# SAS/STAT® 9.3 User's Guide

# The MI Procedure

## (Chapter)

SAS® Documentation

This document is an individual chapter from *SAS/STAT® 9.3 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2011. *SAS/STAT® 9.3 User's Guide*. Cary, NC: SAS Institute Inc.



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

1st electronic book, July 2011

# Chapter 56

# The MI Procedure

**Contents**

# 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 or ordinal response, or 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.

For each imputation, the process begins with the filling in of all missing values sequentially over the variables taken one at a time, and then these filled-in values are imputed sequentially over the variables at each of the 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, the MIANALYZE procedure can be used to generate valid statistical inferences about these parameters by combining results from the *m* analyses.

Often, as few as three to five imputations are adequate in multiple imputation (Rubin 1996, p. 480). The relative efficiency of the small *m* imputation estimator is high for cases with little missing information (Rubin 1987, p. 114). (Also see the section "Multiple Imputation Efficiency" on page 4580.)

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 4580.

# Getting Started: MI Procedure

The Fitness data described in the REG procedure are measurements of 31 individuals in a physical fitness course. See Chapter 76, "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 4552 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 56.1 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 five 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 56.1** 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 | 5 |
| 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 56.2 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 4553.

**Figure 56.2** Missing Data Patterns

**Missing Data Patterns**

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | . | 4 | 12.90 |
| 3 | X | . | . | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | . | 2 | 6.45 |

**Missing Data Patterns**

| | Group Means | | |
|---|---|---|---|
| Group | Oxygen | RunTime | RunPulse |
| 1 | 46.353810 | 10.809524 | 171.666667 |
| 2 | 47.109500 | 10.137500 | . |
| 3 | 52.461667 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 9.885000 | . |

After the completion of *m* imputations, the "Variance Information" table in Figure 56.3 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 4578.

**Figure 56.3** Variance Information

Variance Information

| Variable | Variance Between | Variance Within | Variance Total | DF |
|---|---|---|---|---|
| Oxygen | 0.056930 | 0.954041 | 1.022356 | 25.549 |
| RunTime | 0.000811 | 0.064496 | 0.065469 | 27.721 |
| RunPulse | 0.922032 | 3.269089 | 4.375528 | 15.753 |

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Oxygen | 0.071606 | 0.068898 | 0.986408 |
| RunTime | 0.015084 | 0.014968 | 0.997015 |
| RunPulse | 0.338455 | 0.275664 | 0.947748 |

The “Parameter Estimates” table in Figure 56.4 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 4578.

**Figure 56.4** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Oxygen | 47.094040 | 1.011116 | 45.0139 | 49.1742 | 25.549 |
| RunTime | 10.572073 | 0.255870 | 10.0477 | 11.0964 | 27.721 |
| RunPulse | 171.787793 | 2.091776 | 167.3478 | 176.2278 | 15.753 |

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Oxygen | 46.783898 | 47.395550 | 50.000000 | -2.87 | 0.0081 |
| RunTime | 10.526392 | 10.599616 | 10.000000 | 2.24 | 0.0336 |
| RunPulse | 170.774818 | 173.122002 | 180.000000 | -3.93 | 0.0012 |

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 56.5 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 56.5** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Oxygen | RunTime | Run Pulse |
|---|---|---|---|---|
| 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 PROC MI:

**PROC MI** < *options* > ;
  **BY** *variables* ;
  **CLASS** *variables* ;
  **EM** < *options* > ;
  **FCS** < *options* > ;
  **FREQ** *variable* ;
  **MCMC** < *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. 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 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* > **;**

Table 56.1 summarizes the options available in the PROC MI statement.

**Table 56.1** 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 |

**Table 56.1** *continued*

| Option | Description |
|---|---|
| SINGULAR= | Specifies the singularity criterion |
| **Statistical Analysis** | |
| ALPHA= | Specifies the level for the confidence interval, $(1-\alpha)$ |
| MU0= | Specifies means under the null hypothesis |
| **Printed Output** | |
| 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=**$\alpha$

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.

**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 are applicable only if you use the MCMC method or the monotone regression method.

When specifying 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, the "MAXIMUM= 100 ." option sets a maximum of 100 for the first analysis variable only and no maximum for the remaining variables. The "MAXIMUM= . 100" option sets a maximum of 100 for the second analysis variable only 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 $\mu_0$ under the null hypothesis $\mu = \mu_0$ 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 $\mu_0$ for a transformed variable are not specified, then a value of zero is used for the resulting $\mu_0$ after transformation.

**NIMPUTE=***number*

specifies the number of imputations. The default is NIMPUTE=5. 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 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 4576 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 56.3 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 4550.

**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 $\mathbf{S} = \mathbf{\Gamma \Lambda \Gamma}'$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_j$, $j = 1, \ldots, v$, 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 $\lambda_j$ 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 on 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 *Base SAS 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 56.1.

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 4576 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 4576 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 4576 for a description of this data set.

## FCS Statement (Experimental)

**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 56.2 summarizes the options available for the FCS statement.

**Table 56.2** Summary of Options in FCS

| Option | Description |
|---|---|
| **Imputation Details** | |
| NBITER= | Specifies the number of burn-in iterations |
| ORDER= | Specifies the variable ordering in the filled-in and imputation phases |
| **Data Set** | |
| OUTITER= | Outputs parameter estimates used in iterations |
| **ODS Output Graphics** | |
| 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=10.

**ORDER=FREQ | VAR**

specifies the variable ordering in which to impute missing values in the filled-in and imputation phases. The ORDER=FREQ option orders the variables by the descending frequency counts of variables and the ORDER=VAR orders the variables as specified in the VAR statement. The default is ORDER=FREQ.

**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 4576 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 requesting plots. 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;
ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 609 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 41, "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 continuous variables and the discriminant function method is used for all classification variables. For each imputed variable, all other 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, 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 the discriminant function method, only the continuous variables can be used as covariate effects.

With an FCS statement, the variables are imputed sequentially in the order specified in the ORDER= option. 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 a discriminant function method to impute missing values without using the ordering of the class levels. For an ordinal classification variable, you can use a logistic regression method 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 4563 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. Only the continuous variables are allowed as covariate effects. The available options are DETAILS, PCOV=, and PRIOR=. The DETAILS option displays the group means and pooled covariance matrix used in each imputation. The PCOV= option specifies the pooled covariance used in the discriminant method. Valid values for the PCOV= option are as follows:

| | |
|---|---|
| FIXED | uses the observed-data pooled covariance matrix for each imputation. |
| POSTERIOR | draws a pooled covariance matrix from its posterior distribution. |

The default is PCOV=POSTERIOR. See the section "Monotone and FCS Discriminant Function Methods" on page 4560 for a detailed description of the method.

The PRIOR= option specifies the prior probabilities of group membership. Valid values for the PRIOR= option are as follows:

| | |
|---|---|
| EQUAL | sets the prior probabilities equal for all groups. |
| PROPORTIONAL | sets the prior probabilities proportion to the group sample sizes. |
| JEFFREYS < =*c* > | specifies a noninformative prior, $0 < c < 1$. If the number $c$ is not specified, JEFFREYS=0.5. |
| RIDGE < =*d* > | specifies a ridge prior, $d > 0$. If the number $d$ is not specified, RIDGE=0.25. |

The default is PRIOR=JEFFREYS. See the section "Monotone and FCS Discriminant Function Methods" on page 4560 for a detailed description of the method.

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

specifies the logistic regression method of classification variables. The available options are DETAILS, ORDER=, and DESCENDING. The DETAILS option displays the regression coefficients in the logistic regression model used in each imputation.

When the imputed variable has more than two response levels, the ordinal logistic regression method is used. The ORDER= option specifies the sorting order for the levels of the response variable. Valid values for the ORDER= option are as follows:

| | |
|---|---|
| DATA | sorts by the order of appearance in the input data set. |
| FORMATTED | sorts by their external formatted values. |
| FREQ | sorts by the descending frequency counts. |
| INTERNAL | sorts by the unformatted values. |

By default, ORDER=FORMATTED.

The option DESCENDING reverses the sorting order for the levels of the response variables.

See the section "Monotone and FCS Logistic Regression Methods" on page 4562 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 4557 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 4558 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. 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) reg(y3= y1 y2) logistic(c1= y1 y2 y1*y2);
   var y1 y2 y3 c1;
run;
```

## FREQ Statement

**FREQ** *variable* **;**

If one variable in your input data set represents the frequency of occurrence of other values in the observation, specify the variable name in a FREQ statement. PROC MI then treats the data set as if each observation appears $n$ times, where $n$ is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered to be equal to the sum of the FREQ variable when PROC MI calculates significance probabilities.

## MCMC Statement

**MCMC** < *options* > **;**

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

Table 56.3 summarizes the options available for the MCMC statement.

**Table 56.3** 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 Output Graphics** | |
| PLOTS=TRACE | Displays trace plots |
| PLOTS=ACF | Displays autocorrelation plots |
| **Traditional Graphics** | |
| TIMEPLOT | Displays trace plots |
| ACFPLOT | Displays autocorrelation plots |
| GOUT= | Specifies the graphics catalog name for saving graphics output |
| **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).

**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. 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 4574; see also Schafer (1997, pp. 120–126) and the *SAS/ETS User's Guide*.

**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.

**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 Software: Reference*.

**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 4575.

**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 4575 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 4576 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 4576 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*

specifies the prior information for the means and covariances. Valid values for *name* are as follows:

| | |
|---|---|
| JEFFREYS | specifies a noninformative prior. |
| RIDGE=*number* | specifies a ridge prior. |
| INPUT=*SAS-data-set* | 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 4567 and the section "Posterior Step" on page 4568. 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 4575 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.

**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. 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 4573 and Schafer (1997, pp. 120–126).

**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 4573.

## 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 56.4 summarizes the options available for the MONOTONE statement.

**Table 56.4** 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 41, "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.

If no covariates are specified, then all preceding variables 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 the discriminant function method, only the continuous variables can be used as covariate effects.

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.

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

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 a discriminant function method to impute missing values without using the ordering of the class levels. For an ordinal classification variable, you can use a logistic regression method 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. Only the continuous variables are allowed as covariate effects. The available options are DETAILS, PCOV=, and PRIOR=. The DETAILS option displays the group means and pooled covariance matrix used in each imputation. The PCOV= option specifies the pooled covariance used in the discriminant method. Valid values for the PCOV= option are as follows:

| | |
|---|---|
| FIXED | uses the observed-data pooled covariance matrix for each imputation. |
| POSTERIOR | draws a pooled covariance matrix from its posterior distribution. |

The default is PCOV=POSTERIOR. See the section "Monotone and FCS Discriminant Function Methods" on page 4560 for a detailed description of the method.

The PRIOR= option specifies the prior probabilities of group membership. Valid values for the PRIOR= option are as follows:

| | |
|---|---|
| EQUAL | sets the prior probabilities equal for all groups. |
| PROPORTIONAL | sets the prior probabilities proportion to the group sample sizes. |
| JEFFREYS < =*c* > | specifies a noninformative prior, $0 < c < 1$. If the number $c$ is not specified, JEFFREYS=0.5. |
| RIDGE < =*d* > | specifies a ridge prior, $d > 0$. If the number $d$ is not specified, RIDGE=0.25. |

The default is PRIOR=JEFFREYS. See the section "Monotone and FCS Discriminant Function Methods" on page 4560 for a detailed description of the method.

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

specifies the logistic regression method of classification variables. The available options are DETAILS, ORDER=, and DESCENDING. The DETAILS option displays the regression coefficients in the logistic regression model used in each imputation.

When the imputed variable has more than two response levels, the ordinal logistic regression method is used. The ORDER= option specifies the sorting order for the levels of the response variable. Valid values for the ORDER= option are as follows:

| | |
|---|---|
| DATA | sorts by the order of appearance in the input data set. |
| FORMATTED | sorts by their external formatted values. |
| FREQ | sorts by the descending frequency counts. |
| INTERNAL | sorts by the unformatted values. |

By default, ORDER=FORMATTED.

The option DESCENDING reverses the sorting order for the levels of the response variables.

See the section “Monotone and FCS Logistic Regression Methods” on page 4562 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 4559 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 4557 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 4558 for a detailed description of the method.

With a MONOTONE statement, the missing values of a variable are imputed when the variable is either explicitly specified in the method or implicitly specified when a method is specified without imputed variables. These variables are imputed sequentially in the order specified in the VAR statement. For example, the following MI procedure statements use the logistic regression method to impute variable c1 from effects y1, y2, and y1 $*$ y2 first, and then use the regression method to impute variable y3 from effects y1, y2, and c1:

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

The variables y1 and y2 are not imputed since y1 is the leading variable in the VAR statement and y2 is not specified as an imputed variable in the MONOTONE 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 $\frac{(\mathsf{Y}+c)^{\lambda}-1}{\lambda}$, where $c$ is a constant such that each value of $\mathsf{Y}+c$ 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 $\mathrm{e}^{(\mathsf{Y}+c)}$, where $c$ is a constant.

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

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

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

The following options provide the constant $c$ and $\lambda$ 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 $\lambda$ value in the power and Box-Cox transformations. You must specify the $\lambda$ value for these two transformations.

For example, the following statement requests that variables log(y1), a logarithmic transformation for the variable y1, and $\sqrt{\mathsf{y2}+1}$, 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 $\boldsymbol{\mu}_0$ for a transformed variable, the same transformation for the variable is also applied to its corresponding MU0= value in the $t$ test. Otherwise, $\boldsymbol{\mu}_0 = 0$ is used for the transformed variable. See Example 56.10 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

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

and the CSSCP matrix is

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

where each summation is over the fully observed observations.

The sample covariance matrix is

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

and is an unbiased estimate of the covariance matrix.

The correlation matrix **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 **D** is a diagonal matrix whose diagonal elements are the square roots of the diagonal elements of **S**.

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

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

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

The variance is

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

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. The books by Little and Rubin (2002), Schafer (1997), and McLachlan and Krishnan (1997) provide a detailed description and applications of the EM algorithm.

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 $g$th 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 $g$th group, the summation is over observations in the $g$th 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.

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 initial 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 continuous 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 $Y_{mis}$:

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

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 56.6 displays the eight groups of observations and an unique missing pattern for each group:

**Figure 56.6** 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 | . | . | . |

Here, an "X" means that the variable is observed in the corresponding group and a "." means that the variable is missing.

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 56.7 displays a data set of three variables with a monotone missing pattern.

**Figure 56.7** Monotone Missing Patterns

**Monotone Missing Data Patterns**

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

Figure 56.8 displays a data set of three variables with a non-monotone missing pattern.

**Figure 56.8** 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 56.5.

**Table 56.5** Imputation Methods in PROC MI

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

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 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 and Oudshoorn 1999, 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, and use the logistic regression method to impute missing values for a classification variable when the variable has a binary or ordinal response, or use the discriminant function method 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 4572.

## 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

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

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 4557, "Monotone and FCS Predictive Mean Matching Methods" on page 4558, "Monotone Propensity Score Method" on page 4559, "Monotone and FCS Discriminant Function Methods" on page 4560, and "Monotone and FCS Logistic Regression Methods" on page 4562 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 $X_1, X_2, \ldots, X_k$.

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 $\sigma_{*j}^2$ are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, $\sigma_j^2$, 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 4557, the following steps are used to generate imputed values:

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

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

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 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).

## 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, …, $g$ 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,

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

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 4567 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

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

   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. 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. 169–170).

For a binary variable $Y_j$ with responses 1 and 2, a logistic regression model is fitted using observations with observed values for the imputed variable $Y_j$ and its covariates $X_1$, $X_2$, ..., $X_k$:

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

where $X_1, X_2, \ldots, X_k$ are covariates for $Y_j$, $p_j = \Pr(R_j = 1 | X_1, X_2, \ldots, X_k)$, and $\text{logit}(p) = \log(p/(1-p))$.

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 $\mathbf{V}_j$.

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

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

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

   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 an observation with missing $Y_j$ and covariates $x_1, x_2, \ldots, x_k$, compute the expected probability that $Y_j = 1$:

   $$p_j = \frac{\exp(\mu_j)}{1 + \exp(\mu_j)}$$

   where $\mu_j = \beta_{*0} + \beta_{*1} x_1 + \beta_{*2} x_2 + \ldots + \beta_{*(k)} x_k$.

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

The preceding logistic regression method can be extended to include the ordinal classification variables with more than two levels of responses. The options ORDER= and DESCENDING can be used to specify the sorting order for the levels of the imputed variables.

## 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.

You can use the ORDER= option to specify the ordering of variables in the filled-in and imputation phases. The ORDER=VAR option orders the variables as specified in the VAR statement, and the default ORDER=FREQ option orders the variables by the descending frequency counts of the variables. For example, with $p$ variables in the VAR statement, the variables $Y_1$, $Y_2$, …, $Y_p$ (in that order) are used in the filled-in and imputation phases, where $Y_1$, $Y_2$, …, $Y_p$ are either the variables listed in the VAR statement (in that order) if the ORDER=VAR option is used, or the variables sorted by the descending frequency counts of the variables if the ORDER=FREQ option is used.

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,

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

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 $t+1$,

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

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 4557, "Monotone and FCS Predictive Mean Matching Methods" on page 4558, "Monotone and FCS Discriminant Function Methods" on page 4560, and "Monotone and FCS Logistic Regression Methods" on page 4562 for these methods.

The FCS method requires fewer iterations than the MCMC method (van Buuren and Oudshoorn 1999). Often, as few as five or 10 iterations are enough to produce satisfactory results (van Buuren and Oudshoorn 1999, 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 $\xi$ is a scatter plot of successive parameter estimates $\xi_i$ against the iteration number $i$. The plot provides a simple way to examine the convergence behavior of the estimation algorithm for $\xi$. 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 56.8 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 $t$th 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 $(Y_{mis}^{(1)}, \boldsymbol{\theta}^{(1)})$ , $(Y_{mis}^{(2)}, \boldsymbol{\theta}^{(2)})$ , $\ldots$, 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

$$\boldsymbol{\Sigma} = \left[ \begin{array}{cc} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}' & \boldsymbol{\Sigma}_{22} \end{array} \right]$$

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

$$\left[ \begin{array}{cc} \boldsymbol{\Sigma}_{11}^{-1} & \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12} \\ -\boldsymbol{\Sigma}_{12}'\boldsymbol{\Sigma}_{11}^{-1} & \boldsymbol{\Sigma}_{22.1} \end{array} \right]$$

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

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

and the conditional covariance matrix

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

## 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

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

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 $\boldsymbol{\mu}$

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

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

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

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

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 $\boldsymbol{\Sigma}$ from prior information for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, 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 $\boldsymbol{\Sigma}$ 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

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

### 2. An Informative Prior for $\mu$ and $\Sigma$

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

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

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 $n^* = d^* + 1$ 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 4567, is given by

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

where

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

### 3. An Informative Prior for $\Sigma$

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 $\mathbf{S}^*$ = 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 4565 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 4565. 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 4557.

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 $\sigma^2_{*j}$ are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$, $\sigma^2_j$, and $\mathbf{V}_j$. The variance is drawn as

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

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 $\sigma^2_{*j}$ 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 TIMEPLOT and ACFPLOT 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 $\chi^2$ 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 $\chi^2$ 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, **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 WLF as an ACFPLOT or TIMEPLOT option, 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 $\xi$ is a scatter plot of successive parameter estimates $\xi_i$ against the iteration number $i$. The plot provides a simple way to examine the convergence behavior of the estimation algorithm for $\xi$. 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.

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 Software: Reference* for an illustration of title options. See Example 56.11 for a usage of the trace plot.

## Autocorrelation Function Plot

To examine relationships of successive parameter estimates $\xi$, 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 $k$th order autocorrelation is computed as

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

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 $\rho_j = 0, j \geq k$.

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 Software: Reference* for a description of title options. See Example 56.8 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.

### 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’.

### 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 $\mathbf{\Sigma}$ of the form

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

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

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

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 56.12 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 56.6 summarizes the options available for OUTITER and the corresponding values for the output variable _TYPE_.

**Table 56.6** 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 4572.

## 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 $i$th 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 $\lambda$ 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 $\lambda$ (Rubin 1987, p. 114):

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

Table 56.7 shows relative efficiencies with different values of $m$ and $\lambda$.

**Table 56.7** 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 for situations with little missing information, only a small number of imputations are necessary. In practice, the number of imputations needed can be informally verified by replicating sets of $m$ imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

## 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).

## Summary of Issues in Multiple Imputation

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

### The MAR Assumption

The missing at random (MAR) assumption is needed for the imputation methods in the MI procedure. Although this assumption cannot be verified with the data, it 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).

### Number of Imputations

Based on the theory of multiple imputation, only a small number of imputations are needed for a data set with little missing information (Rubin 1987, p. 114). The number of imputations can be informally verified 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).

## 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 56.8. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

**Table 56.8** 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 | | |
| 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 logistic function | MONOTONE | PROPENSITY (/DETAILS) |
| 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, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 609 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 608 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 56.9.

**Table 56.9** 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 Laengelmavesi. 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 85, "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 56.2 with the propensity score method and in Example 56.3 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, Height, Width, and Species. The Fish2 data set is used in Example 56.4 with the logistic regression method and in Example 56.5 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 56.7 and in Example 56.8.

The Fitness1 data set created in the section “Getting Started: MI Procedure” on page 4524 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, the height, 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 (Bream and Pike)       |
| and three measurements: Length, Height, Width.                       |
| Some values have been set to missing, and the resulting data set     |
| has a monotone missing pattern in the variables                      |
| Length, Height, Width, and Species.                                  |
*----------------------------------------------------------------------*;
data Fish2;
   title 'Fish Measurement Data';
   input Species $ Length Height Width @@;
   datalines;
Bream   30.0  11.520  4.020         .   31.2  12.480  4.306
Bream   31.1  12.378  4.696     Bream   33.5  12.730  4.456
    .   34.0  12.444   .        Bream   34.7  13.602  4.927
Bream   34.5  14.180  5.279     Bream   35.0  12.670  4.690
Bream   35.1  14.005  4.844     Bream   36.2  14.227  4.959
    .   36.2  14.263   .        Bream   36.2  14.371  4.815
Bream   36.4  13.759  4.368     Bream   37.3  13.913  5.073
```

```
Bream   37.2  14.954  5.171     Bream   37.2  15.438  5.580
Bream   38.3  14.860  5.285     Bream   38.5  14.938  5.198
    .   38.6  15.633  5.134     Bream   38.7  14.474  5.728
Bream   39.5  15.129  5.570         .   39.2  15.994   .
Bream   39.7  15.523  5.280     Bream   40.6  15.469  6.131
    .   40.5    .      .        Bream   40.9  16.360  6.053
Bream   40.6  16.362  6.090     Bream   41.5  16.517  5.852
Bream   41.6  16.890  6.198     Bream   42.6  18.957  6.603
Bream   44.1  18.037  6.306     Bream   44.0  18.084  6.292
Bream   45.3  18.754  6.750     Bream   45.9  18.635  6.747
Bream   46.5  17.624  6.371
Pike    34.8   5.568  3.376     Pike    37.8   5.708  4.158
Pike    38.8   5.936  4.384         .   39.8    .      .
Pike    40.5   7.290  4.577     Pike    41.0   6.396  3.977
   .    45.5   7.280  4.323     Pike    45.5   6.825  4.459
Pike    45.8   7.786  5.130     Pike    48.0   6.960  4.896
Pike    48.7   7.792  4.870     Pike    51.2   7.680  5.376
Pike    55.1   8.926  6.171         .   59.7  10.686   .
Pike    64.0   9.600  6.144     Pike    64.0   9.600  6.144
Pike    68.0  10.812  7.480
;
```

The following statements create the Fish3 data set:

```
*-----------------------------Fish3 Data-----------------------------*
| The data set contains two species of the fish (Bream and Pike)     |
| and three measurements: Length, Height, Width.                     |
| Some values have been set to missing, and the resulting data set   |
| has an arbitrary missing pattern.                                  |
*--------------------------------------------------------------------*;
data Fish3;
   title 'Fish Measurement Data';
   input Species $ Length Height Width @@;
   datalines;
Bream   30.0  11.520  4.020         .   31.2  12.480  4.306
Bream   31.1  12.378  4.696     Bream   33.5  12.730  4.456
    .     .   12.444   .        Bream   34.7  13.602  4.927
Bream   34.5  14.180  5.279         .   35.0    .     4.690
Bream   35.1  14.005  4.844     Bream   36.2  14.227  4.959
    .   36.2  14.263   .        Bream   36.2  14.371  4.815
Bream   36.4  13.759  4.368     Bream   37.3  13.913  5.073
Bream   37.2  14.954  5.171         .   37.2    .     5.580
Bream   38.3  14.860  5.285     Bream   38.5  14.938  5.198
    .   38.6  15.633  5.134     Bream   38.7  14.474  5.728
Bream   39.5  15.129  5.570         .   39.2  15.994   .
Bream   39.7  15.523  5.280     Bream   40.6  15.469  6.131
    .   40.5    .      .        Bream   40.9  16.360  6.053
Bream   40.6  16.362  6.090     Bream   41.5  16.517  5.852
Bream   41.6  16.890  6.198     Bream   42.6  18.957  6.603
Bream     .   18.037   .        Bream     .   18.084  6.292
Bream   45.3  18.754  6.750     Bream   45.9  18.635  6.747
Bream   46.5  17.624  6.371
Pike    34.8   5.568  3.376     Pike    37.8   5.708  4.158
Pike    38.8   5.936  4.384         .   39.8    .      .
```

```
Pike    40.5   7.290  4.577     Pike    41.0   6.396  3.977
   .    45.5   7.280  4.323     Pike    45.5   6.825  4.459
Pike    45.8   7.786  5.130     Pike    48.0   6.960  4.896
Pike    48.7   7.792  4.870     Pike    51.2   7.680  5.376
Pike    55.1   8.926  6.171        .    59.7  10.686   .
Pike    64.0   9.600  6.144     Pike    64.0   9.600  6.144
Pike      .   10.812  7.480
;
```

## Example 56.1: 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 56.1.1 describes the method and options used in the procedure if a positive number is specified in the NIMPUTE= option.

**Output 56.1.1** 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 56.1.2 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 56.1.2** Missing Data Patterns

Missing Data Patterns

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | . | 4 | 12.90 |
| 3 | X | . | . | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | . | 2 | 6.45 |

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Oxygen | RunTime | RunPulse |
| 1 | 46.353810 | 10.809524 | 171.666667 |
| 2 | 47.109500 | 10.137500 | . |
| 3 | 52.461667 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 9.885000 | . |

With the SIMPLE option, the procedure displays simple descriptive univariate statistics for available cases in the "Univariate Statistics" table in Output 56.1.3 and correlations from pairwise available cases in the "Pairwise Correlations" table in Output 56.1.4.

**Output 56.1.3** Univariate Statistics

Univariate Statistics

| Variable | N | Mean | Std Dev | Minimum | Maximum |
|---|---|---|---|---|---|
| Oxygen | 28 | 47.11618 | 5.41305 | 37.38800 | 60.05500 |
| RunTime | 28 | 10.68821 | 1.37988 | 8.63000 | 14.03000 |
| RunPulse | 22 | 171.86364 | 10.14324 | 148.00000 | 186.00000 |

Univariate Statistics

| | Missing Values | |
|---|---|---|
| Variable | Count | Percent |
| Oxygen | 3 | 9.68 |
| RunTime | 3 | 9.68 |
| RunPulse | 9 | 29.03 |

**Output 56.1.4** 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 56.1.5.

**Output 56.1.5** 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 56.1.6 displays the iteration history for the EM algorithm.

**Output 56.1.6** 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 56.1.7 displays the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of a multivariate normal distribution from the data set Fitness1.

**Output 56.1.7** 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 56.1.8 is a TYPE=COV data set. The observation with _TYPE_='MEAN' contains the MLE for the parameter $\mu$, and the observations with _TYPE_='COV' contain the MLE for the parameter $\Sigma$ of a multivariate normal distribution from the data set Fitness1.

**Output 56.1.8** 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 56.2: 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 56.2.1 describes the method and options used in the multiple imputation process. By default, five imputations are created for the missing data.

**Output 56.2.1** Model Information

```
                          The MI Procedure

                         Model Information

     Data Set                              WORK.FISH1
     Method                                Monotone
     Number of Imputations                 5
     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 56.2.2 displays the detailed model specification. By default, the observations are sorted into five groups based on their propensity scores.

**Output 56.2.2** Monotone Model Specification

```
                    Monotone Model Specification

                                          Imputed
          Method                          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 56.2.3 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 56.2.3** Missing Data Patterns

Missing Data Patterns

| Group | Length1 | Length2 | Length3 | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 30 | 85.71 |
| 2 | X | X | . | 3 | 8.57 |
| 3 | X | . | . | 2 | 5.71 |

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Length1 | Length2 | Length3 |
| 1 | 30.603333 | 33.436667 | 38.720000 |
| 2 | 29.033333 | 31.666667 | . |
| 3 | 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 56.2.4 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 4578.

**Output 56.2.4** Variance Information

**Variance Information**

| Variable | Variance: Between | Variance: Within | Variance: Total | DF |
|---|---|---|---|---|
| Length2 | 0.001500 | 0.465422 | 0.467223 | 32.034 |
| Length3 | 0.049725 | 0.547434 | 0.607104 | 27.103 |

**Variance Information**

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Length2 | 0.003869 | 0.003861 | 0.999228 |
| Length3 | 0.108999 | 0.102610 | 0.979891 |

The "Parameter Estimates" table in Output 56.2.5 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 zero by default.

**Output 56.2.5** Parameter Estimates

**Parameter Estimates**

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Length2 | 33.006857 | 0.683537 | 31.61460 | 34.39912 | 32.034 |
| Length3 | 38.361714 | 0.779169 | 36.76328 | 39.96015 | 27.103 |

**Parameter Estimates**

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Length2 | 32.957143 | 33.060000 | 0 | 48.29 | <.0001 |
| Length3 | 38.080000 | 38.545714 | 0 | 49.23 | <.0001 |

The following statements list the first 10 observations of the data set outex2, as shown in Output 56.2.6. 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 56.2.6** 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 56.3: 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 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 56.2.3 in Example 56.2.

The “Monotone Model Specification” table in Output 56.3.1 displays the model specification.

**Output 56.3.1** 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 56.3.2 and Output 56.3.3.

**Output 56.3.2** Regression Model

Regression Models for Monotone Method

| Imputed Variable | Effect | Obs-Data | Imputation 1 | Imputation 2 | Imputation 3 |
|---|---|---|---|---|---|
| Length2 | Intercept | -0.04249 | -0.049184 | -0.055470 | -0.051346 |
| Length2 | Length1 | 0.98587 | 1.001934 | 0.995275 | 0.992294 |

Regression Models for Monotone Method

| Imputed Variable | Effect | Imputation 4 | Imputation 5 |
|---|---|---|---|
| Length2 | Intercept | -0.064193 | -0.030719 |
| Length2 | Length1 | 0.983122 | 0.995883 |

**Output 56.3.3** Regression Predicted Mean Matching Model

Regression Models for Monotone Predicted Mean Matching Method

| Imputed Variable | Effect | Obs Data | Imputation 1 | Imputation 2 | Imputation 3 |
|---|---|---|---|---|---|
| Length3 | Intercept | -0.01304 | 0.004134 | -0.011417 | -0.034177 |
| Length3 | Length1 | -0.01332 | 0.025320 | -0.037494 | 0.308765 |
| Length3 | Length2 | 0.98918 | 0.955510 | 1.025741 | 0.673374 |
| Length3 | Length1*Length2 | -0.02521 | -0.034964 | -0.022017 | -0.017919 |

Regression Models for Monotone Predicted Mean Matching Method

| Imputed Variable | Effect | Imputation 4 | Imputation 5 |
|---|---|---|---|
| Length3 | Intercept | -0.010532 | 0.004685 |
| Length3 | Length1 | 0.156606 | -0.147118 |
| Length3 | Length2 | 0.828384 | 1.146440 |
| Length3 | Length1*Length2 | -0.029335 | -0.034671 |

After the completion of five imputations by default, the "Variance Information" table in Output 56.3.4 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 4578.

**Output 56.3.4** Variance Information

Variance Information

| Variable | Variance Between | Variance Within | Variance Total | DF |
|---|---|---|---|---|
| Length2 | 0.000133 | 0.439512 | 0.439672 | 32.15 |
| Length3 | 0.000386 | 0.486913 | 0.487376 | 32.131 |

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Length2 | 0.000363 | 0.000363 | 0.999927 |
| Length3 | 0.000952 | 0.000951 | 0.999810 |

The "Parameter Estimates" table in Output 56.3.5 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 56.3.5** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Length2 | 33.104571 | 0.663078 | 31.75417 | 34.45497 | 32.15 |
| Length3 | 38.424571 | 0.698123 | 37.00277 | 39.84637 | 32.131 |

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Length2 | 33.088571 | 33.117143 | 35.000000 | -2.86 | 0.0074 |
| Length3 | 38.397143 | 38.445714 | 45.000000 | -9.42 | <.0001 |

The following statements list the first 10 observations of the data set outex3 in Output 56.3.6. 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 56.3.6** 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 56.4: 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 out=outex4;
   class Species;
   monotone reg( Length Width/ details)
            logistic( Species= Length Height Width Height*Width/ details);
   var Length Height Width Species;
run;
```

The "Model Information" table in Output 56.4.1 describes the method and options used in the multiple imputation process.

**Output 56.4.1** Model Information

```
                              The MI Procedure

                             Model Information

          Data Set                             WORK.FISH2
          Method                               Monotone
          Number of Imputations                5
          Seed for random number generator     1305417
```

The "Monotone Model Specification" table in Output 56.4.2 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 56.4.2** Monotone Model Specification

```
                         Monotone Model Specification

                                              Imputed
                      Method                  Variables

                      Regression              Height Width
                      Logistic Regression     Species
```

The "Missing Data Patterns" table in Output 56.4.3 lists distinct missing data patterns with corresponding frequencies and percentages. The table confirms a monotone missing pattern for these variables.

**Output 56.4.3** Missing Data Patterns

Missing Data Patterns

| Group | Length | Height | Width | Species | Freq | Percent |
|---|---|---|---|---|---|---|
| 1 | X | X | X | X | 43 | 82.69 |
| 2 | X | X | X | . | 3 | 5.77 |
| 3 | X | X | . | . | 4 | 7.69 |
| 4 | X | . | . | . | 2 | 3.85 |

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Length | Height | Width |
| 1 | 41.997674 | 12.819512 | 5.359860 |
| 2 | 38.433333 | 11.797667 | 4.587667 |
| 3 | 42.275000 | 13.346750 | . |
| 4 | 40.150000 | . | . |

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 56.4.4.

**Output 56.4.4** Regression Model

Regression Models for Monotone Method

| Imputed | | | Imputation | | |
|---|---|---|---|---|---|
| Variable | Effect | Obs-Data | 1 | 2 | 3 |
| Width | Intercept | 0.00682 | 0.054140 | 0.018049 | -0.015137 |
| Width | Length | 0.75519 | 0.838485 | 0.768945 | 0.789577 |
| Width | Height | 0.73890 | 0.832117 | 0.831748 | 0.809482 |

Regression Models for Monotone Method

| Imputed | | Imputation | |
|---|---|---|---|
| Variable | Effect | 4 | 5 |
| Width | Intercept | 0.024027 | 0.084643 |
| Width | Length | 0.728779 | 0.631217 |
| Width | Height | 0.747734 | 0.745232 |

**Output 56.4.5** Logistic Regression Model

Logistic Models for Monotone Method

| Imputed Variable | Effect | Obs-Data | Imputation 1 | Imputation 2 | Imputation 3 |
|---|---|---|---|---|---|
| Species | Intercept | 22.80713 | 22.807129 | 22.807129 | 22.807129 |
| Species | Length | -14.44698 | -14.446980 | -14.446980 | -14.446980 |
| Species | Height | 43.11236 | 43.112363 | 43.112363 | 43.112363 |
| Species | Width | -9.64352 | -9.643524 | -9.643524 | -9.643524 |
| Species | Height*Width | -9.73015 | -9.730154 | -9.730154 | -9.730154 |

Logistic Models for Monotone Method

| Imputed Variable | Effect | Imputation 4 | Imputation 5 |
|---|---|---|---|
| Species | Intercept | 22.807129 | 22.807129 |
| Species | Length | -14.446980 | -14.446980 |
| Species | Height | 43.112363 | 43.112363 |
| Species | Width | -9.643524 | -9.643524 |
| Species | Height*Width | -9.730154 | -9.730154 |

The following statements list the first 10 observations of the data set outex4 in Output 56.4.5:

```
proc print data=outex4(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

**Output 56.4.6** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Species | Length | Height | Width |
|---|---|---|---|---|---|
| 1 | 1 | Bream | 30.0 | 11.520 | 4.02000 |
| 2 | 1 | Bream | 31.2 | 12.480 | 4.30600 |
| 3 | 1 | Bream | 31.1 | 12.378 | 4.69600 |
| 4 | 1 | Bream | 33.5 | 12.730 | 4.45600 |
| 5 | 1 | Bream | 34.0 | 12.444 | 4.62964 |
| 6 | 1 | Bream | 34.7 | 13.602 | 4.92700 |
| 7 | 1 | Bream | 34.5 | 14.180 | 5.27900 |
| 8 | 1 | Bream | 35.0 | 12.670 | 4.69000 |
| 9 | 1 | Bream | 35.1 | 14.005 | 4.84400 |
| 10 | 1 | Bream | 36.2 | 14.227 | 4.95900 |

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 56.5: 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 nimpute=3 out=outex5;
   class Species;
   monotone reg( Height Width)
            discrim( Species= Length Height Width/ details);
   var Length Height Width Species;
run;
```

The "Model Information" table in Output 56.5.1 describes the method and options used in the multiple imputation process.

**Output 56.5.1** Model Information

```
                          The MI Procedure

                         Model Information

          Data Set                             WORK.FISH2
          Method                               Monotone
          Number of Imputations                3
          Seed for random number generator     7545417
```

The "Monotone Model Specification" table in Output 56.5.2 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 56.5.2** Monotone Model Specification

```
                    Monotone Model Specification

                                         Imputed
                   Method                Variables

                   Regression            Height Width
                   Discriminant Function Species
```

The "Missing Data Patterns" table in Output 56.5.3 lists distinct missing data patterns with corresponding frequencies and percentages. The table confirms a monotone missing pattern for these variables.

**Output 56.5.3** Missing Data Patterns

Missing Data Patterns

| Group | Length | Height | Width | Species | Freq | Percent |
|---|---|---|---|---|---|---|
| 1 | X | X | X | X | 43 | 82.69 |
| 2 | X | X | X | . | 3 | 5.77 |
| 3 | X | X | . | . | 4 | 7.69 |
| 4 | X | . | . | . | 2 | 3.85 |

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Length | Height | Width |
| 1 | 41.997674 | 12.819512 | 5.359860 |
| 2 | 38.433333 | 11.797667 | 4.587667 |
| 3 | 42.275000 | 13.346750 | . |
| 4 | 40.150000 | . | . |

When you use the DETAILS option, the parameters estimated from the observed data and the parameters used in each imputation are displayed in Output 56.5.4.

**Output 56.5.4** Discriminant Model

Group Means for Monotone Discriminant Method

| | | | Imputation | | |
|---|---|---|---|---|---|
| Species | Variable | Obs-Data | 1 | 2 | 3 |
| Bream | Length | -0.36712 | -0.198907 | -0.375696 | -0.307771 |
| Bream | Height | 0.64051 | 0.756448 | 0.684845 | 0.658337 |
| Bream | Width | 0.20882 | 0.465034 | 0.254438 | 0.252637 |
| Pike | Length | 0.85554 | 0.656521 | 0.677957 | 1.024069 |
| Pike | Height | -1.31185 | -1.431954 | -1.436355 | -1.119520 |
| Pike | Width | -0.25768 | -0.381503 | -0.420441 | -0.136188 |

The following statements list the first 10 observations of the data set outex5 in Output 56.5.5. 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 56.5.5** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Species | Length | Height | Width |
|---|---|---|---|---|---|
| 1 | 1 | Bream | 30.0 | 11.520 | 4.02000 |
| 2 | 1 | Bream | 31.2 | 12.480 | 4.30600 |
| 3 | 1 | Bream | 31.1 | 12.378 | 4.69600 |
| 4 | 1 | Bream | 33.5 | 12.730 | 4.45600 |
| 5 | 1 | Bream | 34.0 | 12.444 | 4.46687 |
| 6 | 1 | Bream | 34.7 | 13.602 | 4.92700 |
| 7 | 1 | Bream | 34.5 | 14.180 | 5.27900 |
| 8 | 1 | Bream | 35.0 | 12.670 | 4.69000 |
| 9 | 1 | Bream | 35.1 | 14.005 | 4.84400 |
| 10 | 1 | Bream | 36.2 | 14.227 | 4.95900 |

## Example 56.6: FCS Method for Continuous Variables

This example uses FCS regression methods to impute values for all continuous variables in a data set with 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=4 mu0=50 10 180 out=outex6;
   fcs nbiter=10 reg(/details);
   var Oxygen RunTime RunPulse;
run;
```

The NIMPUTE=4 option specifies the total number of imputations. The FCS statement requests multivariate imputations by FCS methods, and the NBITER=10 option (which is the default) specifies the number of burn-in iterations before each imputation.

The "Model Information" table in Output 56.6.1 describes the method and options used in the multiple imputation process.

**Output 56.6.1** Model Information

The MI Procedure

Model Information

| | |
|---|---|
| Data Set | WORK.FITNESS1 |
| Method | FCS |
| Number of Imputations | 4 |
| Number of Burn-in Iterations | 10 |
| Seed for random number generator | 1213 |

The "FCS Model Specification" table in Output 56.6.2 describes methods and imputed variables in the imputation model. With the REG option in the FCS statement, the procedure uses the regression method to impute variables RunTime, RunPulse, and Oxygen in the model.

**Output 56.6.2** FCS Model Specification

FCS Model Specification

| Method | Imputed Variables |
|---|---|
| Regression | Oxygen RunTime RunPulse |

The "Missing Data Patterns" table in Output 56.6.3 lists distinct missing data patterns with corresponding frequencies and percentages. With the default ORDER=FREQ option, variables are ordered by the descending frequency counts for the missing values in the filled-in and imputation phases.

**Output 56.6.3** Missing Data Patterns

Missing Data Patterns

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | . | 4 | 12.90 |
| 3 | X | . | . | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | . | 2 | 6.45 |

Missing Data Patterns

| Group | Group Means Oxygen | RunTime | RunPulse |
|---|---|---|---|
| 1 | 46.353810 | 10.809524 | 171.666667 |
| 2 | 47.109500 | 10.137500 | . |
| 3 | 52.461667 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 9.885000 | . |

When you use the DETAILS option, the parameters used in each imputation are displayed in Output 56.6.4, Output 56.6.5, and Output 56.6.6.

**Output 56.6.4** FCS Regression Model for Oxygen

Regression Models for FCS Method

| Imputed Variable | Effect | Imputation 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Oxygen | Intercept | -0.000578 | -0.040829 | -0.100644 | 0.200243 |
| Oxygen | RunTime | -0.706222 | -0.588050 | -0.732917 | -0.539925 |
| Oxygen | RunPulse | -0.163355 | -0.211405 | -0.393984 | -0.156234 |

**Output 56.6.5** FCS Regression Model for RunTime

The MI Procedure

Regression Models for FCS Method

| Imputed Variable | Effect | Imputation 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| RunTime | Intercept | -0.174786 | 0.145997 | -0.240973 | -0.291107 |
| RunTime | Oxygen | -0.876802 | -0.630979 | -0.982318 | -0.879243 |
| RunTime | RunPulse | -0.084348 | -0.055832 | -0.231270 | -0.133229 |

**Output 56.6.6** FCS Regression Model for RunPulse

The MI Procedure

Regression Models for FCS Method

| Imputed Variable | Effect | Imputation 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| RunPulse | Intercept | -0.162535 | -0.598755 | 0.078072 | -0.097289 |
| RunPulse | Oxygen | -0.804417 | -0.544019 | -0.032744 | -0.335796 |
| RunPulse | RunTime | -0.057307 | 0.215520 | 0.313246 | 0.146078 |

The following statements list the first 10 observations of the data set outex6 in Output 56.6.7. 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 56.6.7** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Oxygen | RunTime | Run Pulse |
|---|---|---|---|---|
| 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 | 7.7722 | 155.233 |
| 5 | 1 | 49.8740 | 9.2200 | 153.146 |
| 6 | 1 | 44.8110 | 11.6300 | 176.000 |
| 7 | 1 | 45.3406 | 11.9500 | 176.000 |
| 8 | 1 | 36.6027 | 10.8500 | 175.250 |
| 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 56.6.8 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 4578.

**Output 56.6.8** Variance Information

Variance Information

| | Variance | | | |
|---|---|---|---|---|
| Variable | Between | Within | Total | DF |
| Oxygen | 0.078728 | 0.975510 | 1.073920 | 23.888 |
| RunTime | 0.001464 | 0.071174 | 0.073003 | 27.318 |
| RunPulse | 1.469522 | 3.666764 | 5.503667 | 11.063 |

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Oxygen | 0.100880 | 0.096679 | 0.976401 |
| RunTime | 0.025709 | 0.025473 | 0.993672 |
| RunPulse | 0.500960 | 0.378278 | 0.913601 |

The “Parameter Estimates” table in Output 56.6.9 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 56.6.9** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Oxygen | 47.032052 | 1.036301 | 44.8927 | 49.1714 | 23.888 |
| RunTime | 10.494632 | 0.270192 | 9.9405 | 11.0487 | 27.318 |
| RunPulse | 169.709378 | 2.345990 | 164.5495 | 174.8693 | 11.063 |

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Oxygen | 46.771075 | 47.346642 | 50.000000 | -2.86 | 0.0086 |
| RunTime | 10.453740 | 10.544396 | 10.000000 | 1.83 | 0.0781 |
| RunPulse | 168.550372 | 170.921431 | 180.000000 | -4.39 | 0.0011 |

## Example 56.7: 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 out=outex7;
   class Species;
   fcs nbiter=5 discrim(Species/details) reg(Height/details);
   var Species Length Height 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 56.7.1 describes the method and options used in the multiple imputation process.

**Output 56.7.1** Model Information

The MI Procedure

Model Information

| | |
|---|---|
| Data Set | WORK.FISH3 |
| Method | FCS |
| Number of Imputations | 5 |
| Number of Burn-in Iterations | 5 |
| Seed for random number generator | 1305417 |

The "FCS Model Specification" table in Output 56.7.2 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 56.7.2** FCS Model Specification

FCS Model Specification

| Method | Imputed Variables |
|---|---|
| Regression | Length Height Width |
| Discriminant Function | Species |

The "Missing Data Patterns" table in Output 56.7.3 lists distinct missing data patterns with corresponding frequencies and percentages. With the default ORDER=FREQ option, the variable ordering by the descending frequency counts is used for the missing values in the filled-in and imputation phases.

**Output 56.7.3** Missing Data Patterns

Missing Data Patterns

| Group | Length | Height | Width | Species | Freq | Percent |
|---|---|---|---|---|---|---|
| 1 | X | X | X | X | 38 | 73.08 |
| 2 | X | X | X | . | 3 | 5.77 |
| 3 | X | X | . | . | 3 | 5.77 |
| 4 | X | . | X | . | 2 | 3.85 |
| 5 | X | . | . | . | 2 | 3.85 |
| 6 | . | X | X | X | 2 | 3.85 |
| 7 | . | X | . | X | 1 | 1.92 |
| 8 | . | X | . | . | 1 | 1.92 |

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Length | Height | Width |
| 1 | 41.515789 | 12.531526 | 5.266474 |
| 2 | 38.433333 | 11.797667 | 4.587667 |
| 3 | 45.033333 | 13.647667 | . |
| 4 | 36.100000 | . | 5.135000 |
| 5 | 40.150000 | . | . |
| 6 | . | 14.448000 | 6.886000 |
| 7 | . | 18.037000 | . |
| 8 | . | 12.444000 | . |

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 56.7.4 and in the “Regression Models for FCS Method” table in Output 56.7.5.

**Output 56.7.4** FCS Discrim Model for Species

Group Means for FCS Discriminant Method

| Species | Variable | Imputation 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Bream | Length | -0.020460 | -0.375046 | -0.455147 | -0.227513 |
| Bream | Height | 0.693833 | 0.623187 | 0.744749 | 0.580846 |
| Bream | Width | 0.397506 | 0.173774 | 0.421867 | 0.167947 |
| Pike | Length | 0.845745 | 1.304043 | 0.708257 | 1.063104 |
| Pike | Height | -1.357333 | -1.140244 | -1.367343 | -1.269584 |
| Pike | Width | -0.341246 | 0.193092 | -0.517978 | -0.366050 |

Group Means for FCS Discriminant Method

| Species | Variable | Imputation 5 |
|---|---|---|
| Bream | Length | -0.149084 |
| Bream | Height | 0.714942 |
| Bream | Width | 0.300103 |
| Pike | Length | 0.382590 |
| Pike | Height | -1.342550 |
| Pike | Width | -0.438790 |

**Output 56.7.5** FCS Regression Model for Height

Regression Models for FCS Method

| Imputed Variable | Effect | Species | Imputation 1 | 2 | 3 |
|---|---|---|---|---|---|
| Height | Intercept | | -0.341941 | -0.366473 | -0.315587 |
| Height | Length | | 0.119780 | 0.126889 | 0.011333 |
| Height | Width | | 0.350410 | 0.310695 | 0.441925 |
| Height | Species | Bream | 0.987346 | 1.008808 | 0.851794 |

Regression Models for FCS Method

| Imputed Variable | Effect | Imputation 4 | 5 |
|---|---|---|---|
| Height | Intercept | -0.361090 | -0.324455 |
| Height | Length | 0.137968 | 0.117460 |
| Height | Width | 0.345254 | 0.317621 |
| Height | Species | 0.999192 | 0.999200 |

The following statements list the first 10 observations of the data set outex7 in Output 56.7.6:

```
proc print data=outex7(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

**Output 56.7.6** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Species | Length | Height | Width |
|---|---|---|---|---|---|
| 1 | 1 | Bream | 30.0000 | 11.5200 | 4.02000 |
| 2 | 1 | Bream | 31.2000 | 12.4800 | 4.30600 |
| 3 | 1 | Bream | 31.1000 | 12.3780 | 4.69600 |
| 4 | 1 | Bream | 33.5000 | 12.7300 | 4.45600 |
| 5 | 1 | Bream | 31.2895 | 12.4440 | 4.05416 |
| 6 | 1 | Bream | 34.7000 | 13.6020 | 4.92700 |
| 7 | 1 | Bream | 34.5000 | 14.1800 | 5.27900 |
| 8 | 1 | Bream | 35.0000 | 13.2992 | 4.69000 |
| 9 | 1 | Bream | 35.1000 | 14.0050 | 4.84400 |
| 10 | 1 | Bream | 36.2000 | 14.2270 | 4.95900 |

After the completion of five imputations by default, the "Variance Information" table in Output 56.7.7 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 4578.

**Output 56.7.7** Variance Information

Variance Information

| | Variance | | | |
|---|---|---|---|---|
| Variable | Between | Within | Total | DF |
| Length | 0.158766 | 1.287899 | 1.478418 | 36.33 |
| Height | 0.007807 | 0.310949 | 0.320317 | 47.194 |
| Width | 0.002160 | 0.016085 | 0.018677 | 35.138 |

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Length | 0.147930 | 0.136011 | 0.973518 |
| Height | 0.030127 | 0.029661 | 0.994103 |
| Width | 0.161157 | 0.146966 | 0.971446 |

The "Parameter Estimates" table in Output 56.7.8 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 56.7.8** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Length | 41.858477 | 1.215902 | 39.39329 | 44.32366 | 36.33 |
| Height | 12.724307 | 0.565966 | 11.58585 | 13.86276 | 47.194 |
| Width | 5.344556 | 0.136663 | 5.06715 | 5.62196 | 35.138 |

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Length | 41.511771 | 42.316960 | 0 | 34.43 | <.0001 |
| Height | 12.622320 | 12.811756 | 0 | 22.48 | <.0001 |
| Width | 5.290049 | 5.393757 | 0 | 39.11 | <.0001 |

## Example 56.8: 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 out=outex8;
   class Species;
   fcs plots=trace
       logistic(Species= Height Width Height*Width /details);
   var Species Height Width;
run;
ods graphics off;
```

The “Model Information” table in Output 56.8.1 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 56.8.1** Model Information

The MI Procedure

Model Information

| | |
|---|---|
| Data Set | WORK.FISH3 |
| Method | FCS |
| Number of Imputations | 5 |
| Number of Burn-in Iterations | 10 |
| Seed for random number generator | 1305417 |

The “FCS Model Specification” table in Output 56.8.2 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 56.8.2** FCS Model Specification

FCS Model Specification

| Method | Imputed Variables |
|---|---|
| Regression | Height Width |
| Logistic Regression | Species |

The “Missing Data Patterns” table in Output 56.8.3 lists distinct missing data patterns with corresponding frequencies and percentages. With the default ORDER=FREQ option, variables are ordered by the descending frequency counts for the missing values in the filled-in and imputation phases.

**Output 56.8.3** Missing Data Patterns

Missing Data Patterns

| | | | | | | --------Group Means------- | |
|---|---|---|---|---|---|---|---|
| Group | Height | Width | Species | Freq | Percent | Height | Width |
| 1 | X | X | X | 40 | 76.92 | 12.627350 | 5.347450 |
| 2 | X | X | . | 3 | 5.77 | 11.797667 | 4.587667 |
| 3 | X | . | X | 1 | 1.92 | 18.037000 | . |
| 4 | X | . | . | 4 | 7.69 | 13.346750 | . |
| 5 | . | X | . | 2 | 3.85 | . | 5.135000 |
| 6 | O | O | O | 2 | 3.85 | . | . |

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 56.8.4.

**Output 56.8.4** FCS Logistic Regression Model for Species

**Logistic Models for FCS Method**

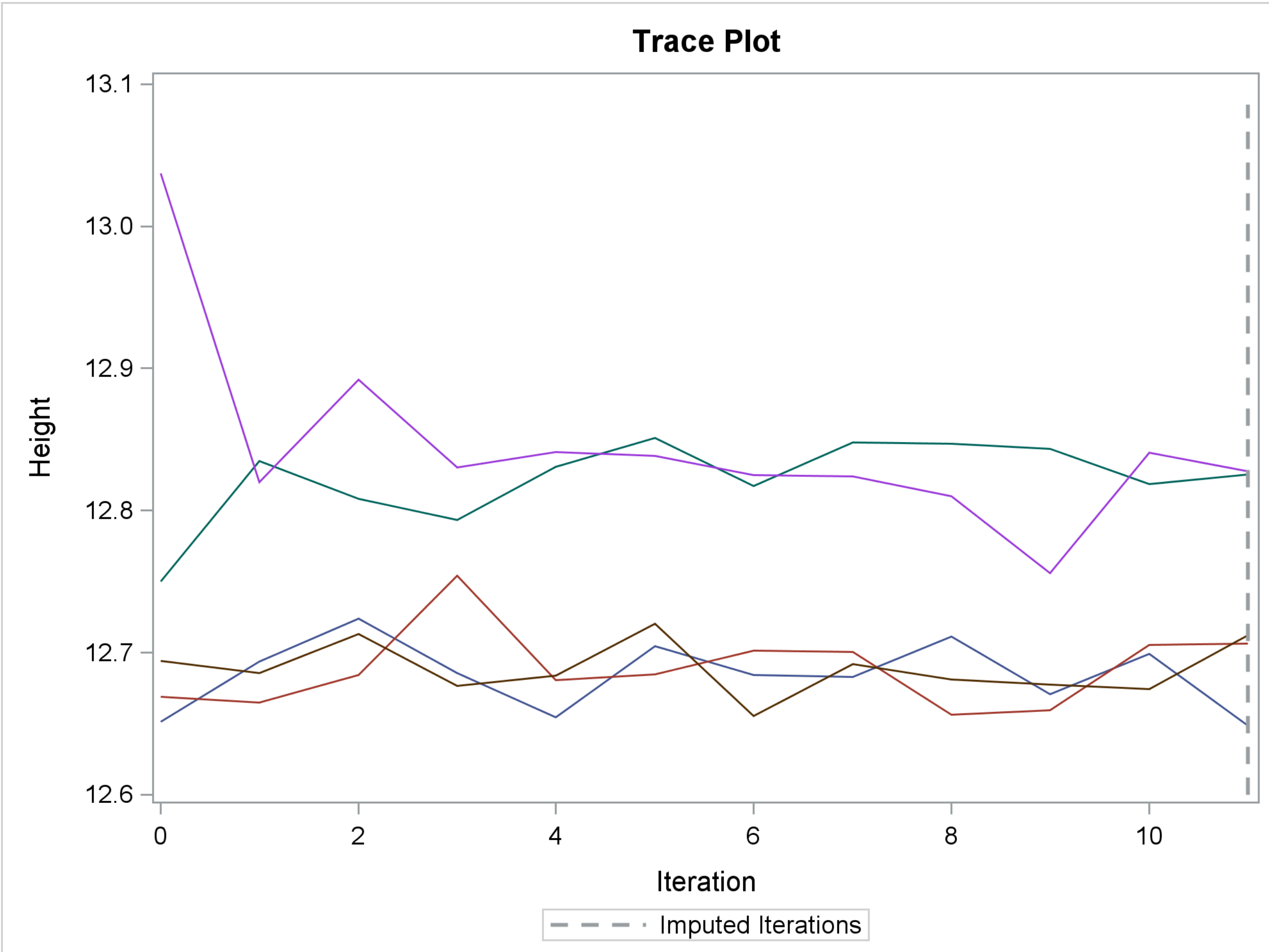
| Imputed Variable | Effect | Imputation 1 | Imputation 2 | Imputation 3 | Imputation 4 |
|---|---|---|---|---|---|
| Species | Intercept | 27.019602 | 27.064278 | 27.262198 | 27.214159 |
| Species | Height | 60.068695 | 60.007370 | 59.980982 | 59.933904 |
| Species | Width | -25.537953 | -25.661405 | -26.044380 | -25.987921 |
| Species | Height*Width | -5.479559 | -5.839848 | -6.786713 | -6.691049 |

**Logistic Models for FCS Method**

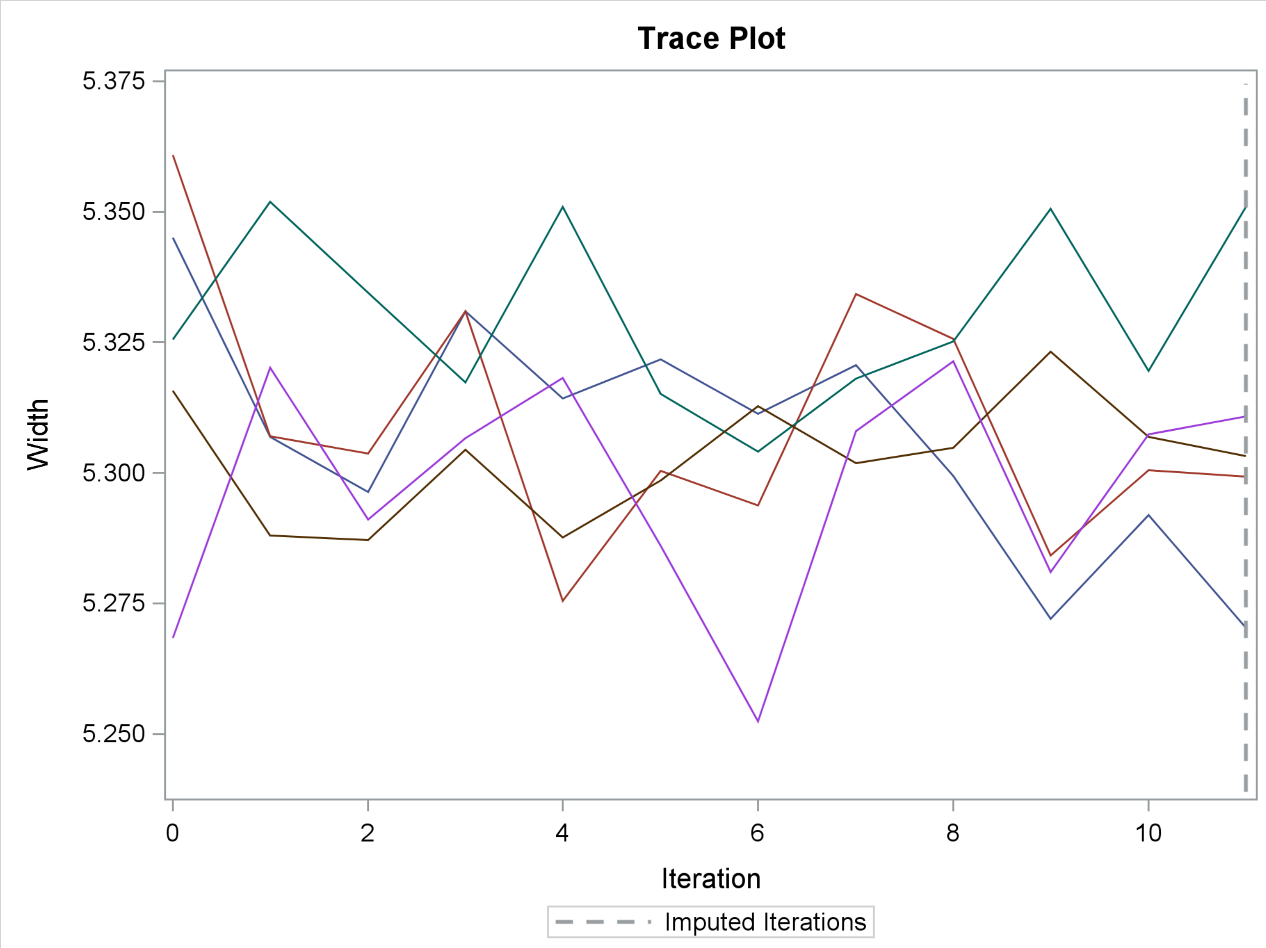
| Imputed Variable | Effect | Imputation 5 |
|---|---|---|
| Species | Intercept | 27.727730 |
| Species | Height | 61.324682 |
| Species | Width | -23.681898 |
| Species | Height*Width | -2.690170 |

With ODS Graphics enabled, the PLOTS=TRACE option displays trace plots of means for all continuous variables by default, as shown in Output 56.8.5 and Output 56.8.6. 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 56.8.5** Trace Plot for Height



**Output 56.8.6** Trace Plot for Width




The following statements list the first 10 observations of the data set outex8 in Output 56.8.7:

```
proc print data=outex8(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

**Output 56.8.7** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Species | Length | Height | Width |
|---|---|---|---|---|---|
| 1 | 1 | Bream | 30.0000 | 11.5200 | 4.02000 |
| 2 | 1 | Bream | 31.2000 | 12.4800 | 4.30600 |
| 3 | 1 | Bream | 31.1000 | 12.3780 | 4.69600 |
| 4 | 1 | Bream | 33.5000 | 12.7300 | 4.45600 |
| 5 | 1 | Bream | 23.9427 | 12.4440 | 3.35343 |
| 6 | 1 | Bream | 34.7000 | 13.6020 | 4.92700 |
| 7 | 1 | Bream | 34.5000 | 14.1800 | 5.27900 |
| 8 | 1 | Bream | 35.0000 | 14.8409 | 4.69000 |
| 9 | 1 | Bream | 35.1000 | 14.0050 | 4.84400 |
| 10 | 1 | Bream | 36.2000 | 14.2270 | 4.95900 |

After the completion of five imputations by default, the "Variance Information" table in Output 56.8.8 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 4578.

**Output 56.8.8** Variance Information

Variance Information

| Variable | Variance Between | Variance Within | Variance Total | DF |
|---|---|---|---|---|
| Height | 0.006302 | 0.313539 | 0.321101 | 45.714 |
| Width | 0.001343 | 0.017068 | 0.018680 | 39.861 |

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Height | 0.024119 | 0.023821 | 0.995258 |
| Width | 0.094387 | 0.089626 | 0.982390 |

The "Parameter Estimates" table in Output 56.8.9 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 56.8.9** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Height | 12.744021 | 0.566658 | 11.60321 | 13.88484 | 45.714 |
| Width | 5.303250 | 0.136673 | 5.02699 | 5.57951 | 39.861 |

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Height | 12.648427 | 12.827767 | 0 | 22.49 | <.0001 |
| Width | 5.256781 | 5.341640 | 0 | 38.80 | <.0001 |

## Example 56.9: 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=6 mu0=50 10 180 ;
   mcmc chain=multiple displayinit initial=em(itprint);
   var Oxygen RunTime RunPulse;
run;
```

The “Model Information” table in Output 56.9.1 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 56.9.1** 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 | 6 |
| 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 56.9.2 lists distinct missing data patterns with corresponding statistics.

**Output 56.9.2** Missing Data Patterns

Missing Data Patterns

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | . | 4 | 12.90 |
| 3 | X | . | . | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | . | 2 | 6.45 |

Missing Data Patterns

| Group | Group Means Oxygen | RunTime | RunPulse |
|---|---|---|---|
| 1 | 46.353810 | 10.809524 | 171.666667 |
| 2 | 47.109500 | 10.137500 | . |
| 3 | 52.461667 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 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 56.9.3.

**Output 56.9.3** EM (Posterior Mode) Iteration History

EM (Posterior Mode) Iteration History

| _Iteration_ | -2 Log L | -2 Log Posterior | Oxygen | RunTime |
|---|---|---|---|---|
| 0 | 254.482800 | 282.909549 | 47.104077 | 10.554858 |
| 1 | 255.081168 | 282.051584 | 47.104077 | 10.554857 |
| 2 | 255.271408 | 282.017488 | 47.104077 | 10.554857 |
| 3 | 255.318622 | 282.015372 | 47.104002 | 10.554523 |
| 4 | 255.330259 | 282.015232 | 47.103861 | 10.554388 |
| 5 | 255.333161 | 282.015222 | 47.103797 | 10.554341 |
| 6 | 255.333896 | 282.015222 | 47.103774 | 10.554325 |
| 7 | 255.334085 | 282.015222 | 47.103766 | 10.554320 |

EM (Posterior Mode) Iteration History

| _Iteration_ | RunPulse |
|---|---|
| 0 | 171.381669 |
| 1 | 171.381652 |
| 2 | 171.381644 |
| 3 | 171.381842 |
| 4 | 171.382053 |
| 5 | 171.382150 |
| 6 | 171.382185 |
| 7 | 171.382196 |

When you use the DISPLAYINIT option in the MCMC statement, the "Initial Parameter Estimates for MCMC" table in Output 56.9.4 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 56.9.4** 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 56.9.5 and Output 56.9.6 display variance information and parameter estimates, respectively, from the multiple imputation.

**Output 56.9.5** Variance Information

Variance Information

| | --------Variance-------- | | | |
|---|---|---|---|---|
| Variable | Between | Within | Total | DF |
| Oxygen | 0.051560 | 0.928170 | 0.988323 | 25.958 |
| RunTime | 0.003979 | 0.070057 | 0.074699 | 25.902 |
| RunPulse | 4.118578 | 4.260631 | 9.065638 | 7.5938 |

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| Oxygen | 0.064809 | 0.062253 | 0.989731 |
| RunTime | 0.066262 | 0.063589 | 0.989513 |
| RunPulse | 1.127769 | 0.575218 | 0.912517 |

**Output 56.9.6** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| Oxygen | 47.164819 | 0.994145 | 45.1212 | 49.2085 | 25.958 |
| RunTime | 10.549936 | 0.273312 | 9.9880 | 11.1118 | 25.902 |
| RunPulse | 170.969836 | 3.010920 | 163.9615 | 177.9782 | 7.5938 |

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| Oxygen | 46.858020 | 47.363540 | 50.000000 | -2.85 | 0.0084 |
| RunTime | 10.476886 | 10.659412 | 10.000000 | 2.01 | 0.0547 |
| RunPulse | 168.252615 | 172.894991 | 180.000000 | -3.00 | 0.0182 |

## Example 56.10: 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 56.10.1 describes the method used in the multiple imputation process.

**Output 56.10.1** 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 | 5 |
| Number of Burn-in Iterations | 200 |
| Number of Iterations | 100 |
| Seed for random number generator | 17655417 |

The “Missing Data Patterns” table in Output 56.10.2 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 56.10.2** Missing Data Patterns

Missing Data Patterns

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | O | 4 | 12.90 |
| 3 | X | O | O | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | O | 2 | 6.45 |

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Oxygen | RunTime | RunPulse |
| 1 | 46.353810 | 10.809524 | 171.666667 |
| 2 | 47.109500 | 10.137500 | . |
| 3 | 52.461667 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 9.885000 | . |

As shown in the table in Output 56.10.2, 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  nimpute=0;
   var Oxygen RunTime RunPulse;
run;
```

The “Missing Data Patterns” table in Output 56.10.3 displays a monotone missing data pattern.

**Output 56.10.3** Monotone Missing Data Patterns

```
                         The MI Procedure

                       Missing Data Patterns

                                Run     Run
          Group     Oxygen     Time    Pulse         Freq     Percent

              1     X          X       X              110       70.97
              2     X          X       .               30       19.35
              3     X          .       .               15        9.68

                       Missing Data Patterns

                   -----------------Group Means----------------
           Group          Oxygen         RunTime        RunPulse

               1       46.152428       10.861364      171.863636
               2       47.796038       10.053333               .
               3       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 56.11: Checking Convergence in MCMC

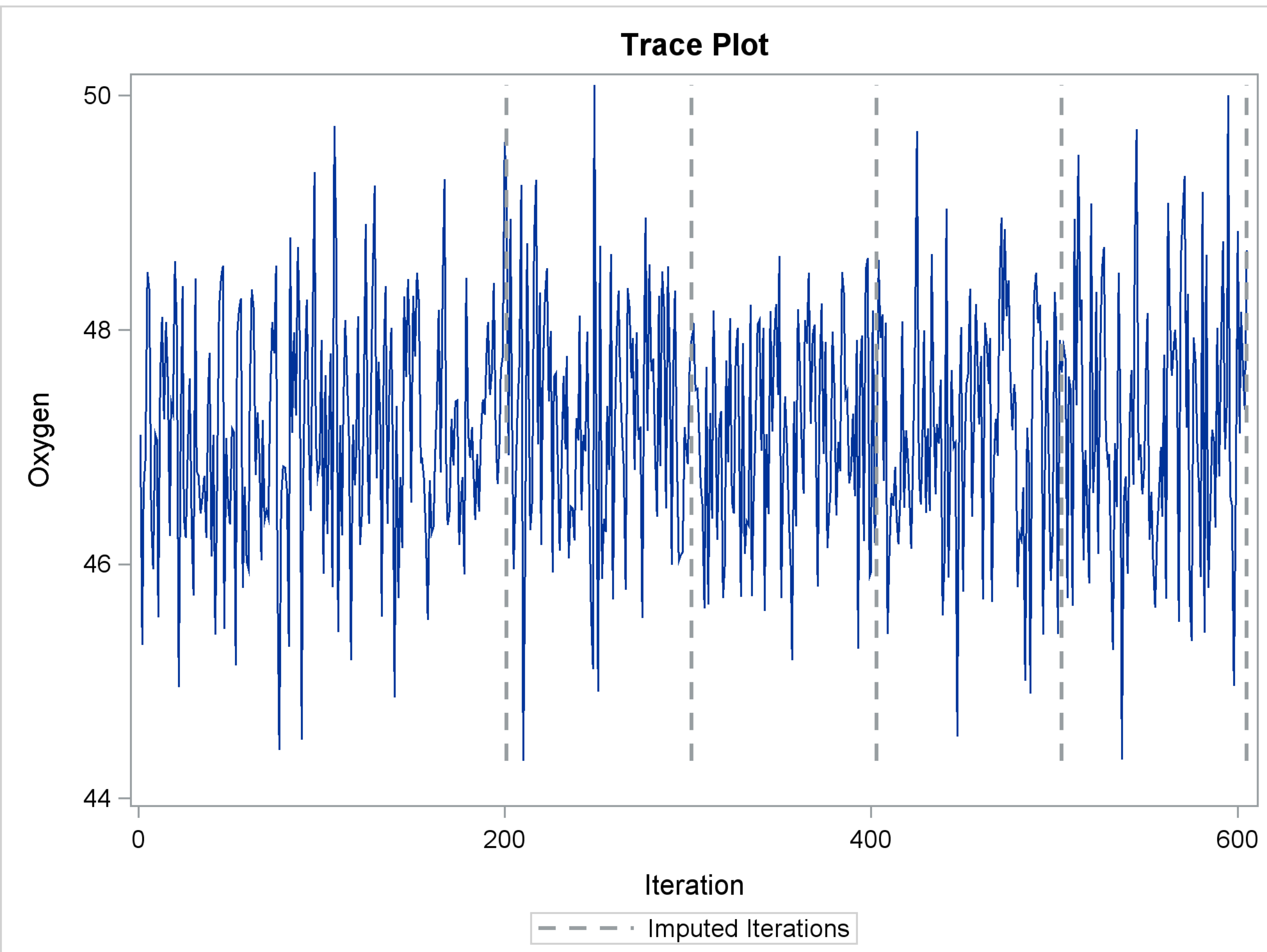
This example uses the MCMC method with a single chain. It also displays trace and autocorrelation plots to check convergence for the single chain.

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 mu0=50 10 180;
   mcmc plots=(trace(mean(Oxygen)) acf(mean(Oxygen)));
   var Oxygen RunTime RunPulse;
run;
ods graphics off;
```

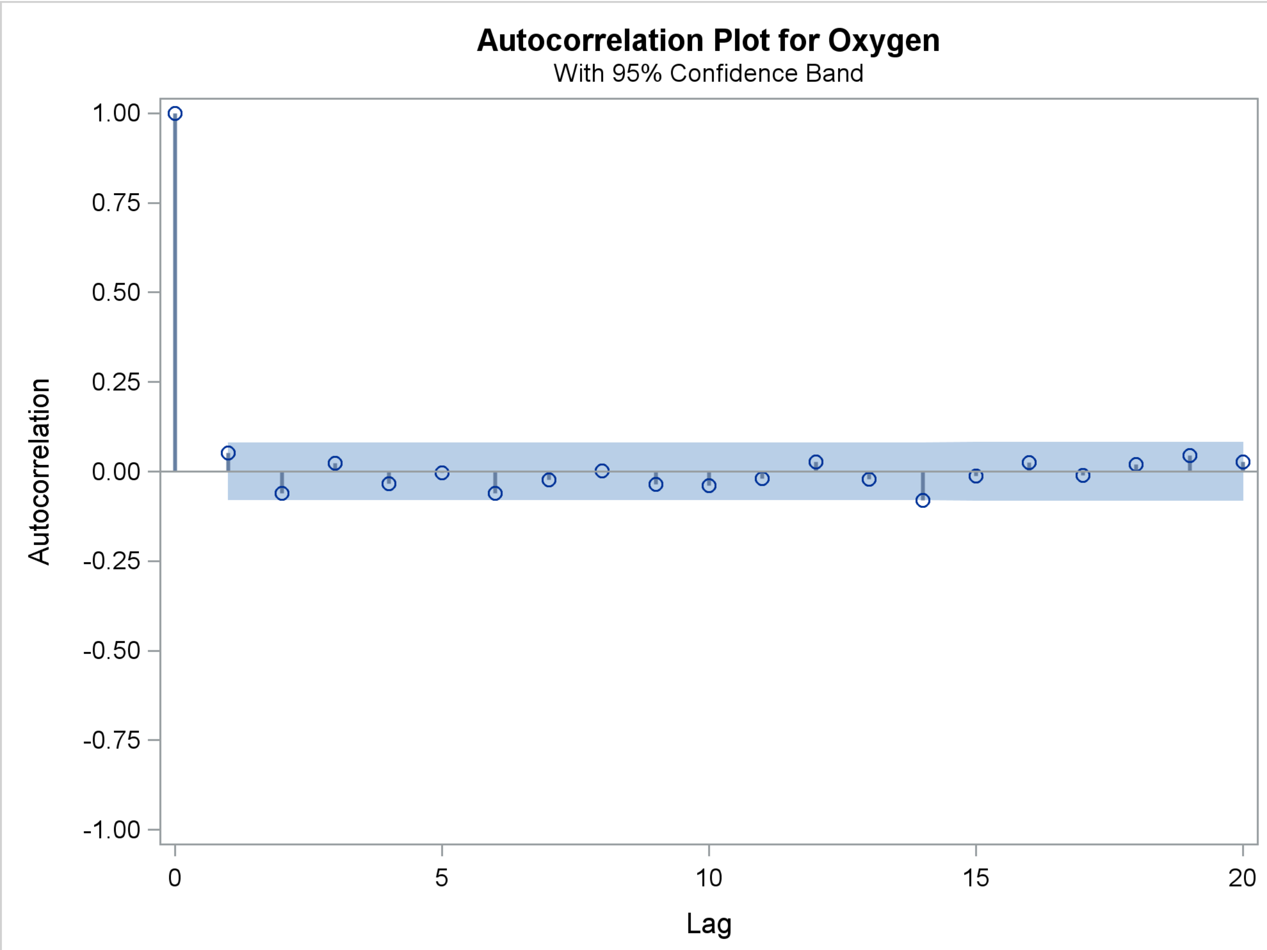
With ODS Graphics enabled, the TRACE(MEAN(OXYGEN)) option in the PLOTS= option displays the trace plot of means for the variable Oxygen, as shown in Output 56.11.1. The dashed vertical lines indicate the imputed iterations—that is, the Oxygen values used in the imputations. The plot shows no apparent trends for the variable Oxygen.

**Output 56.11.1** 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 56.11.2. The autocorrelation function plot shows no significant positive or negative autocorrelation.

**Output 56.11.2** 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 4585.

## Example 56.12: Saving and Using Parameters for MCMC

This example uses the MCMC method with multiple chains as specified in Example 56.9. 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 three imputations:

```
proc mi data=Fitness1 seed=21355417 nimpute=6 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 56.12.1. 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 56.12.1** 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 56.12.2 describes the method used in the multiple imputation process. The remaining tables for the example are identical to the tables in Output 56.9.2, Output 56.9.4, Output 56.9.5, and Output 56.9.6 in Example 56.9.

**Output 56.12.2** Model Information

```
                              The MI Procedure

                             Model Information

        Data Set                               WORK.FITNESS1
        Method                                 MCMC
        INEST Data Set                         WORK.MIEST
        Number of Imputations                  6
```

## Example 56.13: 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 mu0=50 10 180 out=outex13;
   transform log(Oxygen);
   mcmc chain=multiple displayinit;
   var Oxygen RunTime RunPulse;
run;
```

The “Missing Data Patterns” table in Output 56.13.1 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 56.13.1** Missing Data Patterns

The MI Procedure

Missing Data Patterns

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | . | 4 | 12.90 |
| 3 | X | . | . | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | . | 2 | 6.45 |

Transformed Variables: Oxygen

Missing Data Patterns

| | Group Means | | |
|---|---|---|---|
| Group | Oxygen | RunTime | RunPulse |
| 1 | 3.829760 | 10.809524 | 171.666667 |
| 2 | 3.851813 | 10.137500 | . |
| 3 | 3.955298 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 9.885000 | . |

Transformed Variables: Oxygen

The "Variable Transformations" table in Output 56.13.2 lists the variables that have been transformed.

**Output 56.13.2** Variable Transformations

Variable Transformations

| Variable | _Transform_ |
|---|---|
| Oxygen | LOG |

The "Initial Parameter Estimates for MCMC" table in Output 56.13.3 displays the starting mean and covariance estimates used in the MCMC method.

**Output 56.13.3** 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 56.13.4 displays variance information from the multiple imputation.

**Output 56.13.4** Variance Information

Variance Information

| | Variance | | | |
|---|---|---|---|---|
| Variable | Between | Within | Total | DF |
| * Oxygen | 0.000016175 | 0.000401 | 0.000420 | 26.499 |
| RunTime | 0.001762 | 0.065421 | 0.067536 | 27.118 |
| RunPulse | 0.205979 | 3.116830 | 3.364004 | 25.222 |

* Transformed Variables

Variance Information

| Variable | Relative Increase in Variance | Fraction Missing Information | Relative Efficiency |
|---|---|---|---|
| * Oxygen | 0.048454 | 0.047232 | 0.990642 |
| RunTime | 0.032318 | 0.031780 | 0.993684 |
| RunPulse | 0.079303 | 0.075967 | 0.985034 |

* Transformed Variables

Output 56.13.5 displays parameter estimates from the multiple imputation. Note that the parameter value of $\mu_0$ has also been transformed using the logarithmic transformation.

**Output 56.13.5** Parameter Estimates

Parameter Estimates

| Variable | Mean | Std Error | 95% Confidence Limits | | DF |
|---|---|---|---|---|---|
| * Oxygen | 3.845175 | 0.020494 | 3.8031 | 3.8873 | 26.499 |
| RunTime | 10.560131 | 0.259876 | 10.0270 | 11.0932 | 27.118 |
| RunPulse | 171.802181 | 1.834122 | 168.0264 | 175.5779 | 25.222 |

* Transformed Variables

Parameter Estimates

| Variable | Minimum | Maximum | Mu0 | t for H0: Mean=Mu0 | Pr > \|t\| |
|---|---|---|---|---|---|
| * Oxygen | 3.838599 | 3.848456 | 3.912023 | -3.26 | 0.0030 |
| RunTime | 10.493031 | 10.600498 | 10.000000 | 2.16 | 0.0402 |
| RunPulse | 171.251777 | 172.498626 | 180.000000 | -4.47 | 0.0001 |

* Transformed Variables

The following statements list the first 10 observations of the data set outex13 in Output 56.13.6. 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 56.13.6** Imputed Data Set in Original Scale

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Oxygen | RunTime | Run Pulse |
|---|---|---|---|---|
| 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 56.13.6 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 56.14: 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 56.10. 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 out=outex14;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

The “Missing Data Patterns” table in Output 56.14.1 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 56.14.1** Missing Data Patterns

The MI Procedure

Missing Data Patterns

| Group | Oxygen | Run Time | Run Pulse | Freq | Percent |
|---|---|---|---|---|---|
| 1 | X | X | X | 21 | 67.74 |
| 2 | X | X | O | 4 | 12.90 |
| 3 | X | O | O | 3 | 9.68 |
| 4 | . | X | X | 1 | 3.23 |
| 5 | . | X | O | 2 | 6.45 |

Missing Data Patterns

| | ------------------ | Group Means | ------------------ |
|---|---|---|---|
| Group | Oxygen | RunTime | RunPulse |
| 1 | 46.353810 | 10.809524 | 171.666667 |
| 2 | 47.109500 | 10.137500 | . |
| 3 | 52.461667 | . | . |
| 4 | . | 11.950000 | 176.000000 |
| 5 | . | 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 56.14.2 shows that a monotone method is used to generate imputed values in the first BY group.

**Output 56.14.2** Model Information

```
----------------------------- Imputation Number=1 ------------------------------

                                The MI Procedure

                               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 56.14.3 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 56.14.3** Monotone Model Specification

```
----------------------------- Imputation Number=1 ------------------------------

                          Monotone Model Specification

                                             Imputed
                         Method              Variables

                         Regression          RunTime RunPulse
```

The "Missing Data Patterns" table in Output 56.14.4 lists distinct missing data patterns with corresponding statistics. It shows a monotone missing pattern for the imputed data set.

**Output 56.14.4** Missing Data Patterns

```
----------------------------- Imputation Number=1 ------------------------------

                              Missing Data Patterns

                                   Run       Run
            Group     Oxygen       Time      Pulse          Freq      Percent

                1     X            X         X                22        70.97
                2     X            X         .                 6        19.35
                3     X            .         .                 3         9.68

                              Missing Data Patterns

                       -----------------Group Means----------------
             Group          Oxygen          RunTime          RunPulse

                 1       46.057479        10.861364        171.863636
                 2       46.745227        10.053333                 .
                 3       52.461667                .                 .
```

The following statements list the first 10 observations of the data set outex14a in Output 56.14.5:

```
proc print data=outex14a(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

**Output 56.14.5** Imputed Data Set

First 10 Observations of the Imputed Data Set

| Obs | _Imputation_ | Oxygen | RunTime | Run Pulse |
|---|---|---|---|---|
| 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.

## References

Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis,* Second Edition, New York: John Wiley & Sons.

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

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

Barnard, J., and 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.

Brand, J. P. L. (1999), "Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets," Ph.D. dissertation, Erasmus University, Rotterdam.

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*, Ser. 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.

Heitjan, F. and Little, R. J. A. (1991), "Multiple Imputation for the Fatal Accident Reporting System," *Applied Statistics*, 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," *The 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. and Rubin, D. B. (2002), *Statistical Analysis with Missing Data*, Second Edition, New York: 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.

Oudshoorn, C. G. M., van Buuren, S., and Rijckevorsel, J. L. A. (1999), "Flexible Multiple Imputation by Chained Equations of the AVO-95 survey," *TNO Prevention and Health*, *TNO Report PG/VGZ/99.045*.

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 and 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., Boshuizen, H. C., and Knook, D. L. (1999), "Multiple Imputation of Missing Blood Pressure Covariates in Survival Analysis," *Statistics in Medicine*, 18, 681–694.

van Buuren, S., Brand, J. P. L., Groothuis-Oudshoorn, C. G. M., and Rubin, D. B. (2006), "Fully Conditional Specification in Multiple Imputation," *Journal of Statistical Computation and Simulation*, 76, 1049–1064.

# Subject Index

**N**



**O**



**P**



**R**



**S**

## T



## W

# Syntax Index

**N**



**O**



**P**



**R**



**S**



**T**

**V**



**W**



**X**

# Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **`yourturn@sas.com`**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **`suggest@sas.com`**.

# SAS® Publishing Delivers!

**Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.**

## SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

**support.sas.com/saspress**

## SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

**support.sas.com/publishing**

## SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

**support.sas.com/spn**

sas THE POWER TO KNOW®