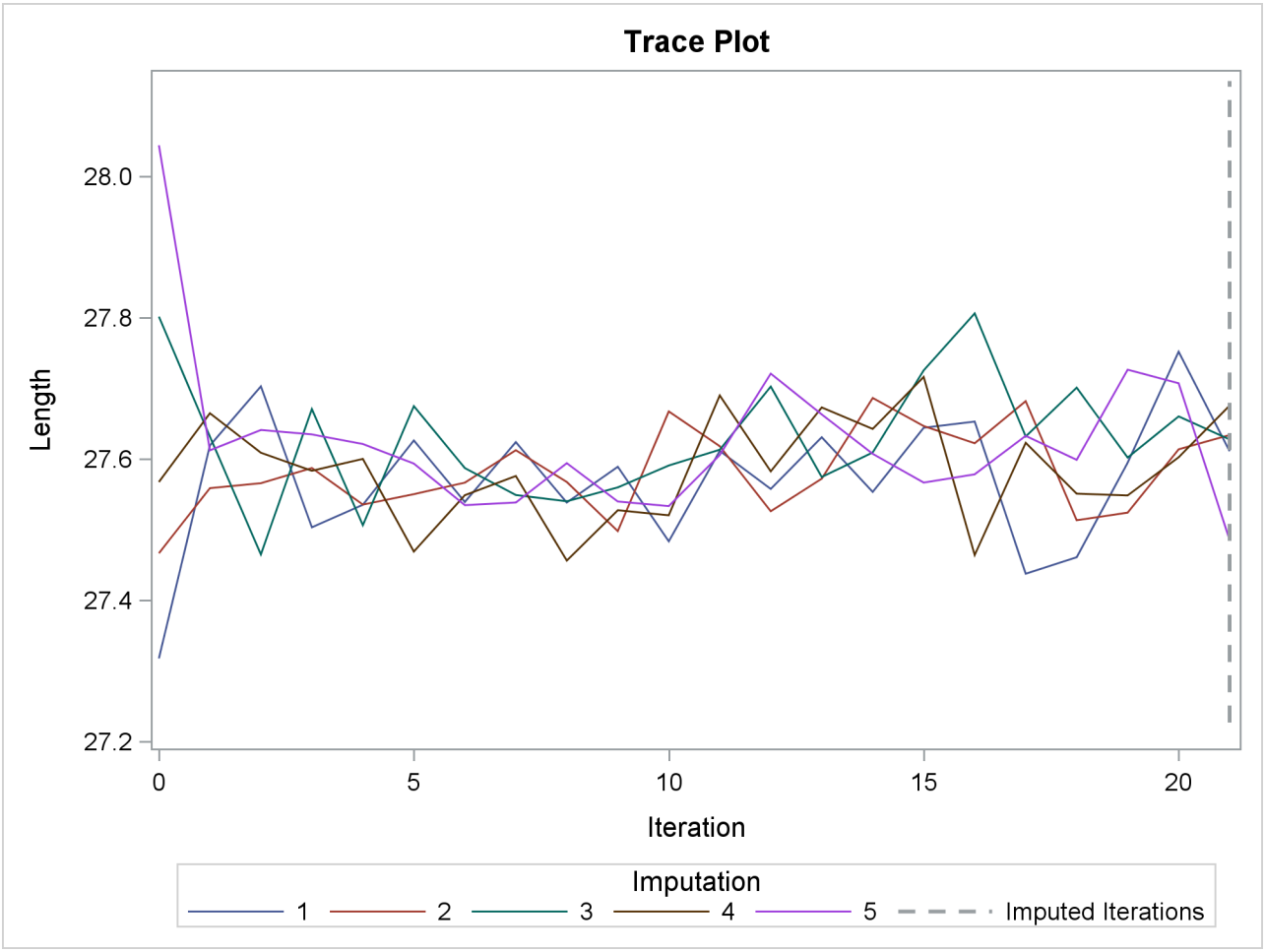
Output 77.8.4 FCS Logistic Regression Model for Species

Logistic Models for FCS Method

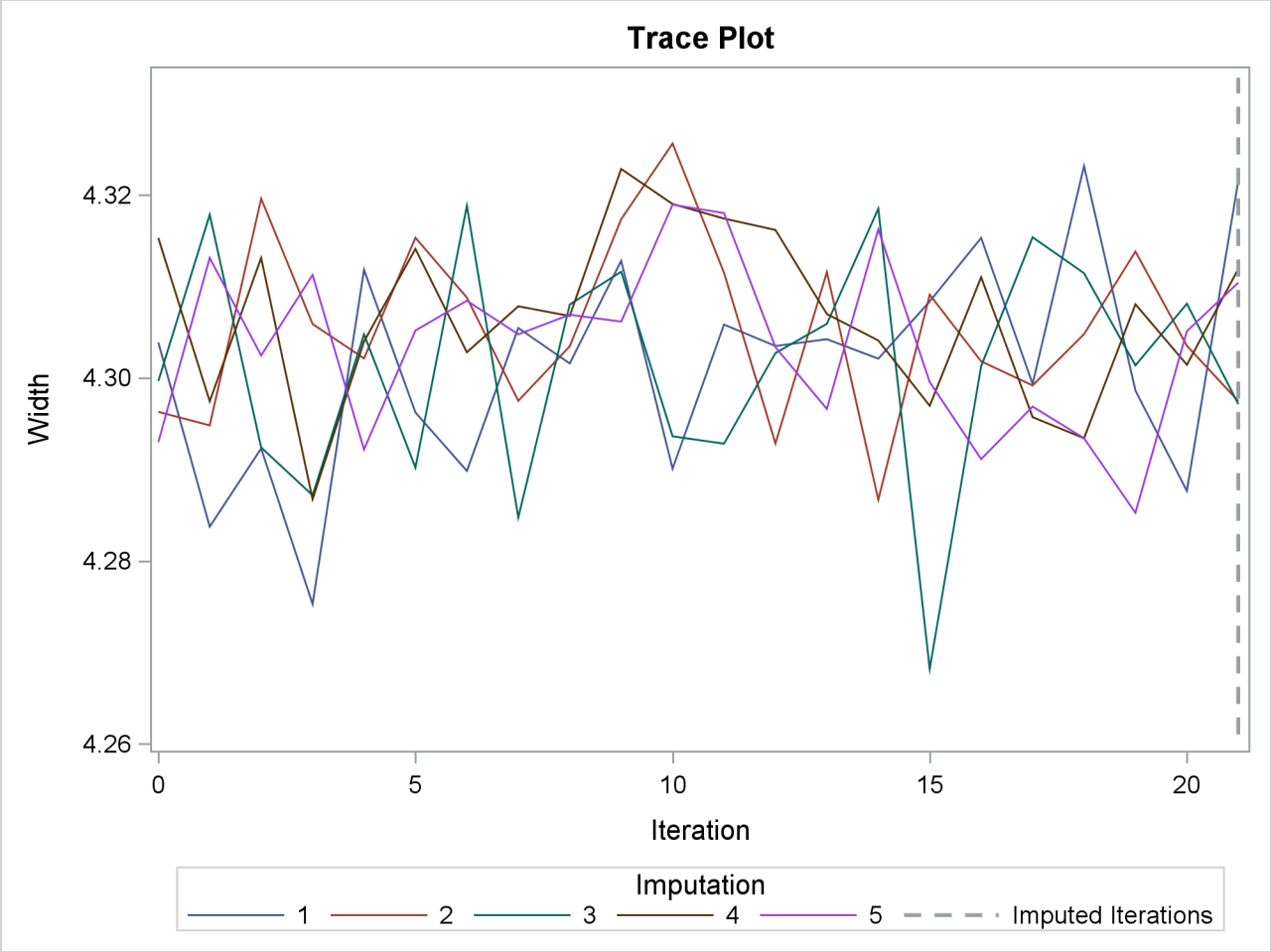
Imputed Variable	Effect	Species	Imputation 1	2	3	4	5
Species	Intercept	Parkki	-2.172588	-2.324226	-2.418362	-1.832884	-0.929242
Species	Intercept	Perch	1.878263	0.445966	1.585375	0.919562	1.547549
Species	Length	Parkki	6.107448	6.377145	2.447654	-1.004869	2.363073
Species	Length	Perch	-5.493897	-4.711566	-7.778194	-5.400749	-0.053788
Species	Width	Parkki	-8.624156	-6.965179	-5.718729	-0.997851	-2.978868
Species	Width	Perch	8.111323	5.608314	9.426901	5.502755	1.241239
Species	Length*Width	Parkki	-0.006404	2.138551	0.883903	0.072525	-0.152662
Species	Length*Width	Perch	1.151183	1.278025	1.117492	-0.195462	0.672738

With ODS Graphics enabled, the PLOTS=TRACE option displays trace plots of means for all continuous variables by default, as shown in Output 77.8.5 and Output 77.8.6. The dashed vertical lines indicate the imputed iterations—that is, the variable values used in the imputations. The plot shows no apparent trends for the two variables.

Output 77.8.5 Trace Plot for Length



Output 77.8.6 Trace Plot for Width



The following statements list the first 10 observations of the data set Outex8 in Output 77.8.7:

```
proc print data=outex8(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.8.7 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length	Width
1	1	Roach	16.2000	2.26800
2	1	Roach	20.3000	2.82170
3	1	Roach	21.2000	3.40493
4	1	Roach	22.4203	3.17460
5	1	Roach	22.2000	3.57420
6	1	Roach	22.8000	3.35160
7	1	Roach	23.1000	3.39570
8	1	Roach	23.7000	3.73166
9	1	Roach	24.7000	3.75440
10	1	Roach	24.3000	3.54780

After the completion of five imputations by default, the "Variance Information" table in Output 77.8.8 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences for continuous variables. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. These statistics are described in the section "Combining Inferences from Multiply Imputed Data Sets" on page 6123.

Output 77.8.8 Variance Information

Variance Information (5 Imputations)

Variable	Variance Between	Variance Within	Variance Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Length	0.005177	0.815388	0.821601	83.332	0.007620	0.007590	0.998484
Width	0.000108	0.028944	0.029074	83.656	0.004496	0.004486	0.999104

The "Parameter Estimates" table in Output 77.8.9 displays a 95% mean confidence interval and a t statistic with its associated p-value for each of the hypotheses requested with the default MU0=0 option.

Output 77.8.9 Parameter Estimates

Parameter Estimates (5 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Length	27.606967	0.906422	25.80424	29.40970	83.332	27.485512	27.675952	0	30.46	<.0001
Width	4.307702	0.170510	3.96860	4.64680	83.656	4.297146	4.321571	0	25.26	<.0001

Example 77.9: MCMC Method

This example uses the MCMC method to impute missing values for a data set with an arbitrary missing pattern. The following statements invoke the MI procedure and specify the MCMC method with six imputations:

```
proc mi data=Fitness1 seed=21355417 nimpute=40 mu0=50 10 180;
   mcmc chain=multiple displayinit initial=em(itprint);
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Output 77.9.1 describes the method used in the multiple imputation process. When you use the CHAIN=MULTIPLE option, the procedure uses multiple chains and completes the default 200 burn-in iterations before each imputation. The 200 burn-in iterations are used to make the iterations converge to the stationary distribution before the imputation.

Output 77.9.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	MCMC
Multiple Imputation Chain	Multiple Chains
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	40
Number of Burn-in Iterations	200
Seed for random number generator	21355417

By default, the procedure uses a noninformative Jeffreys prior to derive the posterior mode from the EM algorithm as the starting values for the MCMC method.

The "Missing Data Patterns" table in Output 77.9.2 lists distinct missing data patterns with corresponding statistics.

Output 77.9.2 Missing Data Patterns

Missing Data Patterns

						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	21	67.74	46.353810	10.809524	171.666667
2	X	X	.	4	12.90	47.109500	10.137500	.
3	X	.	.	3	9.68	52.461667	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	.	2	6.45	.	9.885000	.

When you use the ITPRINT option within the INITIAL=EM option, the procedure displays the "EM (Posterior Mode) Iteration History" table in Output 77.9.3.

Output 77.9.3 EM (Posterior Mode) Iteration History

EM (Posterior Mode) Iteration History

Iteration	**-2 Log L**	**-2 Log Posterior**	**Oxygen**	**RunTime**	**RunPulse**
0	254.482800	282.909549	47.104077	10.554858	171.381669
1	255.081168	282.051584	47.104077	10.554857	171.381652
2	255.271408	282.017488	47.104077	10.554857	171.381644
3	255.318622	282.015372	47.104002	10.554523	171.381842
4	255.330259	282.015232	47.103861	10.554388	171.382053
5	255.333161	282.015222	47.103797	10.554341	171.382150
6	255.333896	282.015222	47.103774	10.554325	171.382185
7	255.334085	282.015222	47.103766	10.554320	171.382196

When you use the DISPLAYINIT option in the MCMC statement, the "Initial Parameter Estimates for MCMC" table in Output 77.9.4 displays the starting mean and covariance estimates used in the MCMC

method. The same starting estimates are used in the MCMC method for multiple chains because the EM algorithm is applied to the same data set in each chain. You can explicitly specify different initial estimates for different imputations, or you can use the bootstrap method to generate different parameter estimates from the EM algorithm for the MCMC method.

Output 77.9.4 Initial Parameter Estimates

Initial Parameter Estimates for MCMC

TYPE	_NAME_	Oxygen	RunTime	RunPulse
MEAN		47.103766	10.554320	171.382196
COV	Oxygen	24.549967	-5.726112	-15.926036
COV	RunTime	-5.726112	1.781407	3.124798
COV	RunPulse	-15.926036	3.124798	83.164045

Output 77.9.5 and Output 77.9.6 display variance information and parameter estimates, respectively, from the multiple imputation.

Output 77.9.5 Variance Information

Variance Information (40 Imputations)

	Variance						
Variable	Between	Within	Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.020887	0.917452	0.938861	27.529	0.023336	0.022830	0.999430
RunTime	0.001897	0.066931	0.068875	27.371	0.029045	0.028264	0.999294
RunPulse	2.240072	3.674188	5.970262	16.273	0.624920	0.389201	0.990364

Output 77.9.6 Parameter Estimates

Parameter Estimates (40 Imputations)

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Oxygen	47.106918	0.968949	45.1206	49.0933	27.529	46.817748	47.429633	50.000000	-2.99	0.0059
RunTime	10.558597	0.262440	10.0205	11.0967	27.371	10.467541	10.659412	10.000000	2.13	0.0424
RunPulse	171.150201	2.443412	165.9775	176.3229	16.273	167.120956	174.569040	180.000000	-3.62	0.0022

Example 77.10: Producing Monotone Missingness with MCMC

This example uses the MCMC method to impute just enough missing values for a data set with an arbitrary missing pattern so that each imputed data set has a monotone missing pattern based on the order of variables in the VAR statement.

The following statements invoke the MI procedure and specify the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern. You must specify a VAR statement to provide the order of variables in order for the imputed data to achieve a monotone missing pattern.

```
proc mi data=Fitness1 seed=17655417 out=outex10;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Output 77.10.1 describes the method used in the multiple imputation process.

Output 77.10.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	Monotone-data MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	25
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	17655417

The "Missing Data Patterns" table in Output 77.10.2 lists distinct missing data patterns with corresponding statistics. Here, an "X" means that the variable is observed in the corresponding group, a "." means that the variable is missing and will be imputed to achieve the monotone missingness for the imputed data set, and an "O" means that the variable is missing and will not be imputed. The table also displays group-specific variable means.

Output 77.10.2 Missing Data Patterns

Missing Data Patterns

						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	21	67.74	46.353810	10.809524	171.666667
2	X	X	O	4	12.90	47.109500	10.137500	.
3	X	O	O	3	9.68	52.461667	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	O	2	6.45	.	9.885000	.

As shown in the table in Output 77.10.2, the MI procedure needs to impute only three missing values from group 4 and group 5 to achieve a monotone missing pattern for the imputed data set.

When you use the MCMC method to produce an imputed data set with a monotone missing pattern, tables of variance information and parameter estimates are not created.

The following statements are used just to show the monotone missingness of the output data set Outex10:

```
proc mi data=outex10 seed=15541 nimpute=0;
   var Oxygen RunTime RunPulse;
run;
```

The "Missing Data Patterns" table in Output 77.10.3 displays a monotone missing data pattern.

Output 77.10.3 Monotone Missing Data Patterns

The MI Procedure

Missing Data Patterns

						Group Means		
Group	Oxygen	RunTime	RunPulse	Freq	Percent	Oxygen	RunTime	RunPulse
1	X	X	X	550	70.97	46.179104	10.861364	171.863636
2	X	X	.	150	19.35	47.774669	10.053333	.
3	X	.	.	75	9.68	52.461667	.	.

The following statements impute one value for each missing value in the monotone missingness data set Outex10:

```
proc mi data=outex10 nimpute=1 seed=51343672 out=outex10a;
   monotone method=reg;
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

You can then analyze these data sets by using other SAS procedures and combine these results by using the MIANALYZE procedure. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

Example 77.11: Checking Convergence in MCMC

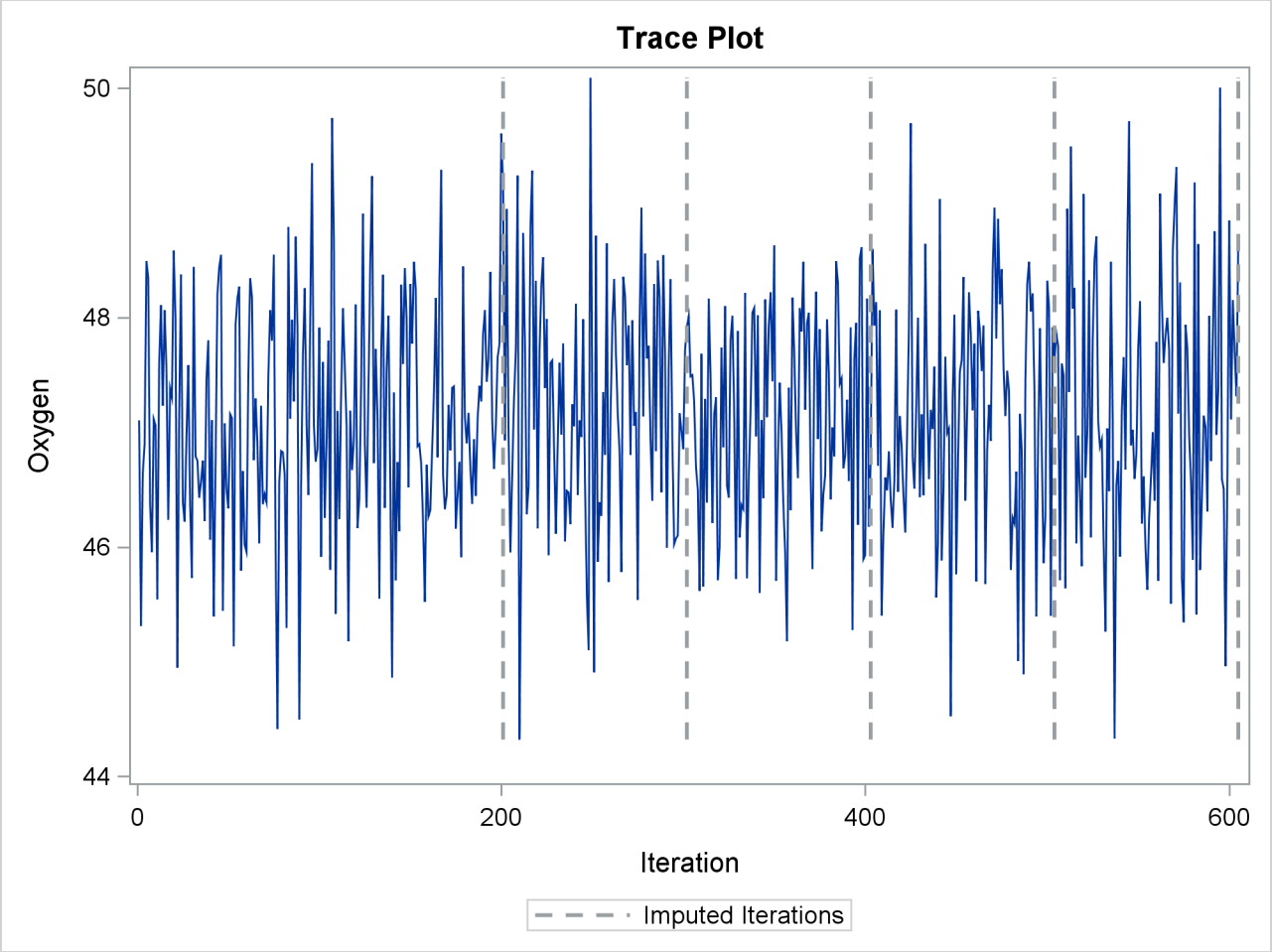
This example uses the MCMC method with a single chain. It also displays trace and autocorrelation plots to check convergence for the single chain.

The following statements use the MCMC method to create an iteration plot for the successive estimates of the mean of Oxygen. These statements also create an autocorrelation function plot for the variable Oxygen.

```
ods graphics on;
proc mi data=Fitness1 seed=501213 nimpute=5 mu0=50 10 180;
   mcmc plots=(trace(mean(Oxygen)) acf(mean(Oxygen)));
   var Oxygen RunTime RunPulse;
run;
```

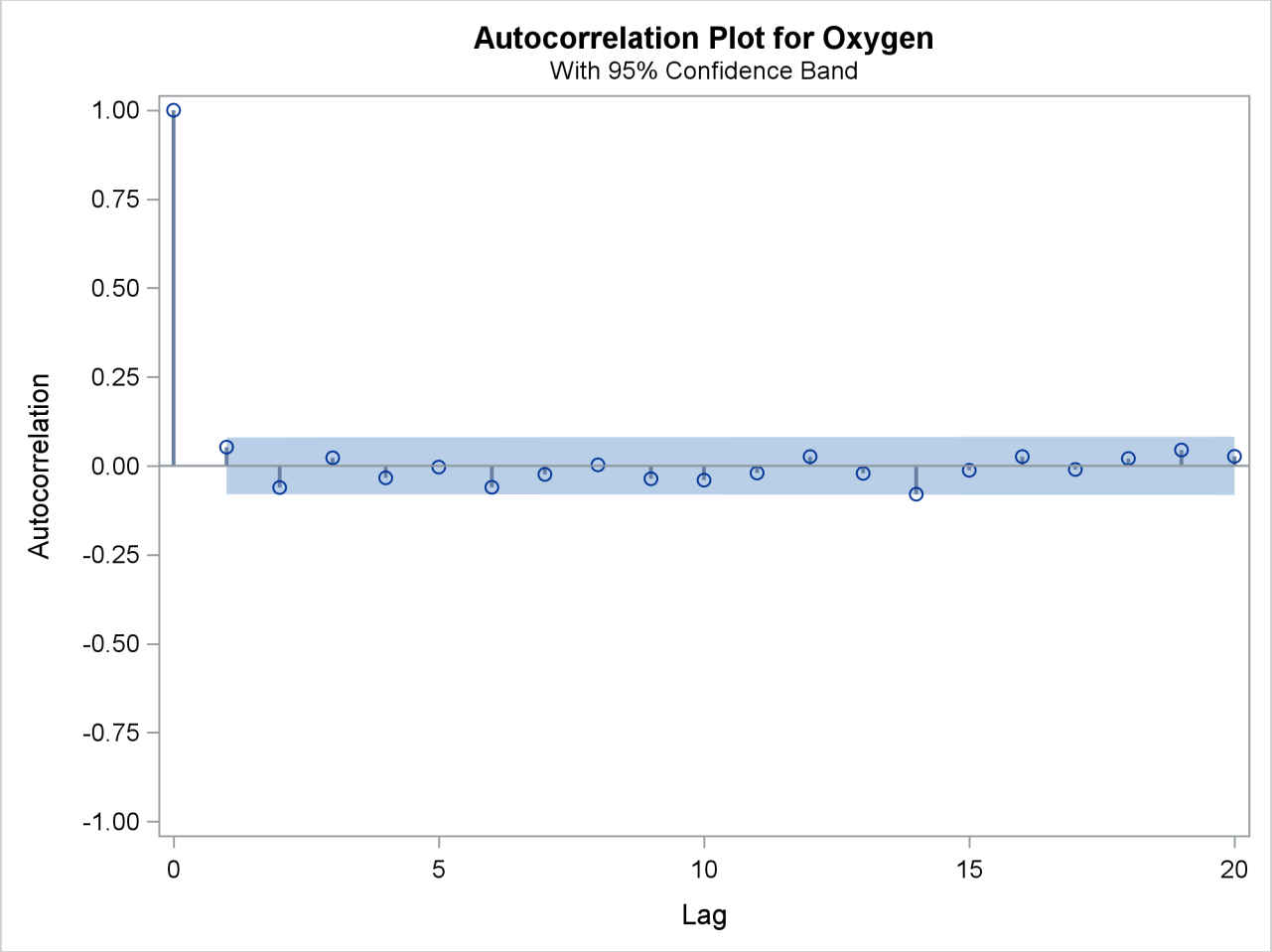
With ODS Graphics enabled, the TRACE(MEAN(OXYGEN)) option in the PLOTS= option displays the trace plot of means for the variable Oxygen, as shown in Output 77.11.1. The dashed vertical lines indicate the imputed iterations—that is, the Oxygen values used in the imputations. The plot shows no apparent trends for the variable Oxygen.

Output 77.11.1 Trace Plot for Oxygen



The ACF(MEAN(OXYGEN)) option in the PLOTS= option displays the autocorrelation plot of means for the variable Oxygen, as shown in Output 77.11.2. The autocorrelation function plot shows no significant positive or negative autocorrelation.

Output 77.11.2 Autocorrelation Function Plot for Oxygen



You can also create plots for the worst linear function, the means of other variables, the variances of variables, and the covariances between variables. Alternatively, you can use the OUTITER option to save statistics such as the means, standard deviations, covariances, –2 log LR statistic, –2 log LR statistic of the posterior mode, and worst linear function from each iteration in an output data set. Then you can do a more in-depth trace (time series) analysis of the iterations with other procedures, such as PROC AUTOREG and PROC ARIMA in the *SAS/ETS User's Guide*.

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MI procedure, see the section "ODS Graphics" on page 6142.

Example 77.12: Saving and Using Parameters for MCMC

This example uses the MCMC method with multiple chains as specified in Example 77.9. It saves the parameter values used for each imputation in an output data set of type EST called Miest. This output data set can then be used to impute missing values in other similar input data sets. The following statements invoke the MI procedure and specify the MCMC method with multiple chains to create 40 imputations:

```
proc mi data=Fitness1 seed=21355417 nimpute=40 mu0=50 10 180;
   mcmc chain=multiple initial=em outest=miest;
   var Oxygen RunTime RunPulse;
run;
```

The following statements list the parameters used for the imputations in Output 77.12.1. Note that the data set includes observations with _TYPE_='SEED' which contains the seed to start the next random number generator.

```
proc print data=miest(obs=15);
   title 'Parameters for the Imputations';
run;
```

Output 77.12.1 OUTEST Data Set

Parameters for the Imputations

Obs	_Imputation_	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	1	SEED		825240167.00	825240167.00	825240167.00
2	1	PARM		46.77	10.47	169.41
3	1	COV	Oxygen	30.59	-8.32	-50.99
4	1	COV	RunTime	-8.32	2.90	17.03
5	1	COV	RunPulse	-50.99	17.03	200.09
6	2	SEED		1895925872.00	1895925872.00	1895925872.00
7	2	PARM		47.41	10.37	173.34
8	2	COV	Oxygen	22.35	-4.44	-21.18
9	2	COV	RunTime	-4.44	1.76	1.25
10	2	COV	RunPulse	-21.18	1.25	125.67
11	3	SEED		137653011.00	137653011.00	137653011.00
12	3	PARM		48.21	10.36	170.52
13	3	COV	Oxygen	23.59	-5.25	-19.76
14	3	COV	RunTime	-5.25	1.66	5.00
15	3	COV	RunPulse	-19.76	5.00	110.99

The following statements invoke the MI procedure and use the INEST= option in the MCMC statement:

```
proc mi data=Fitness1 mu0=50 10 180;
   mcmc inest=miest;
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Output 77.12.2 describes the method used in the multiple imputation process. The remaining tables for the example are identical to the tables in Output 77.9.2, Output 77.9.4, Output 77.9.5, and Output 77.9.6 in Example 77.9.

Output 77.12.2 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	MCMC
INEST Data Set	WORK.MIEST
Number of Imputations	40

Example 77.13: Transforming to Normality

This example applies the MCMC method to the Fitness1 data set in which the variable Oxygen is transformed. Assume that Oxygen is skewed and can be transformed to normality with a logarithmic transformation. The following statements invoke the MI procedure and specify the transformation. The TRANSFORM statement specifies the log transformation for Oxygen. Note that the values displayed for Oxygen in all of the results correspond to transformed values.

```
proc mi data=Fitness1 seed=32937921
        nimpute=pctmissing mu0=50 10 180
        out=outex13;
   transform log(Oxygen);
   mcmc chain=multiple displayinit;
   var Oxygen RunTime RunPulse;
run;
```

The NIMPUTE=PCTMISSING option uses the percentage of the incomplete cases as the number of imputations.

The "Model Information" table in Output 77.13.1 describes the method and options used in the multiple imputation process.

Output 77.13.1 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FITNESS1
Method	MCMC
Multiple Imputation Chain	Multiple Chains
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	33
Number of Burn-in Iterations	200
Seed for random number generator	32937921

The "Missing Data Patterns" table in Output 77.13.2 lists distinct missing data patterns with corresponding statistics for the Fitness1 data. Note that the values of Oxygen shown in the tables are transformed values.

Output 77.13.2 Missing Data Patterns

Missing Data Patterns

						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	21	67.74	3.829760	10.809524	171.666667
2	X	X	.	4	12.90	3.851813	10.137500	.
3	X	.	.	3	9.68	3.955298	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	.	2	6.45	.	9.885000	.

Transformed Variables: Oxygen

For the NIMPUTE=PCTMISSING option, the percentage of the incomplete cases, 10/31 = 32.3%, is used as the number of imputations. Thus, 33 imputations (after rounding up) are generated.

The "Variable Transformations" table in Output 77.13.3 lists the variables that have been transformed.

Output 77.13.3 Variable Transformations

Variable Transformations	
Variable	**_Transform_**
Oxygen	LOG

The "Initial Parameter Estimates for MCMC" table in Output 77.13.4 displays the starting mean and covariance estimates used in the MCMC method.

Output 77.13.4 Initial Parameter Estimates

Initial Parameter Estimates for MCMC				
TYPE	**_NAME_**	**Oxygen**	**RunTime**	**RunPulse**
MEAN		3.846122	10.557605	171.382949
COV	Oxygen	0.010827	-0.120891	-0.328772
COV	RunTime	-0.120891	1.744580	3.011180
COV	RunPulse	-0.328772	3.011180	82.747609
Transformed Variables: Oxygen				

Output 77.13.5 displays variance information from the multiple imputation.

Output 77.13.5 Variance Information

Variance Information (33 Imputations)							
		Variance					
Variable	**Between**	**Within**	**Total**	**DF**	**Relative Increase in Variance**	**Fraction Missing Information**	**Relative Efficiency**
* **Oxygen**	0.000008582	0.000407	0.000416	27.572	0.021732	0.021297	0.999355
RunTime	0.002396	0.068237	0.070706	27.17	0.036179	0.034989	0.998941
RunPulse	0.862937	3.230086	4.119173	21.41	0.275252	0.218114	0.993434
*** Transformed Variables**							

Output 77.13.6 displays parameter estimates from the multiple imputation. Note that the parameter value of μ_0 has also been transformed using the logarithmic transformation.

Output 77.13.6 Parameter Estimates

Parameter Estimates (33 Imputations)										
				95% Confidence Limits					t for H0:	
Variable	**Mean**	**Std Error**			**DF**	**Minimum**	**Maximum**	**Mu0**	**Mean=Mu0**	**Pr > |t|**
* **Oxygen**	3.846347	0.020388	3.8046	3.8881	27.572	3.838599	3.851483	3.912023	-3.22	0.0033
RunTime	10.543129	0.265905	9.9977	11.0886	27.17	10.452018	10.633302	10.000000	2.04	0.0509
RunPulse	171.648705	2.029575	167.4329	175.8645	21.41	169.858210	173.316307	180.000000	-4.11	0.0005
*** Transformed Variables**										

The following statements list the first 10 observations of the data set Outex13 in Output 77.13.7. Note that the values for Oxygen are in the original scale.

```
proc print data=outex13(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.13.7 Imputed Data Set in Original Scale

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	RunPulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	7.1440	167.012
5	1	49.8740	9.2200	170.092
6	1	44.8110	11.6300	176.000
7	1	38.5834	11.9500	176.000
8	1	43.7376	10.8500	158.851
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

Note that the results in Output 77.13.7 can also be produced from the following statements without using a TRANSFORM statement. A transformed value of log(50)=3.91202 is used in the MU0= option.

```
data temp;
   set Fitness1;
   LogOxygen= log(Oxygen);
run;
proc mi data=temp seed=14337921 mu0=3.91202 10 180 out=outtemp;
   mcmc chain=multiple displayinit;
   var LogOxygen RunTime RunPulse;
run;
data outex13;
   set outtemp;
   Oxygen= exp(LogOxygen);
run;
```

Example 77.14: Multistage Imputation

This example uses two separate imputation procedures to complete the imputation process. In the first case, the MI procedure statements use the MCMC method to impute just enough missing values for a data set with an arbitrary missing pattern so that each imputed data set has a monotone missing pattern. In the second case, the MI procedure statements use a MONOTONE statement to impute missing values for data sets with monotone missing patterns.

The following statements are identical to those in Example 77.10. The statements invoke the MI procedure and specify the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern.

```
proc mi data=Fitness1 seed=17655417 nimpute=10 out=outex14;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

The "Missing Data Patterns" table in Output 77.14.1 lists distinct missing data patterns with corresponding statistics. Here, an "X" means that the variable is observed in the corresponding group, a "." means that the variable is missing and will be imputed to achieve the monotone missingness for the imputed data set, and an "O" means that the variable is missing and will not be imputed. The table also displays group-specific variable means.

Output 77.14.1 Missing Data Patterns

The MI Procedure

Missing Data Patterns

						Group Means		
Group	Oxygen	RunTime	RunPulse	Freq	Percent	Oxygen	RunTime	RunPulse
1	X	X	X	21	67.74	46.353810	10.809524	171.666667
2	X	X	O	4	12.90	47.109500	10.137500	.
3	X	O	O	3	9.68	52.461667	.	.
4	.	X	X	1	3.23	.	11.950000	176.000000
5	.	X	O	2	6.45	.	9.885000	.

As shown in the table, the MI procedure needs to impute only three missing values from group 4 and group 5 to achieve a monotone missing pattern for the imputed data set. When the MCMC method is used to produce an imputed data set with a monotone missing pattern, tables of variance information and parameter estimates are not created.

The following statements impute one value for each missing value in the monotone missingness data set Outex14:

```
proc mi data=outex14
        nimpute=1 seed=51343672
        out=outex14a;
   monotone reg;
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

You can then analyze these data sets by using other SAS procedures and combine these results by using the MIANALYZE procedure. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

The "Model Information" table in Output 77.14.2 shows that a monotone method is used to generate imputed values in the first BY group.

Output 77.14.2 Model Information

The MI Procedure

Imputation Number=1

Model Information	
Data Set	WORK.OUTEX14
Method	Monotone
Number of Imputations	1
Seed for random number generator	51343672

The "Monotone Model Specification" table in Output 77.14.3 describes methods and imputed variables in the imputation model. The MI procedure uses the regression method to impute the variables RunTime and RunPulse in the model.

Output 77.14.3 Monotone Model Specification

Imputation Number=1

Monotone Model Specification

Method	Imputed Variables
Regression	RunTime RunPulse

The "Missing Data Patterns" table in Output 77.14.4 lists distinct missing data patterns with corresponding statistics. It shows a monotone missing pattern for the imputed data set.

Output 77.14.4 Missing Data Patterns

Imputation Number=1

Missing Data Patterns

						Group Means		
Group	**Oxygen**	**RunTime**	**RunPulse**	**Freq**	**Percent**	**Oxygen**	**RunTime**	**RunPulse**
1	X	X	X	22	70.97	46.057479	10.861364	171.863636
2	X	X	.	6	19.35	46.745227	10.053333	.
3	X	.	.	3	9.68	52.461667	.	.

The following statements list the first 10 observations of the data set Outex14a in Output 77.14.5:

```
proc print data=outex14a(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.14.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	RunPulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	7.1569	169.914
5	1	49.8740	9.2200	159.315
6	1	44.8110	11.6300	176.000
7	1	39.8345	11.9500	176.000
8	1	45.3196	10.8500	151.252
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

This example presents an alternative to the full-data MCMC imputation, in which imputation of only a few missing values is needed to achieve a monotone missing pattern for the imputed data set. The example uses a monotone MCMC method that imputes fewer missing values in each iteration and achieves approximate stationarity in fewer iterations (Schafer 1997, p. 227). The example also demonstrates how to combine the monotone MCMC method with a method for monotone missing data, which does not rely on iterations of steps.

Example 77.15: Creating Control-Based Pattern Imputation in Sensitivity Analysis

This example illustrates the pattern-mixture model approach to 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 consists of two groups of equally allocated 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 a regression model such as

$$Y1 = Trt\ Y0$$

can be used to test the treatment effect.

Suppose that the variables Trt and Y0 are fully observed and the variable Y1 contains missing values in both the treatment and control groups. Multiple imputation for missing values often assumes that the values are missing at random. 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.

Table 77.11 shows the variables in the data set. For the control-based pattern imputation, all missing Y1 values are imputed based on the model that is constructed using observed Y1 data from the control group (Trt=0) only.

Table 77.11 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 77.15.1 lists the first 10 observations.

Output 77.15.1 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

The following statements implement the control-based pattern imputation:

```
proc mi data=Mono1 seed=14823 nimpute=15 out=outex15;
   class Trt;
   monotone reg (/details);
   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 "Model Information" table in Output 77.15.2 describes the method that is used in the multiple imputation process.

Output 77.15.2 Model Information

The MI Procedure

Model Information	
Data Set	WORK.MONO1
Method	Monotone
Number of Imputations	15
Seed for random number generator	14823

The "Monotone Model Specification" table in Output 77.15.3 describes methods and imputed variables in the imputation model. The MI procedure uses the regression method to impute the variable Y1.

Output 77.15.3 Monotone Model Specification

Monotone Model Specification	
Method	**Imputed Variables**
Regression	y1

The "Missing Data Patterns" table in Output 77.15.4 lists distinct missing data patterns and their corresponding frequencies and percentages. The table confirms a monotone missing pattern for these variables.

Output 77.15.4 Missing Data Patterns

Missing Data Patterns						
					Group Means	
Group	**y0**	**y1**	**Freq**	**Percent**	**y0**	**y1**
1	X	X	75	75.00	9.996993	10.709706
2	X	.	25	25.00	10.181488	.

By default, for each imputed variable, all available observations are used in the imputation model. When you specify the MODEL option in the MNAR statement, the "Observations Used for Imputation Models Under MNAR Assumption" table in Output 77.15.5 lists the subset of observations that are used for the imputation model for Y1.

Output 77.15.5 Observations Used for Imputation Models under MNAR Assumption

Observations Used for Imputation Models Under MNAR Assumption	
Imputed Variable	**Observations**
y1	Trt = 0

When you specify the DETAILS option, the parameters that are estimated from the observed data and the parameters that are used in each imputation are displayed in Output 77.15.6.

Output 77.15.6 Regression Model

Regression Models for Monotone Method

Imputed Variable	Effect	Obs-Data	Imputation 1	2	3	4	5	6	7	8	9
y1	Intercept	-0.30169	-0.174265	-0.280404	-0.275183	0.090601	-0.457480	-0.241909	-0.501351	-0.058460	-0.436650
y1	y0	0.69364	0.641733	0.629970	0.507776	0.752283	0.831001	0.970075	0.724584	0.623638	0.563499

Regression Models for Monotone Method

Imputed Variable	Effect	Imputation 10	11	12	13	14	15
y1	Intercept	-0.509949	-0.542411	-0.082799	-0.243293	-0.502742	-0.213113
y1	y0	0.621280	0.677104	0.562119	0.512430	0.693212	0.699355

The following statements list the first 10 observations of the output data set Outex15 in Output 77.15.7:

```
proc print data=outex15(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.15.7 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Trt	y0	y1
1	1	0	10.5212	11.3604
2	1	0	8.5871	8.5178
3	1	0	9.3274	9.5786
4	1	0	9.7519	9.6060
5	1	0	9.3495	9.4369
6	1	1	11.5192	13.2344
7	1	1	10.7841	10.7873
8	1	1	9.7717	10.9407
9	1	1	10.1455	10.8279
10	1	1	8.2463	9.6844

Example 77.16: Adjusting Imputed Continuous Values in Sensitivity Analysis

This example illustrates the pattern-mixture model approach to multiple imputation under the MNAR assumption by using specified shift parameters to adjust imputed continuous values.

Suppose that a pharmaceutical company is conducting a clinical trial to test the efficacy of a new drug. The trial consists of two groups of equally allocated 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 variables Y1 and Y2 are the efficacy scores at two successive follow-up visits.

Suppose the data set Fcs1 contains the data from the trial that have possible missing values in Y1 and Y2. Output 77.16.1 lists the first 10 observations in the data set Fcs1.

Output 77.16.1 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt	y0	y1	y2
1	0	11.4826	11.0428	13.1181
2	0	9.6775	11.0418	8.9792
3	0	9.9504	.	11.2598
4	0	11.0282	11.4097	.
5	0	10.7107	10.5782	.
6	1	9.0601	8.4791	10.6421
7	1	9.0467	9.4985	10.4719
8	1	10.6290	9.4941	.
9	1	10.1277	10.9886	11.1983
10	1	9.6910	8.4576	10.9535

Also suppose that for the treatment group, the distribution of missing Y1 responses has an expected value that is 0.4 lower than that of the corresponding distribution of the observed Y1 responses. Similarly, the distribution of missing Y2 responses has an expected value that is 0.5 lower than that of the corresponding distribution of the observed Y1 responses.

The following statements adjust the imputed Y1 and Y2 values by –0.4 and –0.5, respectively, for observations in the treatment group:

```
proc mi data=Fcs1 seed=52387 out=outex16;
   class Trt;
   fcs nbiter=25 reg( /details);
   mnar adjust( y1 /shift=-0.4 adjustobs=(Trt='1'))
        adjust( y2 /shift=-0.5 adjustobs=(Trt='1'));
   var Trt y0 y1 y2;
run;
```

The MNAR statement imputes missing values for scenarios under the MNAR assumption. The ADJUST option specifies parameters for adjusting the imputed values for specified subsets of observations. The first ADJUST option specifies the shift parameter $\delta = -0.4$ for the imputed Y1 values for observations for which TRT=1. The second ADJUST option specifies the shift parameter $\delta = -0.5$ for the imputed Y2 values for observations for which TRT=1.

Because Trt is listed in the VAR statement, it is used as a covariate for other imputed variables in the imputation process. In addition, because Trt is specified in the ADJUSTOBS= suboption, it is also used to select the subset of observations from which the imputed values for the variable are to be adjusted.

The "Model Information" table in Output 77.16.2 describes the method that is used in the multiple imputation process.

Output 77.16.2 Model Information

The MI Procedure

Model Information	
Data Set	WORK.FCS1
Method	FCS
Number of Imputations	25
Number of Burn-in Iterations	25
Seed for random number generator	52387

The "FCS Model Specification" table in Output 77.16.3 describes methods and imputed variables in the imputation model. The MI procedure uses the regression method to impute all the variables.

Output 77.16.3 FCS Model Specification

FCS Model Specification	
Method	**Imputed Variables**
Regression	y0 y1 y2
Discriminant Function	Trt

The "Missing Data Patterns" table in Output 77.16.4 lists distinct missing data patterns and their corresponding frequencies and percentages.

Output 77.16.4 Missing Data Patterns

Missing Data Patterns									
							Group Means		
Group	**Trt**	**y0**	**y1**	**y2**	**Freq**	**Percent**	**y0**	**y1**	**y2**
1	X	X	X	X	39	39.00	10.108397	10.380942	10.606255
2	X	X	X	.	29	29.00	10.207179	10.626839	.
3	X	X	.	X	32	32.00	9.604041	.	10.396557

The "MNAR Adjustments to Imputed Values" table in Output 77.16.5 lists the adjustment parameters for the five imputations.

Output 77.16.5 MNAR Adjustments to Imputed Values

MNAR Adjustments to Imputed Values		
Imputed Variable	**Observations**	**Shift**
y1	Trt = 1	-0.4000
y2	Trt = 1	-0.5000

The following statements list the first 10 observations of the data set Outex16 in Output 77.16.6:

```
proc print data=outex16(obs=10);
   var _Imputation_ Trt y0 y1 y2;
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.16.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Trt	y0	y1	y2
1	1	0	11.4826	11.0428	13.1181
2	1	0	9.6775	11.0418	8.9792
3	1	0	9.9504	11.1409	11.2598
4	1	0	11.0282	11.4097	10.8214
5	1	0	10.7107	10.5782	9.4899
6	1	1	9.0601	8.4791	10.6421
7	1	1	9.0467	9.4985	10.4719
8	1	1	10.6290	9.4941	10.7865
9	1	1	10.1277	10.9886	11.1983
10	1	1	9.6910	8.4576	10.9535

Example 77.17: Adjusting Imputed Classification Levels in Sensitivity Analysis

This example illustrates the pattern-mixture model approach to multiple imputation under the MNAR assumption by adjusting imputed classification levels.

Carpenter and Kenward (2013, pp. 240–241) describe an implementation of sensitivity analysis that adjusts an imputed missing covariate, where the covariate is a nominal classification variable.

Suppose a high school class is conducting a study to analyze the effects of an extra web-based study class and grade level on the improvement of test scores. The regression model that is used for the study is

$$\text{Score} = \text{Grade}\ \ \text{Study}\ \ \text{Score0}$$

where Grade is the grade level (with the values 6 to 8), Study is an indicator variable (with the values 1 for "completes the study class" and 0 for "does not complete the study class"), Score0 is the current test score, and Score is the test score for the subsequent test.

Also suppose that Study, Score0, and Score are fully observed and the classification variable Grade contains missing grade levels. Output 77.17.1 lists the first 10 observations in the data set Mono2.

Output 77.17.1 Student Test Data

First 10 Obs in the Student Test Data

Obs	Grade	Score0	Score	Study
1	6	64.4898	68.8210	1
2	6	72.0700	76.5328	1
3	6	65.7766	75.5567	1
4	.	70.2853	76.0180	1
5	6	74.3388	80.0617	1
6	6	70.2207	76.1606	1
7	6	68.6904	77.9770	1
8	.	72.6758	79.6895	1
9	6	64.8939	69.3889	1
10	6	66.6038	72.7793	1

The following statements use the MONOTONE and MNAR statements to impute missing values for Grade under the MNAR assumption:

```
proc mi data=Mono2 seed=34857 nimpute=20 out=outex17;
   class Study Grade;
   monotone logistic (Grade / link=glogit);
   mnar adjust( Grade (event='6') /shift=2);
   var Study Score0 Score Grade;
run;
```

The LINK=GLOGIT suboption specifies that the generalized logit function be used in fitting the logistic model for Grade. The ADJUST option specifies a shift parameter $\delta = 2$ that is applied to the generalized logit model function values for the response level GRADE=6. This assumes that students who have a missing grade level are more likely to be students in grade 6.

The "Model Information" table in Output 77.17.2 describes the method that is used in the multiple imputation process.

Output 77.17.2 Model Information

The MI Procedure

Model Information	
Data Set	WORK.MONO2
Method	Monotone
Number of Imputations	20
Seed for random number generator	34857

The "Monotone Model Specification" table in Output 77.17.3 describes methods and imputed variables in the imputation model. The MI procedure uses the logistic regression method (generalized logit model) to impute the variable Grade.

Output 77.17.3 Monotone Model Specification

Monotone Model Specification

Method	Imputed Variables
Regression	Score0 Score
Logistic Regression	Grade

The "Missing Data Patterns" table in Output 77.17.4 lists distinct missing data patterns and their corresponding frequencies and percentages.

Output 77.17.4 Missing Data Patterns

Missing Data Patterns

							Group Means	
Group	Study	Score0	Score	Grade	Freq	Percent	Score0	Score
1	X	X	X	X	128	85.33	70.418230	74.469573
2	X	X	X	.	22	14.67	69.338503	73.666293

The "MNAR Adjustments to Imputed Values" table in Output 77.17.5 lists the adjustment parameter for the 10 imputations.

Output 77.17.5 MNAR Adjustments to Imputed Values

MNAR Adjustments to Imputed Values

Imputed Variable	Event	Shift
Grade	6	2.0000

The following statements list the first 10 observations of the data set Outex17 in Output 77.17.6:

```
proc print data=outex17(obs=10);
   var _Imputation_ Grade Study Score0 Score;
   title 'First 10 Observations of the Imputed Student Test Data Set';
run;
```

Output 77.17.6 Imputed Data Set

First 10 Observations of the Imputed Student Test Data Set

Obs	_Imputation_	Grade	Study	Score0	Score
1	1	6	1	64.4898	68.8210
2	1	6	1	72.0700	76.5328
3	1	6	1	65.7766	75.5567
4	1	6	1	70.2853	76.0180
5	1	6	1	74.3388	80.0617
6	1	6	1	70.2207	76.1606
7	1	6	1	68.6904	77.9770
8	1	6	1	72.6758	79.6895
9	1	6	1	64.8939	69.3889
10	1	6	1	66.6038	72.7793

Example 77.18: Adjusting Imputed Values with Parameters in a Data Set

This example illustrates the pattern-mixture model approach in multiple imputation under the MNAR assumption by adjusting imputed values, using parameters that are stored in a data set.

Suppose that a pharmaceutical company is conducting a clinical trial to test the efficacy of a new drug. The trial consists of two groups of equally allocated 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 a regression model such as

$$Y1 = Trt\ \ Y0$$

can be used to test the efficacy of the treatment effect.

Now suppose that the variables Trt and Y0 are fully observed and the variable Y1 contains missing values in both the treatment and control groups. Table 77.12 shows the variables in the data set.

Table 77.12 Variables

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

Suppose the data set Mono3 contains the data from the trial that have missing values in Y1. Output 77.18.1 lists the first 10 observations.

Output 77.18.1 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	13.1344
7	1	10.7841	.
8	1	9.7717	10.8407
9	1	10.1455	10.7279
10	1	8.2463	9.5844

Multiple imputation often assumes that missing values are MAR. Here, however, it is plausible that the distributions of missing Y1 responses in the treatment and control groups have lower expected values than the corresponding distributions of the observed Y1 responses. Carpenter and Kenward (2013, pp. 129–130) describe an implementation of the pattern-mixture model approach that uses different shift parameters for the treatment and control groups, where the two parameters are correlated.

Assume that the expected shifts of the missing follow-up responses in the control and treatment groups, δ_c and δ_t, have a multivariate normal distribution

$$\begin{pmatrix} \delta_c \\ \delta_t \end{pmatrix} \sim N\left(\begin{pmatrix} -0.5 \\ -1 \end{pmatrix}, \begin{pmatrix} 0.01 & 0.001 \\ 0.001 & 0.01 \end{pmatrix} \right)$$

The following statements generate shift parameters for the control and treatment groups for six imputations:

```
proc iml;

   nimpute= 10;
   call randseed( 15323);
   mean= { -0.5 -1};
   cov= { 0.01 0.001 , 0.001 0.01};

  /*---- Simulate nimpute bivariate normal variates ----*/
   d= randnormal( nimpute, mean, cov);

   impu= j(nimpute, 1, 0);
   do j=1 to nimpute;  impu[j,]= j;  end;
   delta= impu || d;

  /*--- Output shift parameters for groups ----*/
   create parm1 from delta[colname={_Imputation_ Shift_C Shift_T}];
   append from delta;
quit;
```

Output 77.18.2 lists the generated shift parameters in Parm1.

Output 77.18.2 Shift Parameters for Imputations

Shift Parameters for Imputations

Obs	_IMPUTATION_	SHIFT_C	SHIFT_T
1	1	-0.56986	-0.90494
2	2	-0.38681	-0.84523
3	3	-0.58342	-0.92793
4	4	-0.48210	-0.99031
5	5	-0.57188	-1.02095
6	6	-0.57604	-1.00853
7	7	-0.44167	-0.93250
8	8	-0.53309	-1.06614
9	9	-0.53281	-1.16694
10	10	-0.53502	-1.11011

The following statements impute missing values for Y1 under the MNAR assumption. The shift parameters for the 10 imputations that are stored in the Parm1 data set are used to adjust the imputed values.

```
proc mi data=Mono3 seed=1423741 nimpute=10 out=outex18;
   class Trt;
   monotone reg;
   mnar adjust( y1 / adjustobs=(Trt='0') parms(shift=shift_c)=parm1)
        adjust( y1 / adjustobs=(Trt='1') parms(shift=shift_t)=parm1);
   var Trt y0 y1;
run;
```

The ADJUST option specifies parameters for adjusting the imputed values of Y1 for specified subsets of observations. The first ADJUST option specifies that the shift parameters that are stored in the variable SHIFT_C are to be applied to the imputed Y1 values of observations where TRT=0 for the corresponding imputations. The second ADJUST option specifies that the shift parameters that are stored in the variable SHIFT_T are to be applied to the imputed Y1 values of observations where TRT=1 for the corresponding imputations.

The "Model Information" table in Output 77.18.3 describes the method that is used in the multiple imputation process.

Output 77.18.3 Model Information

The MI Procedure

Model Information	
Data Set	WORK.MONO3
Method	Monotone
Number of Imputations	10
Seed for random number generator	1423741

The "Monotone Model Specification" table in Output 77.18.4 describes methods and imputed variables in the imputation model. The MI procedure uses the regression method to impute the variable Y1.

Output 77.18.4 Monotone Model Specification

Monotone Model Specification	
Method	**Imputed Variables**
Regression	y0 y1

The "Missing Data Patterns" table in Output 77.18.5 lists distinct missing data patterns and their corresponding frequencies and percentages. The table confirms a monotone missing pattern for these variables.

Output 77.18.5 Missing Data Patterns

Missing Data Patterns							
						Group Means	
Group	**Trt**	**y0**	**y1**	**Freq**	**Percent**	**y0**	**y1**
1	X	X	X	75	75.00	9.996993	10.655039
2	X	X	.	25	25.00	10.181488	.

The "MNAR Adjustments to Imputed Values" table in Output 77.18.6 lists the adjustment parameters for the 10 imputations.

Output 77.18.6 MNAR Adjustments to Imputed Values

MNAR Adjustments to Imputed Values

Imputed Variable	Imputation	Observations	Shift
y1	1	Trt = 0	-0.5699
	1	Trt = 1	-0.9049
	2	Trt = 0	-0.3868
	2	Trt = 1	-0.8452
	3	Trt = 0	-0.5834
	3	Trt = 1	-0.9279
	4	Trt = 0	-0.4821
	4	Trt = 1	-0.9903
	5	Trt = 0	-0.5719
	5	Trt = 1	-1.0209
	6	Trt = 0	-0.5760
	6	Trt = 1	-1.0085
	7	Trt = 0	-0.4417
	7	Trt = 1	-0.9325
	8	Trt = 0	-0.5331
	8	Trt = 1	-1.0661
	9	Trt = 0	-0.5328
	9	Trt = 1	-1.1669
	10	Trt = 0	-0.5350
	10	Trt = 1	-1.1101

The following statements list the first 10 observations of the data set Outex18 in Output 77.18.7:

```
proc print data=outex18(obs=10);
   var _Imputation_ Trt Y0 Y1;
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 77.18.7 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Trt	y0	y1
1	1	0	10.5212	11.3604
2	1	0	8.5871	8.5178
3	1	0	9.3274	8.2456
4	1	0	9.7519	10.5152
5	1	0	9.3495	9.4369
6	1	1	11.5192	13.1344
7	1	1	10.7841	9.4660
8	1	1	9.7717	10.8407
9	1	1	10.1455	10.7279
10	1	1	8.2463	9.5844

References

Allison, P. (2012). "Why You Probably Need More Imputations Than You Think." Accessed February 20, 2015. http://www.statisticalhorizons.com/more-imputations.

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.

Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*. 2nd ed. New York: John Wiley & Sons.

Barnard, J., and Meng, X. L. (1999). "Applications of Multiple Imputation in Medical Studies: From AIDS to NHANES." *Statistical Methods in Medical Research* 8:17–36.

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

Bodner, T. E. (2008). "What Improves with Increased Missing Data Imputations?" *Structural Equation Modeling* 15:651–675.

Brand, J. P. L. (1999). "Development, Implementation, and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets." Ph.D. thesis, Erasmus University.

Carpenter, J. R., and Kenward, M. G. (2013). *Multiple Imputation and Its Application*. New York: John Wiley & Sons.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society, Series B* 39:1–38.

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.

Gelman, A., and Rubin, D. B. (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science* 7:457–472.

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

Graham, J. W., Olchowski, A. E., and Gilreath, T. D. (2007). "How Many Imputations Are Really Needed? Some Practical Clarifications of Multiple Imputation Theory." *Prevention Science* 8:206–213.

Heitjan, F., and Little, R. J. A. (1991). "Multiple Imputation for the Fatal Accident Reporting System." *Journal of the Royal Statistical Society, Series C* 40:13–29.

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.

Lavori, P. W., Dawson, R., and Shera, D. (1995). "A Multiple Imputation Strategy for Clinical Trials with Truncation of Patient Data." *Statistics in Medicine* 14:1913–1925.

Li, K. H. (1988). "Imputation Using Markov Chains." *Journal of Statistical Computation and Simulation* 30:57–79.

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. (1993). "Pattern-Mixture Models for Multivariate Incomplete Data." *Journal of the American Statistical Association* 88:125–134.

Little, R. J. A. (1995). "Modeling the Drop-Out Mechanism in Repeated-Measures Studies." *Journal of the American Statistical Association* 90:1112–1121.

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

Liu, C. (1993). "Bartlett's Decomposition of the Posterior Distribution of the Covariance for Normal Monotone Ignorable Missing Data." *Journal of Multivariate Analysis* 46:198–206.

McLachlan, G. J., and Krishnan, T. (1997). *The EM Algorithm and Extensions*. New York: John Wiley & Sons.

Molenberghs, G., and Kenward, M. G. (2007). *Missing Data in Clinical Studies*. New York: John Wiley & Sons.

National Research Council (2010). *The Prevention and Treatment of Missing Data in Clinical Trials.* Panel on Handling Missing Data in Clinical Trials, Committee on National Statistics, Division of Behavioral and Social Sciences and Education. Washington, DC: National Academies Press.

O'Neill, R. T., and Temple, R. (2012). "The Prevention and Treatment of Missing Data in Clinical Trials: An FDA Perspective on the Importance of Dealing with It." *Clinical Pharmacology and Therapeutics* 91:550–554.

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.

Rosenbaum, P. R., and Rubin, D. B. (1983). "The Central Role of the Propensity Score in Observational Studies for Causal Effects." *Biometrika* 70:41–55.

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.

Schafer, J. L. (1999). "Multiple Imputation: A Primer." *Statistical Methods in Medical Research* 8:3–15.

Schenker, N., and Taylor, J. M. G. (1996). "Partially Parametric Techniques for Multiple Imputation." *Computational Statistics and Data Analysis* 22:425–446.

Tanner, M. A., and Wong, W. H. (1987). "The Calculation of Posterior Distributions by Data Augmentation." *Journal of the American Statistical Association* 82:528–540.

Van Buuren, S. (2007). "Multiple Imputation of Discrete and Continuous Data by Fully Conditional Specification." *Statistical Methods in Medical Research* 16:219–242.

Van Buuren, S. (2012). *Flexible Imputation of Missing Data*. Boca Raton, FL: Chapman & Hall/CRC.

Van Buuren, S., Boshuizen, H. C., and Knook, D. L. (1999). "Multiple Imputation of Missing Blood Pressure Covariates in Survival Analysis." *Statistics in Medicine* 18:681–694.

Von Hippel, P. T. (2009). "How to Impute Interactions, Squares, and Other Transformed Variables." *Sociological Methodology* 39:265–291.

White, I. R., Daniel, R., and Royston, P. (2010). "Avoiding Bias Due to Perfect Prediction in Multiple Imputation of Incomplete Categorical Variables." *Computational Statistics and Data Analysis* 54:2267–2275.

White, I. R., Royston, P., and Wood, A. M. (2011). "Multiple Imputation Using Chained Equations: Issues and Guidance for Practice." *Statistics in Medicine* 30:377–399.

Subject Index

adjusted degrees of freedom
    MI procedure, 6124
analyst's model
    MI procedure, 6126
approximate Bayesian bootstrap
    MI procedure, 6108
arbitrary missing pattern
    MI procedure, 6098
autocorrelation function plot
    MI procedure, 6119

Bayes' theorem
    MI procedure, 6111
Bayesian inference
    MI procedure, 6111
between-imputation variance
    MI procedure, 6123
bootstrap
    MI procedure, 6083

CCMV
    MI procedure, 6086
combining inferences
    MI procedure, 6123
complete case missing value
    MI procedure, 6086
converge in EM algorithm
    MI procedure, 6076
convergence in EM algorithm
    MI procedure, 6083
convergence in FCS Methods
    MI procedure, 6110
convergence in MCMC
    MI procedure, 6118, 6136
cumulative logit model
    MI procedure, 6091

degrees of freedom
    MI procedure, 6124
discriminant function method
    MI procedure, 6102

EM algorithm
    MI procedure, 6094, 6136

FCS method
    MI procedure, 6108
fraction of missing information
    MI procedure, 6124

generalized logit model
    MI procedure, 6091
graphics
    saving output (MI), 6139

imputation methods
    MI procedure, 6098
imputation model
    MI procedure, 6135
imputer's model
    MI procedure, 6126
input data set
    MI procedure, 6072, 6083, 6088, 6120

logistic regression method
    MI procedure, 6104
LR statistics
    MI procedure, 6118

MAR
    MI procedure, 6066, 6096, 6135
MCAR
    MI procedure, 6096
MCMC method
    MI procedure, 6111
MCMC monotone-data imputation
    MI procedure, 6136
MI procedure
    adjusted degrees of freedom, 6124
    analyst's model, 6126
    approximate Bayesian bootstrap, 6108
    arbitrary missing pattern, 6098
    autocorrelation function plot, 6119
    Bayes' theorem, 6111
    Bayesian inference, 6111
    between-imputation variance, 6123
    bootstrap, 6083
    CCMV, 6086
    combining inferences, 6123
    complete case missing value, 6086
    converge in EM algorithm, 6076
    convergence in EM algorithm, 6083
    convergence in FCS Methods, 6110
    convergence in MCMC, 6118, 6136
    cumulative logit model, 6091
    degrees of freedom, 6124
    discriminant function method, 6102
    EM algorithm, 6094, 6136
    FCS method, 6108

fraction of missing information, 6124
generalized logit model, 6091
imputation methods, 6098
imputation model, 6135
imputer's model, 6126
input data set, 6072, 6083, 6088, 6120
introductory example, 6067
logistic regression method, 6104
LR statistics, 6118
MAR, 6066, 6096, 6135
MCAR, 6096
MCMC method, 6111
MCMC monotone-data imputation, 6136
missing at random, 6066, 6096, 6135
missing not at random, 6066
MNAR, 6066, 6071, 6086, 6129
monotone method, 6099
monotone missing pattern, 6065, 6097
multiple imputation efficiency, 6124
multivariate normality assumption, 6135
NCMV, 6087
neighboring case missing value, 6087
number of imputations, 6125, 6135
ODS graph names, 6142
ODS table names, 6141
output data sets, 6074, 6077, 6084, 6121
output parameter estimates, 6084
parameter simulation, 6127
pattern-mixture model, 6128
predictive mean matching method, 6101
producing monotone missingness, 6115
propensity score method, 6108, 6136
random number generators, 6074
regression method, 6100, 6135
relative efficiency, 6124
relative increase in variance, 6124
saving graphics output, 6139
selection model, 6128
sensitivity analyses, 6127
sensitivity analysis, 6086
singularity, 6074
Summary of Issues in Multiple Imputation, 6135
suppressing output, 6074
syntax, 6070
total variance, 6123
trace plot, 6119
transformation, 6092
within-imputation variance, 6123
worst linear function of parameters, 6119
MI procedure, EM statement
output data sets, 6076
missing at random
MI procedure, 6066, 6096, 6135
missing not at random
MI procedure, 6066
MNAR
MI procedure, 6066, 6071, 6086, 6129
monotone method
MI procedure, 6099
monotone missing pattern
MI procedure, 6065, 6097
multiple imputation efficiency
MI procedure, 6124
multiple imputations analysis, 6064
multivariate normality assumption
MI procedure, 6135

NCMV
MI procedure, 6087
neighboring case missing value
MI procedure, 6087
number of imputations
MI procedure, 6125, 6135

ODS graph names
MI procedure, 6142
output data sets
MI procedure, 6074, 6077, 6084, 6121
MI procedure, EM statement, 6076
output parameter estimates
MI procedure, 6084

parameter simulation
MI procedure, 6127
pattern-mixture model
MI procedure, 6128
predictive mean matching method
MI procedure, 6101
producing monotone missingness
MI procedure, 6115
propensity score method
MI procedure, 6108, 6136

random number generators
MI procedure, 6074
regression method
MI procedure, 6100, 6135
relative efficiency
MI procedure, 6124
relative increase in variance
MI procedure, 6124

selection model
MI procedure, 6128
sensitivity analyses
MI procedure, 6127
sensitivity analysis
MI procedure, 6086
singularity

MI procedure, 6074
suppressing output
MI procedure, 6074

total variance
MI procedure, 6123
trace plot
MI procedure, 6119
transformation
MI procedure, 6092

within-imputation variance
MI procedure, 6123
worst linear function of parameters
MI procedure, 6119

Syntax Index

ACF option
MCMC statement (MI), 6084
ACFPLOT option
MCMC statement (MI), 6137
ADJUST option
MNAR statement (MI), 6087
ADJUSTOBS= option
MNAR statement, 6087
ALPHA= option
PROC MI statement, 6072

BOOTSTRAP option
MCMC statement (MI), 6083
BOXCOX transformation
TRANSFORM statement (MI), 6092
BY statement
MI procedure, 6075

C= option
TRANSFORM statement (MI), 6093
CCONF= option
MCMC statement (MI), 6138
CCONNECT= option
MCMC statement (MI), 6140
CFRAME= option
MCMC statement (MI), 6138, 6140
CHAIN= option
MCMC statement (MI), 6082
CLASS statement
MI procedure, 6075
CLASSEFFECTS= option
FCS statement (MI), 6079
MONOTONE statement (MI), 6089
CNEEDLES= option
MCMC statement (MI), 6138
CONVERGE option
EM statement (MI), 6076
CONVERGE= option
MCMC statement (MI), 6083
COV option
MCMC statement (MI), 6137, 6139
CREF= option
MCMC statement (MI), 6138
CSYMBOL= option
MCMC statement (MI), 6138, 6140

DATA= option
PROC MI statement, 6072
DELTA= option
MNAR statement, 6087
DESCENDING option
FCS statement (MI), 6079
MONOTONE statement (MI), 6090
DETAILS option
FCS statement (MI), 6079
MONOTONE statement (MI), 6089, 6090
DISCRIM option
FCS statement (MI), 6079
MONOTONE statement (MI), 6089
DISPLAYINIT option
MCMC statement (MI), 6082
DISPLAYPATTERN= option
PROC MI statement, 6072

EM statement
MI procedure, 6075
EXP transformation
TRANSFORM statement (MI), 6092

FCS statement
MI procedure, 6076
FREQ statement
MI procedure, 6081

GOUT= option
MCMC statement (MI), 6139

HSYMBOL= option
MCMC statement (MI), 6138, 6140

IMPUTE= option
MCMC statement (MI), 6082
INEST= option
MCMC statement (MI), 6083
INITIAL option
EM statement (MI), 6076
INITIAL= option
MCMC statement (MI), 6083
ITPRINT option
EM statement (MI), 6076
MCMC statement (MI), 6083

LAMBDA= option
TRANSFORM statement (MI), 6093
LCONF= option
MCMC statement (MI), 6138
LCONNECT= option
MCMC statement (MI), 6140

LIKELIHOOD= option
FCS statement (MI), 6079
MONOTONE statement (MI), 6090
LINK= option
FCS statement (MI), 6080
MONOTONE statement (MI), 6091
LOG option
MCMC statement (MI), 6138, 6140
LOG transformation
TRANSFORM statement (MI), 6092
LOGISTIC option
FCS statement (MI), 6079
MONOTONE statement (MI), 6090
LOGIT transformation
TRANSFORM statement (MI), 6092
LREF= option
MCMC statement (MI), 6138

MAXIMUM= option
PROC MI statement, 6072
MAXITER= option
EM statement (MI), 6076
MCMC statement (MI), 6083
MCMC statement
MI procedure, 6082
MEAN option
MCMC statement (MI), 6137, 6139
MI procedure, BY statement, 6075
MI procedure, CLASS statement, 6075
MI procedure, EM statement, 6075
CONVERGE option, 6076
INITIAL= option, 6076
ITPRINT option, 6076
MAXITER= option, 6076
OUT= option, 6076
OUTEM= option, 6076
OUTITER= option, 6076
XCONV option, 6076
MI procedure, FCS statement, 6076
CLASSEFFECTS= option, 6079
DESCENDING option, 6079
DETAILS option, 6079
DISCRIM option, 6079
LIKELIHOOD= option, 6079
LINK= option, 6080
LOGISTIC option, 6079
NBITER= option, 6077
ORDER= option, 6080
OUTITER= option, 6077
PCOV= option, 6079
PRIOR= option, 6079
REG option, 6080
REGPMM option, 6081
REGPREDMEANMATCH option, 6081
REGRESSION option, 6080
TRACE option, 6077
MI procedure, FREQ statement, 6081
MI procedure, MCMC statement, 6082
ACF option, 6084
ACFPLOT option, 6137
BOOTSTRAP option, 6083
CCONF= option, 6138
CCONNECT= option, 6140
CFRAME= option, 6138, 6140
CHAIN= option, 6082
CNEEDLES= option, 6138
CONVERGE= option, 6083
COV option, 6137, 6139
CREF= option, 6138
CSYMBOL= option, 6138, 6140
DISPLAYINIT option, 6082
GOUT= option, 6139
HSYMBOL= option, 6138, 6140
IMPUTE= option, 6082
INEST= option, 6083
INITIAL= option, 6083
ITPRINT option, 6083
LCONF= option, 6138
LCONNECT= option, 6140
LOG option, 6138, 6140
LREF= option, 6138
MAXITER= option, 6083
MEAN option, 6137, 6139
NAME= option, 6138, 6140
NBITER= option, 6083
NITER= option, 6084
NLAG= option, 6138
OUTEST= option, 6084
OUTITER= option, 6084
PRIOR= option, 6085
START= option, 6085
SYMBOL= option, 6138, 6140
TIMEPLOT option, 6139
TITLE= option, 6139, 6140
TRACE option, 6084
WCONF= option, 6139
WCONNECT= option, 6141
WLF option, 6085, 6137, 6139
WNEEDLES= option, 6139
WREF= option, 6139
XCONV= option, 6083
MI procedure, MNAR statement, 6086
ADJUST option, 6087
ADJUSTOBS= option, 6087
DELTA= option, 6087
MODEL option, 6086
MODELOBS= option, 6086
PARMS= option, 6088

SCALE= option, 6087
SHIFT= option, 6087
SIGMA= option, 6087
MI procedure, MONOTONE statement, 6088
CLASSEFFECTS= option, 6089
DESCENDING option, 6090
DETAILS option, 6089, 6090
DISCRIM option, 6089
LIKELIHOOD= option, 6090
LINK= option, 6091
LOGISTIC option, 6090
ORDER= option, 6091
PCOV= option, 6090
PRIOR= option, 6090
PROPENSITY option, 6091
REG option, 6091
REGPMM option, 6091
REGPREDMEANMATCH option, 6091
REGRESSION option, 6091
MI procedure, PROC MI statement, 6071
ALPHA= option, 6072
DATA= option, 6072
DISPLAYPATTERN= option, 6072
MAXIMUM= option, 6072
MINIMUM= option, 6073
MINMAXITER= option, 6073
MU0= option, 6073
NIMPUTE= option, 6073
NOPRINT option, 6074
OUT= option, 6074
ROUND= option, 6074
SEED option, 6074
SIMPLE, 6074
SINGULAR option, 6074
THETA0= option, 6073
MI procedure, TRANSFORM statement, 6092
BOXCOX transformation, 6092
C= option, 6093
EXP transformation, 6092
LAMBDA= option, 6093
LOG transformation, 6092
LOGIT transformation, 6092
POWER transformation, 6093
MI procedure, VAR statement, 6093
MINIMUM= option
PROC MI statement, 6073
MINMAXITER= option
PROC MI statement, 6073
MNAR statement
MI procedure, 6086
MODEL option
MNAR statement (MI), 6086
MODELOBS= option
MNAR statement (MI), 6086
MONOTONE statement
MI procedure, 6088
MU0= option
PROC MI statement, 6073

NAME= option
MCMC statement (MI), 6138, 6140
NBITER= option
FCS statement (MI), 6077
MCMC statement (MI), 6083
NIMPUTE= option
PROC MI statement, 6073
NITER= option
MCMC statement (MI), 6084
NLAG= option
MCMC statement (MI), 6138
NOPRINT option
PROC MI statement, 6074

ORDER= option
FCS statement (MI), 6080
MONOTONE statement (MI), 6091
OUT= option
EM statement (MI), 6076
PROC MI statement, 6074
OUTEM= option
EM statement (MI), 6076
OUTEST= option
MCMC statement (MI), 6084
OUTITER= option
EM statement (MI), 6076
FCS statement (MI), 6077
MCMC statement (MI), 6084

PARMS= option
MNAR statement, 6088
PCOV= option
FCS statement (MI), 6079
MONOTONE statement (MI), 6090
POWER transformation
TRANSFORM statement (MI), 6093
PRIOR= option
FCS statement (MI), 6079
MCMC statement (MI), 6085
MONOTONE statement (MI), 6090
PROC MI statement, *see* MI procedure
PROPENSITY option
MONOTONE statement (MI), 6091

REG option
FCS statement (MI), 6080
MONOTONE statement (MI), 6091
REGPMM option
FCS statement (MI), 6081
MONOTONE statement (MI), 6091

REGPREDMEANMATCH option
FCS statement (MI), 6081
MONOTONE statement (MI), 6091
REGRESSION option
FCS statement (MI), 6080
MONOTONE statement (MI), 6091
ROUND= option
PROC MI statement, 6074

SCALE= option
MNAR statement, 6087
SEED option
PROC MI statement, 6074
SHIFT= option
MNAR statement, 6087
SIGMA= option
MNAR statement, 6087
SIMPLE option
PROC MI statement, 6074
SINGULAR option
PROC MI statement, 6074
START= option
MCMC statement (MI), 6085
SYMBOL= option
MCMC statement (MI), 6138, 6140

THETA0= option
PROC MI statement, 6073
TIMEPLOT option
MCMC statement (MI), 6139
TITLE= option
MCMC statement (MI), 6139, 6140
TRACE option
FCS statement (MI), 6077
MCMC statement (MI), 6084
TRANSFORM statement
MI procedure, 6092

VAR statement
MI procedure, 6093

WCONF= option
MCMC statement (MI), 6139
WCONNECT= option
MCMC statement (MI), 6141
WLF option
MCMC statement (MI), 6085, 6137, 6139
WNEEDLES= option
MCMC statement (MI), 6139
WREF= option
MCMC statement (MI), 6139

XCONV option
EM statement (MI), 6076
XCONV= option
MCMC statement (MI), 6083