Chapter 9
The MI Procedure

Chapter Table of Contents

Chapter 9
The MI Procedure

Overview

The experimental MI procedure performs multiple imputation of missing data. 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 analyzing only complete cases has its simplicity, the information contained in the incomplete cases is lost. This approach also ignores possible systematic differences between the complete cases and the incomplete cases, and the resulting inference may 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 available 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. PROC CORR also estimates a correlation by using all cases with nonmissing values for this pair of variables. This makes better use of the available data, but the resulting correlation matrix may not be positive definite.

Another strategy for handling missing data is simple imputation, which substitutes 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 with the variable mean of the complete cases, or it can be imputed with the mean conditional on observed values of other variables. This approach treats missing values as if they were known in the complete-data analysis. However, single imputation does not reflect the uncertainty about the predictions of the unknown missing values, and the resulting estimated variances of the parameter estimates will be biased toward zero (Rubin 1987, p. 13).

Instead of filling in a single value for each missing value, multiple imputation (Rubin 1976; 1987) replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute. 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, confidence intervals with the correct probability coverage.

Multiple imputation inference involves three distinct phases:

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

The new MI procedure creates multiply imputed data sets for incomplete multivariate data. It uses methods that incorporate appropriate variability across the *m* imputations. The method of choice depends on the patterns of missingness. A data set with variables Y_1, Y_2, ..., 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, either a parametric regression method (Rubin 1987) that assumes multivariate normality or a nonparametric method that uses propensity scores (Rubin 1987; Lavori, Dawson, and Shera 1995) is appropriate. For data sets with arbitrary missing patterns, a Markov Chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality is used to impute all missing values or just enough missing values to make the imputed data sets have monotone missing patterns.

Once the *m* complete data sets are analyzed using standard SAS procedures, the new MIANALYZE procedure can be used to generate valid statistical inferences about these parameters by combining results from the *m* analyses. These two procedures are available in experimental form in Release 8.2 of the SAS System.

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 "Multiple Imputation Efficiency" section on page 174.

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 may not be the same. The consequence for different scenarios (Schafer 1997, pp. 139–143) is discussed in the "Imputer's Model Versus Analyst's Model" section on page 174.

In addition to the multiple imputation method, a simulation-based method of parameter simulation can also be used to analyze the data for many incomplete-data problems. Although the MI procedure does not offer a simulation-based method of parameter simulation, the choice between the two methods (Schafer 1997, pp. 89–90, 135–136) is examined in the "Parameter Simulation Versus Multiple Imputation" section on page 175.

Getting Started

Consider the following Fitness data set that has been altered to contain an arbitrary pattern of missingness:

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;

   data FitMiss;
      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 “Statistical Assumptions for Multiple Imputation” section on page 154 for a detailed description of the MAR assumption.

The following statements invoke the MI procedure and impute missing values for the FitMiss data set.

```
   proc mi data=FitMiss seed=37851 mu0=50 10 180 out=outmi;
      var Oxygen RunTime RunPulse;
   run;
```

```
                              The MI Procedure

                             Model Information

           Data Set                             WORK.FITMISS
           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     37851
```

Figure 9.1. Model Information

The “Model Information” table describes the method used in the multiple imputation process. By default, the procedure 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 EM algorithm and is used as the starting value for the chain.

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 has influence on 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 MI Procedure

                            Missing Data Patterns

                               Run     Run
            Group    Oxygen    Time    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               .
```

Figure 9.2. Missing Data Patterns

The “Missing Data Patterns” table lists distinct missing data patterns with corresponding frequencies and percents. Here, 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 an individual’s value is observed or missing for each variable to be analyzed. For a detailed description of missing data patterns, see the “Missing Data Patterns” section on page 155.

```
                                The MI Procedure

                   Multiple Imputation Variance Information

                    -----------------Variance----------------
 Variable          Between          Within           Total       DF

 Oxygen           0.045321        0.937239        0.991624   26.113
 RunTime          0.005853        0.072217        0.079241    24.45
 RunPulse         0.611864        3.247163        3.981400   19.227

                    Multiple Imputation Variance Information

                                Relative       Fraction
                                Increase        Missing
                    Variable     in Variance    Information

                    Oxygen          0.058027       0.056263
                    RunTime         0.097265       0.092202
                    RunPulse        0.226116       0.197941
```

Figure 9.3. Variance Information

After the completion of *m* imputations, the “Multiple Imputation Variance Information” table 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 and the fraction of missing information for each variable are also displayed. A detailed description of these statistics is provided in the “Combining Inferences from Multiply Imputed Data Sets” section on page 173.

The following “Multiple Imputation Parameter Estimates” table 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 “Combining Inferences from Multiply Imputed Data Sets” section on page 173.

```
                                The MI Procedure

                    Multiple Imputation Parameter Estimates

  Variable            Mean      Std Error    95% Confidence Limits        DF

  Oxygen         47.126919       0.995803      45.0804      49.1734   26.113
  RunTime        10.546494       0.281498       9.9661      11.1269    24.45
  RunPulse      171.621676       1.995344     167.4487     175.7946   19.227

                    Multiple Imputation Parameter Estimates

                                                            t for H0:
Variable         Minimum        Maximum            Mu0    Mean=Mu0   Pr > |t|

Oxygen         46.849494      47.318758      50.000000        -2.89     0.0077
RunTime        10.464123      10.669193      10.000000         1.94     0.0638
RunPulse      170.623678     172.680679     180.000000        -4.20     0.0005
```

Figure 9.4. Parameter Estimates

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 ten observations of data set outmi.

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

```
                 First 10 Observations of the Imputed Data Set

                                                                 Run
              Obs    _Imputation_      Oxygen    RunTime     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     6.1569    138.583
                5          1          49.8740     9.2200    164.163
                6          1          44.8110    11.6300    176.000
                7          1          46.0264    11.9500    176.000
                8          1          42.3040    10.8500    182.486
                9          1          39.4420    13.0800    174.000
               10          1          60.0550     8.6300    170.000
```

Figure 9.5. Imputed Data Set

The table 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.

Syntax

The following statements are available in PROC MI.

> **PROC MI** < *options* > **;**
>
> **BY** *variables* **;**
> **EM** < *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 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.

The MCMC statement uses a Markov chain Monte Carlo method to impute values for a data set with an arbitrary missing pattern. The MONOTONE statement uses either a parametric regression method or a nonparametric method based on propensity scores to impute values for a data set with a monotone missing pattern. Note that you can use either an MCMC statement or a MONOTONE statement, but not both. When neither of these two statements is specified, the MCMC method with its default options is used.

The TRANSFORM statement lists the variables to be transformed before the imputation process. The imputed values of these transformed variables will be 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 in alphabetical order.

PROC MI Statement

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

The following table summarizes the options available in the PROC MI statement.

Table 9.1. Summary of PROC MI Options

Tasks	**Options**
Specify data sets	
input data set	DATA=
output data set with imputed values	OUT=
Specify imputation details	
number of imputations	NIMPUTE=
seed to begin random number generator	SEED=
units to round imputed variable values	ROUND=
maximum values for imputed variable values	MAXIMUM=
minimum values for imputed variable values	MINIMUM=
singularity tolerance	SINGULAR=
Specify statistical analysis	
level for the confidence interval, $(1-\alpha)$	ALPHA=
means under the null hypothesis	MU0=
Control printed output	
suppress all displayed output	NOPRINT
displays univariate statistics and correlations	SIMPLE

The following options can be used in the PROC MI statement (in alphabetical order):

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

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

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. A missing value indicates no restriction on the maximum for the corresponding variable. The default is MAXIMUM=., no restriction on the maximum.

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 not applicable if you specify the METHOD=PROPENSITY option in the MONOTONE statement.

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. A missing value indicates no restriction on the minimum for the corresponding variable. The default is MINIMUM=. , no restriction on the minimum.

MU0=*numbers*
THETA0=*numbers*

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

If a variable is transformed as specified in a TRANSFORM statement, then the same transformation for that variable is also applied to its corresponding specified MU0= value in the t test. If the parameter values $\boldsymbol{\mu}_0$ for a transformed variable is not specified, then $\boldsymbol{\mu}_0 = 0$ is used for that transformed variable.

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). For more information, refer to the chapter "Using the Output Delivery System" in the *SAS/STAT User's Guide, Version 8*.

OUT=*SAS-data-set*

creates an output SAS data set containing 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 replaced by the imputed values. See the "Output Data Sets" section on page 171 for a description of this data set.

If you want to create a permanent SAS data set, you must specify a two-level name. For more information on permanent SAS data sets, refer to the section “SAS Files” in *SAS Language Reference: Concepts, Version 8*.

ROUND=*numbers*

specifies the units to round variables in the 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 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.

You can use the ROUND= option to set 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 9.3 for a usage of this option.

SEED=*number*

specifies a positive integer. PROC MI uses the value of the SEED= option 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 control the value of the seed explicitly rather than rely on the clock reading.

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 “Descriptive Statistics” section on page 152.

SINGULAR=*p*

specifies the criterion for determining the singularity of a covariance matrix, where $0 < p < 1$. The default is SINGULAR=1E−8.

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

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MI to obtain separate analyses on observations in groups 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 your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING 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 using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts, Version 8*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide, Version 8*.

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. 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.

You can also use the EM statement with the NIMPUTE=0 option in the PROC statement to compute the EM estimates without multiple imputation, as shown in Example 9.1 in the “Examples” section on page 177.

The following five options are available with the EM statement.

CONVERGE=*p*
sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. 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.

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

OUTEM=*SAS-data-set*
creates an output SAS data set of TYPE=COV containing the MLE of the parameter vector $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. These estimates are computed with the EM algorithm. See the “Output Data Sets” section on page 171 for a description of this output data set.

OUTITER $<$ **(options)** $>$ **=***SAS-data-set*
creates an output SAS data set of TYPE=COV containing 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 “Output Data Sets” section on page 171 for a description of this data set.

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence for 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. The following table summarizes the options available for the MCMC statement.

Table 9.2. Summary of Options in MCMC

Tasks	**Options**
Specify data sets	
input parameter estimates for imputations	INEST=
output parameter estimates used in imputations	OUTEST=
output parameter estimates used in iterations	OUTITER=
Specify imputation details	
monotone/full imputation	IMPUTE=
single/multiple chain	CHAIN=
number of burn-in iterations for each chain	NBITER=
number of iterations between imputations in a chain	NITER=
initial parameter estimates for MCMC	INITIAL=
prior parameter information	PRIOR=
starting parameters	START=
Specify output graphics	
displays time-series plots	TIMEPLOT=
displays autocorrelation plots	ACFPLOT=
graphics catalog name for saving graphics output	GOUT=
Control printed output	
displays worst linear function	WLF
displays initial parameter values for MCMC	DISPLAYINIT

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

ACFPLOT < *(options* < */ display-options* > *)* >
displays the autocorrelation function plots of parameters from iterations.

The available options are:

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:

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 for 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 LREF=1, a solid line.

LOG
requests that the logarithmic transformations of parameters be used to compute the autocorrelations. It's 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.

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 for 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 for 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 for 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 statement

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

requests autocorrelation function plots for the means and variances of the variable `y1`, respectively. Logarithmic transformations of both the means and variances are used in the plots. For a detailed description of the autocorrelation function plot, see the "Autocorrelation Function Plot" section on page 169; refer also to Schafer (1997, pp. 120-126) and the *SAS/ETS User's Guide, Version 8*.

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 process 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, refer to the chapter "The GREPLAY Procedure" in *SAS/GRAPH Software: Reference, Version 8*.

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 containing 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 “Input Data Sets” section on page 170.

INITIAL=EM < (*options*) >
INITIAL=INPUT=*SAS-data-set*

specifies the initial mean and covariance estimates for the MCMC process. 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 “Input Data Sets” section on page 170 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 EM is used to start the EM algorithm for the posterior mode, and the resulting EM estimates are used to begin the MCMC process.

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 process. If you specify the BOOTSTRAP option without the number, *p*=0.75 is used by default.

CONVERGE=*p*

sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. 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 “Output Data Sets” section on page 171 for a description of this data set.

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

creates an output SAS data set of TYPE=COV containing 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 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. When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. See the “Output Data Sets” section on page 171 for a description of this data set.

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 containing prior information.

For a detailed description of the prior information, see the “Bayesian Estimation of the Mean Vector and Covariance Matrix” section on page 161 and the “Posterior Step” section on page 162. 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 “Input Data Sets” section on page 170 for a description of this data set.

START=VALUE | DIST

specifies that the initial parameter estimates are used as either the starting value (START=VALUE) or as the starting distribution (START=DIST) in the first imputation step of each chain. The default is START=VALUE.

TIMEPLOT $<$ *(options* $<$ */ display-options* $>$ *)* $>$

displays the time-series plots of parameters from iterations.

The available options are:

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

The display-options provide additional information for the time-series plots. The available display-options are:

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 time-series plots. The default is CSYMBOL=BLACK.

HSYMBOL=*number*

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

LOG

requests that the logarithmic transformations of parameters be used. It's 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.

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 time-series plots. The default is TITLE='Time-series Plot for Iterations'.

For a detailed description of the time-series plot, see the "Time-Series Plot" section on page 168 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 "Worst Linear Function of Parameters" section on page 168.

MONOTONE Statement

MONOTONE < *options* > **;**

The MONOTONE statement specifies an imputation method 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. When both MONOTONE and MCMC statements are specified, the MONOTONE statement is not used.. You can specify the following options in a MONOTONE statement.

METHOD=REG | REGRESSION
METHOD=PROPENSITY < / **NGROUPS =** *number*>

specifies the imputation method for a data set with a monotone missing pattern. You can specify either METHOD=REG, a parametric regression method, or METHOD=PROPENSITY, a nonparametric method based on propensity scores. The default is METHOD=REG.

When METHOD=PROPENSITY is specified, the MAXIMUM=, MINIMUM=, and ROUND= options, which make the imputed values more consistent with the observed variable values, are not applicable.

NGROUPS=*number*

specifies the number of groups based on propensity scores for METHOD=PROPENSITY. The default is NGROUPS=5.

See the "Regression Method for Monotone Missing Data" section on page 157 for a detailed description of the regression method, and the "Propensity Score Method for Monotone Missing Data" section on page 158 for the propensity score method.

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 reverse-transformed to create the imputed data set.

The following transformations can be used as the *transform* in the TRANSFORM statement.

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

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

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

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

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

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

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

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

For example, the statement

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

requests that variables $\log(\mathrm{y1})$, a logarithmic transformation for the variable y1, and $\sqrt{\mathrm{y2} + 1}$, a power transformation for the variable y2, be used in the imputation.

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 9.7 for a usage of the TRANSFORM statement.

VAR Statement

VAR *variables* **;**

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

Details

Descriptive Statistics

Suppose $\mathbf{Y}$ is the $n \times p$ matrix of complete data, which may 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 $\mathbf{R}$ containing the Pearson product-moment correlations of the variables is derived by scaling the corresponding covariance matrix:

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

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

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

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

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 (1987), Schafer (1997), and McLachlan and Krishnan (1997) provide 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 non-decreasing 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

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

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

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

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.

Refer to 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. 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 option ITPRINT. PROC MI lists the iteration number, the likelihood -2 Log L, and parameter values $\boldsymbol{\mu}$ at each iteration. You can also save the MLE derived from the EM algorithm in a SAS data set specified with 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 on 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 SAS 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}.

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

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.

Furthermore, the MI and MIANALYZE procedures assume that the parameters $\boldsymbol{\theta}$ of the data model and the parameters ϕ 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 ϕ, 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 an individual's value is observed or missing for each variable to be analyzed. 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. The following figure displays the eight groups of observations and an unique missing pattern for each group:

```
                    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      .     .     .
```

Figure 9.6. Missing Data Patterns

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, ..., 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, the following figure displays a data set of three variables with a monotone missing pattern. Note that this data set does not have any observations with missing patterns such as in Groups 3, 5, 6, 7, or 8 in the previous example.

```
                Monotone Missing Data Patterns

                   Group    Y1    Y2    Y3

                     1      X     X     X
                     2      X     X     .
                     3      X     .     .
```

Figure 9.7. Monotone Missing Patterns

Imputation Mechanisms

This section describes the three methods for multiple imputation that are available in the MI procedure. The method of choice depends on the patterns of missingness in the data.

- For data sets with monotone missing patterns, either a parametric regression method (Rubin 1987) that assumes multivariate normality or a nonparametric method that uses propensity scores (Rubin 1987; Lavori, Dawson, and Shera 1995) is appropriate.
- For data sets with arbitrary missing patterns, a Markov Chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality is used to impute either all 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 strategies. For example, in addition to the MCMC method, you can also implement other methods, such as a regression method, that do not use Markov chains.

With an arbitrary missing data pattern, you can often use the MCMC method, which creates multiple imputations by drawing simulations from a Bayesian predictive distribution for normal data. Another way to handle a data set with an arbitrary missing data pattern is to use the MCMC approach to impute enough values to make the missing data pattern monotone. Then, you can use a more flexible imputation method. This approach is described in the "Producing Monotone Missingness with the MCMC Method" section on page 164.

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. It often makes sense to use a normal model to create multiple imputations even when the observed data are somewhat non-normal, as supported by simulation studies described in Schafer (1997) and the original references therein.

You can also use a TRANSFORM statement to transform variables to conform to the multivariate normality assumption. With a TRANSFORM statement, variables are transformed before the imputation process and then are reverse-transformed to create the imputed data set.

Li (1988) presented an argument for convergence of the MCMC method in the continuous case in theory and used it to create imputations for incomplete multivariate continuous data. But in practice, it is not easy to check the convergence of a Markov chain, especially for parameters from a large number of variables. PROC MI generates statistics and plots that you can use to check for convergence of the MCMC process. The details are described in the "Convergence in MCMC" section on page 167.

Regression Method for Monotone Missing Data

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

In the regression method, a regression model is fitted for each variable with missing values, with the previous variables as covariates. Based on the fitted regression coefficients, 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). The process is repeated sequentially for variables with missing values. That is, for a variable Y_j with missing values, a model

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

is fitted using observations with observed values for variables Y_1, Y_2, ..., Y_j.

The fitted model includes the regression parameter estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_{j-1})$ 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 derived from the intercept and variables $Y_1, Y_2, ..., Y_{j-1}$.

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

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

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

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

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.

The missing values are then replaced by

$$\beta_{*0} + \beta_{*1} \, y_1 + \beta_{*2} \, y_2 + \ldots + \beta_{*(j-1)} \, y_{j-1} + z_i \, \sigma_{*j}$$

where $y_1, y_2, ..., y_{j-1}$ are the covariate values of the first $j - 1$ variables and z_i is a simulated normal deviate.

Propensity Score Method for Monotone Missing Data

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 each 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).

A data set with variables Y_1, Y_2, ..., Y_p (in that order) is said to have a monotone missing pattern when the event that a variable Y_j is observed for a particular individual implies that all previous variables Y_k, $k < j$, are also observed for that individual. The propensity score method uses the following steps to impute values for each variable Y_j with missing values:

1. Create an indicator variable R_j with the value 0 for observations with missing Y_j and 1 otherwise.

2. Fit a logistic regression model

$$\text{logit}(p_j) = \beta_0 + \beta_1 Y_1 + \beta_2 Y_2 + \ldots + \beta_{j-1} Y_{j-1}$$

where $p_j = Pr(R_j = 0|Y_1, Y_2, ..., Y_{j-1})$ and $\text{logit}(p) = \log(p/(1-p))$.

3. Create a propensity score for each observation to estimate the probability that it is missing.

4. Divide the observations into a fixed number of groups (typically assumed to be five) based on these propensity scores.

5. Apply 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 analogue 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.

Note that 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 an univariate analysis, but it is not appropriate for analyses involving relationship among variables, such as a regression analysis. It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

MCMC Method for Arbitrary Missing 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 pseudo-random 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 one.

In MCMC simulation, one constructs a Markov chain long enough for the distribution of the elements to stabilize to a stationary distribution, which is the distribution of interest. By repeatedly simulating steps of the chain, the method simulates draws from the distribution of interest. Refer to 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, one 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/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 be helpful to stabilize the inference about the mean vector for a near singular covariance matrix.

The two steps are iterated long enough for the results to be reliable for a multiply imputed data set (Schafer 1997, p. 72). That is, with a current parameter estimate $\boldsymbol{\theta}^{(t)}$ at the $t\mathrm{th}$ 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 $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis}^{(t+1)})$.

This creates a Markov chain

$(Y_{mis}^{(1)}, \boldsymbol{\theta}^{(1)})$, $(Y_{mis}^{(2)}, \boldsymbol{\theta}^{(2)})$, ... ,

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

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

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

Imputation Step

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

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

Also suppose

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

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

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

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

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

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

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

and the conditional covariance matrix

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

Bayesian Estimation of the Mean Vector and Covariance Matrix

Suppose that $\mathbf{Y} = (\,\mathbf{y}'_1, \mathbf{y}'_2, ..., \mathbf{y}'_n\,)'$ 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 $\boldsymbol{\Lambda}$. Then the SSCP matrix

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

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

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

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

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

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

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

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

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

where $\tau > 0$ is a fixed number. The posterior distribution (Anderson 1984, p. 270; Schafer 1997, p. 152) is

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

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 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 μ 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, a noninformative prior can be used by specifying the PRIOR=JEFFREYS option. The posterior distributions (Schafer 1997, p. 154) are

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

2. An Informative Prior for μ and $\boldsymbol{\Sigma}$

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

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

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 μ, 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 “Bayesian Estimation of the Mean Vector and Covariance Matrix” section on page 161, is given by

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

where

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

3. An Informative Prior for $\boldsymbol{\Sigma}$

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 μ to stabilize the inference about μ. 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’.

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

Corresponding to the prior for $\boldsymbol{\Sigma}$

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

the posterior distribution for $\boldsymbol{\Sigma}$ (Anderson 1984, p. 269) is

$$\boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} \quad \sim \quad W^{-1}\left((n-1)+d^{*},\ (n-1)\mathbf{S}+d^{*}\mathbf{S}^{*}\right)$$

Thus, an estimate of $\boldsymbol{\Sigma}$ is given by the weighted average

$$\frac{1}{(n-1)+d^{*}}\ \left((n-1)\mathbf{S}+d^{*}\mathbf{S}^{*}\right)$$

and the posterior distribution for $(\boldsymbol{\mu},\boldsymbol{\Sigma})$ becomes

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

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

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

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 only needs to impute 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”. A “.” means that the variable is missing and will be imputed and an “O” means that the variable is missing and will not be imputed. The tables of “Multiple Imputation Variance Information” and “Multiple Imputation Parameter Estimates” 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 similar to 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 “MCMC Method for Arbitrary Missing Data” section on page 159 except that here 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 the missing variables that will result 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 “MCMC Method for Arbitrary Missing Data” section on page 159. Instead of simulating the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters from the full imputed data set, the P-step here 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 “Regression Method for Monotone Missing Data” section on page 157.

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 nonmissing observations.

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 $Y_1, Y_2, ..., Y_{j-1}$.

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

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

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

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

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

These simulated values of $\boldsymbol{\beta}_*$ and σ_{*j}^2 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, refer to Schafer (1997, pp. 226–235).

MCMC Method Specifications

With MCMC, 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. Refer to 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 process 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 as the starting distribution (START=DIST) for the MCMC process. For multiple chains, these estimates are used again as either the starting value (START=VALUE) or as 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 process.

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 process.

With INITIAL=EM, 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 in INITIAL=EM 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 in the EM algorithm. By default, MAXITER=200.

With the BOOTSTRAP option, you can use overdispersed starting values for the MCMC process. 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).

Convergence in MCMC

The theoretical convergence of the MCMC process has been explored under various conditions, as described in Schafer (1997, p. 70). However, in practice, verification of convergence is not a simple matter and cannot be easily implemented in the MI procedure.

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, the worst linear function, and observed-data LR statistics. You can then monitor the convergence in a single chain by displaying time-series plots and autocorrelations for those parameter values (Schafer 1997, p. 120). The time-series 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 option.

You can apply EM 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 provide useful statistics and plots that can be used to check the convergence of the MCMC process.

LR Statistics

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

In each iteration, the LR statistic is given by

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

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

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

For large samples, these LR statistics tends to be approximately χ^2 distributed with degrees of freedom equal to the dimension of $\boldsymbol{\theta}$ (Schafer 1997, p. 131). For example, with a large number of iterations, if the values of the LR statistic do not behave like a random sample from the described χ^2 distribution, then there is evidence that the MCMC process 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 process. 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 $\mathbf{v} = \hat{\boldsymbol{\theta}}_{(-1)} - \hat{\boldsymbol{\theta}}$ is the difference between the estimated value of $\boldsymbol{\theta}$ one step prior to convergence and the converged value $\hat{\boldsymbol{\theta}}$.

You can display the coefficients of the worst linear function, $\mathbf{v}$, by specifying the WLF option in the MCMC statement. You can save the function value from each iteration in an OUTITER= data set by specifying the WLF option in the OUTITER option. You can also display the worst linear function values from iterations in an autocorrelation plot or a time-series 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 may take much longer than the WLF to converge, as described in Schafer (1997, p.130).

Time-Series Plot

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

You can display time-series plots for the worst linear function, the 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 the plus sign (+) as the plot symbol to display the points with a height of one (percentage screen unit) in a time-series plot. You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol.

By default, the plot title "Time-Series Plot" is displayed in a time-series plot. You can request another title by using the TITLE= option in TIMEPLOT. 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. Refer to the chapter "The SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference, Version 8* for a description of title options. See Example 9.6 for a usage of the time-series plot.

Autocorrelation Function Plot

To examine relationships of successive parameter estimates ξ, the autocorrelation function (ACF) can be used. For a stationary series, $\xi_i, i \geq 1$, in time series 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, the 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 sign (*) 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, and the CNEEDLES= and WNEEDLES= options to change the color and width of the needles. You can also use the LREF=, CREF=, and WREF= options to change the line type, color, and width of the reference line. Similarly, you can use the LCONF=, CCONF=, and WCONF= options to change the line type, color, and width of the confidence limits.

By default, the plot title “Autocorrelation Plot” is displayed in a autocorrelation function plot. You can request another title by using the TITLE= option in ACFPLOT. 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. Refer to the chapter “The SAS/GRAPH Statements” in *SAS/GRAPH Software: Reference, Version 8* for a description of title options. See Example 9.6 for a usage of the autocorrelation function plot.

Input Data Sets

You can specify the input data set with missing values with the DATA= option in the PROC MI statement. When a MCMC method is used, you can specify the data set containing the reference distribution information for imputation with the INEST= option, the data set containing initial parameter estimates for the MCMC process with the INITIAL=INPUT= option, and the data set containing 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 containing 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. See Example 9.8 for a usage of this option.

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 process. 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 n^* from observations with _TYPE_='N'.

You can also use this data set to specify a prior distribution for μ 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 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 containing MLE computed with the EM algorithm with the OUTEM= option in the EM statement. When a MCMC method is used, you can specify the data set containing parameter estimates used in each imputation with the OUTEST= option and the data set containing parameters used in the imputation step for each iteration with the OUTITER option in the MCMC statement.

OUT=*SAS-data-set*

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 replaced by imputed values.

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 9.8 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 MEAN 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 a MCMC statement

The OUTITER= data set in a 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. The following table summarizes the options available for OUTITER and the corresponding values for the output variable _TYPE_.

Table 9.3. Summary of Options for OUTITER in a MCMC statement

Options	**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 "Convergence in MCMC" section on page 167.

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 $\hat{U}_i$ are the point and variance estimates from the ith imputed data set, i=1, 2, ..., 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 $\overline{U}$ is the within-imputation variance, which is the average of the m complete-data estimates:

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

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)

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

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

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

When the complete-data degrees of freedom v_0 is 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. Barnard and Rubin (1999) recommend the use of an 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.

The degrees of freedom v_m depends on m and the ratio

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

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 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:

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

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

Multiple Imputation Efficiency

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

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

The following table shows relative efficiencies with different values of m and λ. For cases with little missing information, only a small number of imputations are necessary.

Table 9.4. Relative Efficiency

	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

Imputer's Model Versus Analyst's Model

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

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 may not be the same.

For example, consider the same 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 may 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's 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 when you include 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).

Note that it is good practice to include a description of the imputer's model with the multiply imputed data set. 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

For many incomplete-data problems, simulation-based methods of parameter simulation and multiple imputation can be used to analyze the data. In parameter simulation, you simulate random values of parameters from the observed-data posterior distribution and make 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 can be suitable (Schafer 1997).

In multiple imputation, the unknown missing data are replaced by multiple sets of simulated values. Each complete data set is then analyzed by standard complete-data methods. The variability among the results from these repeated analyses provides a measure of the uncertainty due to missing data. Combining this between-imputation variation with the ordinary within-imputation sample variation provides statistical inference for the parameters of interest. Multiple imputation is suitable for analyses that are more exploratory in nature.

Multiple imputation only requires 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).

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 the following table. For more information on ODS, refer to the chapter "Using the Output Delivery System" in the *SAS/STAT User's Guide, Version 8.*

Table 9.5. ODS Tables Produced in PROC MI

ODS Table Name	Description	Option
ModelInfo	Model information	
MissPattern	Missing data patterns	
Transform	Variable Transformations	TRANSFORM statement
Univariate	Univariate statistics for available cases	SIMPLE
Corr	Pairwise correlations for available cases	SIMPLE
EMInitEst	Initial parameter values for EM	EM statement
EMEst	MLE of the parameter vector from EM	EM statement
EMIter	EM iteration history for MLE	ITPRINT in EM statement
EMPIter	EM iteration history for posterior mode	ITPRINT in INITIAL=EM
EMPEst	Posterior mode parameter values from EM	INITIAL=EM
EMWlf	Coefficients of the worst linear function	WLF
MCMCInitEst	Initial parameter estimates for MCMC	DISPLAYINIT in MCMC
VarianceInfo	Between-imputation, within-imputation, and total variances	
ParmEst	Parameter estimates	

Examples

The following FitMono data set has a monotone missing data pattern and is used in Example 9.2 with the propensity score method and in Example 9.3 with the regression method. The FitMiss data set created in the “Getting Started” section is used in other examples. Note that the original data set has been altered for these examples.

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;
   data FitMono;
      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
   45.681  11.95  176     49.091  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     54.625   8.92  146
   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
   ;
```

Example 9.1. EM Algorithm for MLE

This example uses the EM algorithm to compute the maximum likelihood estimates for the parameters of a multivariate normal distribution using 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 FitMiss.

```
   proc mi data=FitMiss seed=55417 simple nimpute=0;
      em itprint outem=outem;
      var Oxygen RunTime RunPulse;
   run;
```

Note when you specify the option NIMPUTE=0, the missing values will not be imputed. The procedure generates the following output:

Output 9.1.1. Model Information

```
                                The MI Procedure

                               Model Information

           Data Set                              WORK.FITMISS
           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      55417
```

The “Model Information” table describes the method and options used in the procedure.

Output 9.1.2. Missing Data Patterns

```
                                The MI Procedure

                              Missing Data Patterns

                                  Run     Run
            Group    Oxygen     Time    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                .
```

The “Missing Data Patterns” table lists distinct missing data patterns with corresponding frequencies and percents. Here, “X” means that the variable is observed in the corresponding group and “.” means that the variable is missing. The table also displays group-specific variable means.

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

Output 9.1.3. Univariate Statistics

```
                                 The MI Procedure

                              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
```

Output 9.1.4. Pairwise Correlations

```
                                 The MI Procedure

                              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
```

With the EM statement, the procedure displays the initial parameter estimates for EM.

Output 9.1.5. Initial Parameter Estimates for EM

```
                                 The MI Procedure

                        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
```

With the ITPRINT option, the "EM (MLE) Iteration History" table displays the iteration history for the EM algorithm.

Output 9.1.6. EM (MLE) Iteration History

```
                                The MI Procedure

                           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
```

The procedure then displays the EM (MLE) parameter estimates, the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of a multivariate normal distribution from the data set FitMiss.

Output 9.1.7. EM (MLE) Parameter Estimates

```
                                The MI Procedure

                          EM (MLE) Parameter Estimates

       _TYPE_    _NAME_              Oxygen         RunTime        RunPulse

       MEAN                       47.104086       10.554864      171.381796
       COV       Oxygen           27.798014       -6.457929      -18.030790
       COV       RunTime          -6.457929        2.015491        3.516092
       COV       RunPulse        -18.030790        3.516092       97.766559
```

You can also output the EM (MLE) parameter estimates into 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;
```

Output 9.1.8. EM Estimates

```
                                   EM Estimates

         Obs    _TYPE_     _NAME_       Oxygen      RunTime    RunPulse

          1      MEAN                  47.1041     10.5549     171.382
          2      COV      Oxygen       27.7980     -6.4579     -18.031
          3      COV      RunTime      -6.4579      2.0155       3.516
          4      COV      RunPulse    -18.0308      3.5161      97.767
```

The output data set outem is a TYPE=COV data set. The observation with _TYPE_='MEAN' contains the MLE for the parameter μ and the observations with _TYPE_='COV' contain the MLE for the parameter Σ of a multivariate normal distribution from the data set FitMiss.

Example 9.2. Propensity Score Method

This example uses the propensity score method to impute missing values 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 outpscore.

```
   proc mi data=FitMono seed=55417 simple out=outpscore;
      monotone method=propensity;
      var Oxygen RunTime RunPulse;
   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 procedure generates the following output:

Output 9.2.1. Model Information

```
                                 The MI Procedure

                                Model Information

           Data Set                             WORK.FITMONO
           Method                               Propensity
           Number of Imputations                5
           Number of Groups on Propensity       5
           Seed for random number generator     55417
```

The “Model Information” table describes the method and options used in the multiple imputation process. By default, the observations are sorted into five groups based on the propensity scores, and five imputations are created for the missing data.

Output 9.2.2. Missing Data Patterns

```
                              The MI Procedure

                            Missing Data Patterns

                             Run     Run
       Group     Oxygen     Time    Pulse        Freq     Percent

           1     X          X       X              23       74.19
           2     X          X       .               5       16.13
           3     X          .       .               3        9.68

                           Missing Data Patterns

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

            1        46.684174       10.776957      170.739130
            2        47.505800       10.280000               .
            3        52.461667               .               .
```

The “Missing Data Patterns” table lists distinct missing data patterns with corresponding frequencies and percents. Here, “X” means that the variable is observed in the corresponding group and “.” means that the variable is missing. The table also displays group-specific variable means.

Output 9.2.3. Variance Information

```
                              The MI Procedure

                 Multiple Imputation Variance Information

                  -----------------Variance-----------------
    Variable         Between         Within          Total       DF

    RunTime         0.001068       0.059100       0.060382   27.498
    RunPulse        1.147555       4.686646       6.063711   17.006

                  Multiple Imputation Variance Information

                               Relative       Fraction
                               Increase        Missing
                  Variable     in Variance    Information

                  RunTime         0.021688       0.021448
                  RunPulse        0.293828       0.246288
```

After the completion of *m* imputations, the “Multiple Imputation Variance Information” table 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 and the fraction of missing information for each variable are also displayed. A detailed description of these statistics is provided in the “Combining Inferences from Multiply Imputed Data Sets” section on page 173.

The “Multiple Imputation Parameter Estimates” table 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 9.2.4. Parameter Estimates

```
                               The MI Procedure

                  Multiple Imputation Parameter Estimates

  Variable            Mean      Std Error    95% Confidence Limits        DF

  RunTime        10.603677       0.245727      10.0999      11.1074   27.498
  RunPulse      170.400000       2.462460     165.2048     175.5952   17.006

                  Multiple Imputation Parameter Estimates

                                                          t for H0:
Variable         Minimum        Maximum            Mu0     Mean=Mu0   Pr > |t|

RunTime        10.558065      10.648387              0        43.15     <.0001
RunPulse      168.967742     171.838710              0        69.20     <.0001
```

The following statements list the first ten observations of the data set outpscore.

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

Output 9.2.5. Imputed Data Set

```
                 First 10 Observations of the Imputed Data Set

                                                    Run      Run
                Obs    _Imputation_    Oxygen     Time    Pulse

                  1          1         44.609    11.37     178
                  2          1         45.313    10.07     185
                  3          1         54.297     8.65     156
                  4          1         59.571     8.63     146
                  5          1         49.874     9.22     156
                  6          1         44.811    11.63     176
                  7          1         45.681    11.95     176
                  8          1         49.091    10.85     156
                  9          1         39.442    13.08     174
                 10          1         60.055     8.63     170
```

Example 9.3. Regression Method

This example uses the regression method to impute missing values in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the regression method. The resulting data set is named outreg.

```
proc mi data=FitMono round=.001 .01 1  mu0= 50 10 150
        seed=55417 out=outreg;
   monotone method=reg;
   var Oxygen RunTime RunPulse;
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 Oxygen, RunTime, and RunPulse in the order listed with 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 Oxygen=50, RunTime=10, and RunPulse=150.

The “Missing Data Patterns” table lists distinct missing data patterns with corresponding frequencies and percents. It is identical to the table in the previous example.

After the completion of five imputations by default, the “Multiple Imputation Variance Information” table displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. The relative increase in variance due to missingness and the fraction of missing information for each variable are also displayed. These statistics are described in the “Combining Inferences from Multiply Imputed Data Sets” section on page 173.

Output 9.3.1. Variance Information

```
                             The MI Procedure

                Multiple Imputation Variance Information

                 -----------------Variance-----------------
   Variable         Between         Within          Total       DF

   RunTime         0.004443       0.068684       0.074016   25.294
   RunPulse        1.790531       4.045134       6.193770   11.846

                 Multiple Imputation Variance Information

                                Relative       Fraction
                                Increase        Missing
                 Variable     in Variance    Information

                 RunTime         0.077629       0.074435
                 RunPulse        0.531166       0.382947
```

The “Multiple Imputation Parameter Estimates” table 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 9.3.2. Parameter Estimates

```
                                The MI Procedure

                    Multiple Imputation Parameter Estimates

  Variable            Mean      Std Error    95% Confidence Limits        DF

  RunTime        10.575871       0.272059      10.0159      11.1359   25.294
  RunPulse      170.425806       2.488729     164.9955     175.8561   11.846

                    Multiple Imputation Parameter Estimates

                                                           t for H0:
Variable         Minimum        Maximum            Mu0    Mean=Mu0   Pr > |t|

RunTime        10.506452      10.680968      10.000000        2.12     0.0443
RunPulse      169.290323     171.935484     150.000000        8.21     <.0001
```

The following statements list the first ten observations of the data set outreg. Note that the imputed values rounded to the same precision as the observed values.

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

Output 9.3.3. Imputed Data Set

```
                 First 10 Observations of the Imputed Data Set

                                                Run      Run
                Obs    _Imputation_    Oxygen     Time    Pulse

                  1          1         44.609    11.37     178
                  2          1         45.313    10.07     185
                  3          1         54.297     8.65     156
                  4          1         59.571     7.18     156
                  5          1         49.874     9.22     192
                  6          1         44.811    11.63     176
                  7          1         45.681    11.95     176
                  8          1         49.091    10.85     174
                  9          1         39.442    13.08     174
                 10          1         60.055     8.63     170
```

Example 9.4. 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 three imputations.

```
proc mi data=FitMiss seed=55417 nimpute=3 mu0=50 10 180;
   mcmc chain=multiple displayinit initial=em(itprint);
   var Oxygen RunTime RunPulse;
run;
```

Output 9.4.1. Model Information

```
                              The MI Procedure

                             Model Information

           Data Set                             WORK.FITMISS
           Method                               MCMC
           Multiple Imputation Chain            Multiple Chains
           Initial Estimates for MCMC           EM Posterior Mode
           Start                                Starting Value
           Prior                                Jeffreys
           Number of Imputations                3
           Number of Burn-in Iterations         200
           Seed for random number generator     55417
```

With CHAIN=MULTIPLE, 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.

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 process.

The following “Missing Data Patterns” table lists distinct missing data patterns with corresponding statistics.

Output 9.4.2. Missing Data Patterns

```
                              The MI Procedure

                            Missing Data Patterns

                             Run     Run
            Group    Oxygen    Time    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 ITPRINT option in INITIAL=EM, the procedure also displays the “EM (Posterior Mode) Iteration History” table.

Output 9.4.3. EM (Posterior Mode) Iteration History

```
                                The MI Procedure

                     EM (Posterior Mode) Iteration History

_Iteration_        -2 Log L   -2 Log Posterior         Oxygen         RunTime

          0      254.482800         282.909590      47.104086       10.554864
          1      255.081159         282.051588      47.104079       10.554859
          2      255.271405         282.017488      47.104077       10.554858
          3      255.318621         282.015372      47.104002       10.554524
          4      255.330259         282.015232      47.103861       10.554388
          5      255.333160         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      171.381796
                                    1      171.381708
                                    2      171.381669
                                    3      171.381853
                                    4      171.382058
                                    5      171.382152
                                    6      171.382186
                                    7      171.382197
```

With the DISPLAYINIT option in the MCMC statement, the following “Initial Parameter Estimates for MCMC” table displays the starting mean and covariance estimates used in MCMC. The same starting estimates are used for the MCMC process 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 to generate different parameter estimates from the EM algorithm for the MCMC process.

Output 9.4.4. Initial Parameter Estimates

```
                                The MI Procedure

                      Initial Parameter Estimates for MCMC

      _TYPE_    _NAME_            Oxygen         RunTime        RunPulse

      MEAN                     47.103766       10.554320      171.382197
      COV       Oxygen         24.549968       -5.726112      -15.926034
      COV       RunTime        -5.726112        1.781407        3.124798
      COV       RunPulse      -15.926034        3.124798       83.164044
```

The following two tables display variance information and parameter estimates from the multiple imputation.

Output 9.4.5. Variance Information

```
                                The MI Procedure

                   Multiple Imputation Variance Information

                    -----------------Variance-----------------
        Variable         Between         Within          Total       DF

        Oxygen          0.009200       0.987880       1.000148   27.778
        RunTime         0.002255       0.069112       0.072119   26.388
        RunPulse        0.043126       3.650388       3.707889   27.653

                    Multiple Imputation Variance Information

                                    Relative       Fraction
                                    Increase        Missing
                    Variable     in Variance    Information

                    Oxygen          0.012418       0.012414
                    RunTime         0.043503       0.043351
                    RunPulse        0.015752       0.015744
```

Output 9.4.6. Parameter Estimates

```
                                The MI Procedure

                    Multiple Imputation Parameter Estimates

   Variable            Mean      Std Error    95% Confidence Limits        DF

   Oxygen         47.198228       1.000074      45.1489      49.2475   27.778
   RunTime        10.510911       0.268549       9.9593      11.0625   26.388
   RunPulse      172.113649       1.925588     168.1670     176.0603   27.653

                    Multiple Imputation Parameter Estimates

                                                             t for H0:
 Variable         Minimum        Maximum            Mu0    Mean=Mu0   Pr > |t|

 Oxygen         47.132351      47.308274      50.000000       -2.80     0.0092
 RunTime        10.456079      10.538446      10.000000        1.90     0.0681
 RunPulse      171.943144     172.344920     180.000000       -4.10     0.0003
```

Example 9.5. 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 the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern. You must specify a VAR list to provide the order of variables for the imputed data to achieve a monotone missing pattern.

```
proc mi data=FitMiss seed=55417 out=outmono;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

Output 9.5.1. Model Information

```
                              The MI Procedure

                             Model Information

     Data Set                             WORK.FITMISS
     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     55417
```

The following “Missing Data Patterns” table 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 9.5.2. Missing Data Pattern

```
                              The MI Procedure

                            Missing Data Patterns

                             Run     Run
      Group     Oxygen     Time    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 only needs to impute three missing values from Group 4 and Group 5 to achieve a monotone missing pattern for the imputed data set.

When using 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 outmono.

```
proc mi data=outmono ( where= (_Imputation_=1) )
   nimpute=0;
   var Oxygen RunTime RunPulse;
run;
```

Output 9.5.3. Monotone Missing Data Pattern

```
                                The MI Procedure

                              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        46.307744       10.861364      171.863636
              2        46.372151       10.053333               .
              3        52.461667               .               .
```

The following statements impute one value for each missing value in the monotone missingness data set outmono. The variable _Imputation_ is renamed to Impute so that it will not be overwritten by the the new variable _Imputation_ being created in the MI procedure.

```
proc mi data=outmono( rename=(_Imputation_=Impute))
        nimpute=1 seed=43672
        out=outds( rename=(Impute=_Imputation_) drop=_Imputation_);
   monotone method=reg;
   var Oxygen RunTime RunPulse;
   by Impute;
run;
```

The variable Impute is renamed to _Imputation_ in the output data outds. This makes the output data set have the same structure as output data sets generated from other imputation methods. You can then analyze these data sets by using other SAS procedures and combine these results by using the procedure MIANALYZE. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

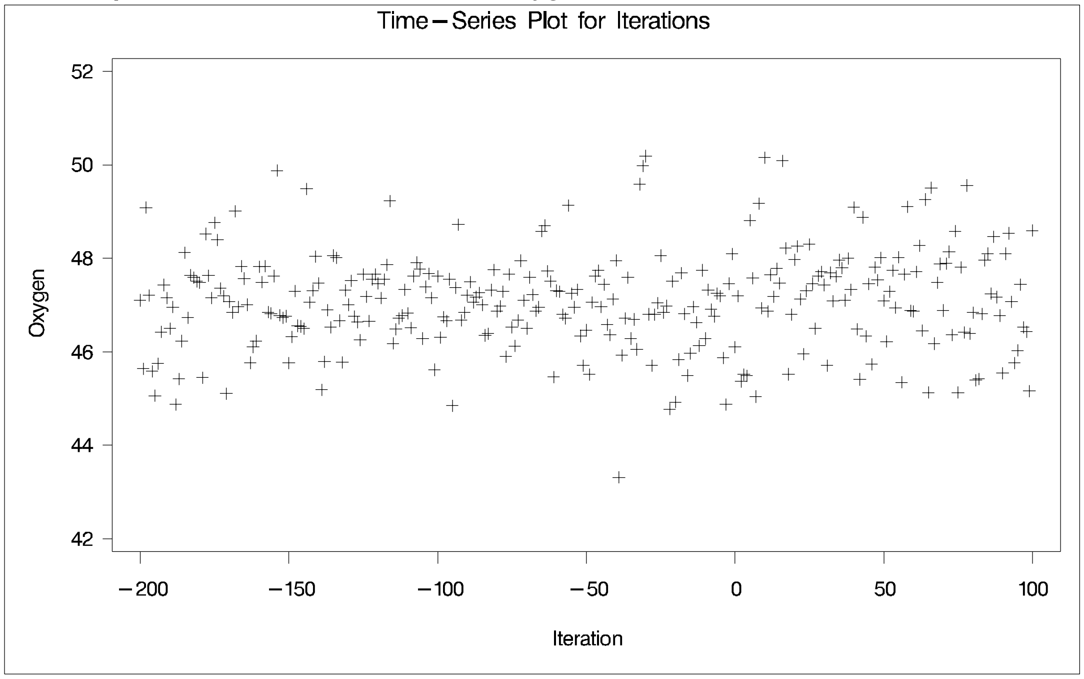
Example 9.6. Checking Convergence in MCMC

This example uses the MCMC method with a single chain. It also displays time-series 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. Note that iterations during the burn-in period are indicated with negative iteration numbers. These statements also create an autocorrelation function plot for the variable Oxygen.

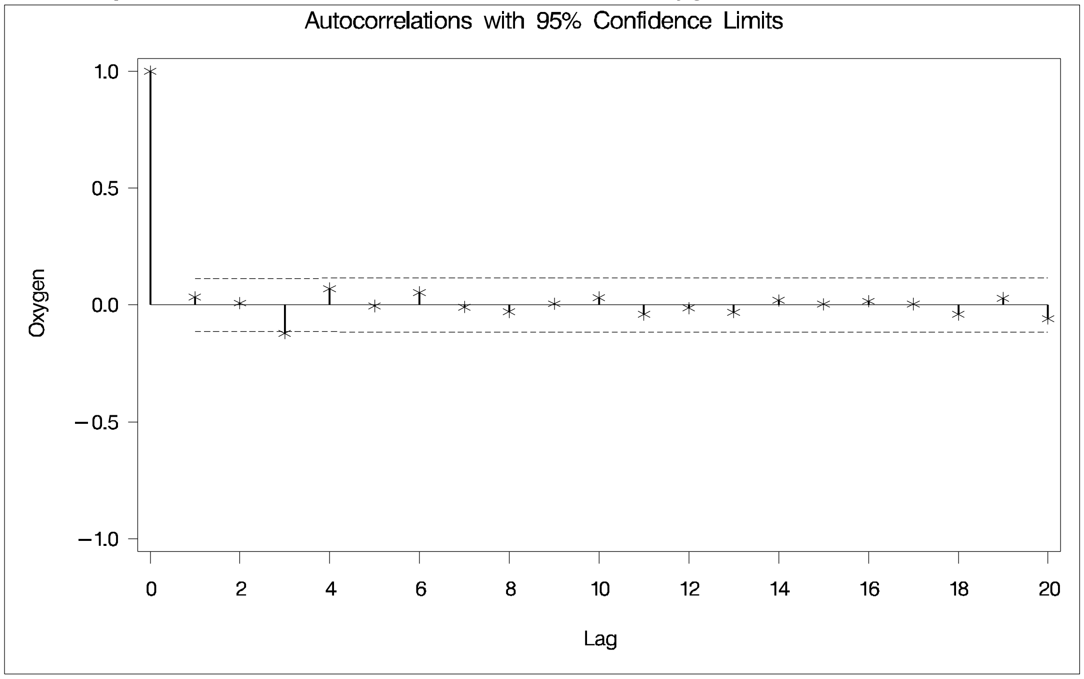
```
proc mi data=FitMiss seed=37921 noprint nimpute=2;
   mcmc timeplot(mean(Oxygen)) acfplot(mean(Oxygen));
   var Oxygen RunTime RunPulse;
run;
```

Output 9.6.1. Time-Series Plot for Oxygen



By default, the MI procedure uses the plus sign (+) as the plot symbol to display the points in the plot. The time-series plot shows no apparent trends for the variable Oxygen.

Output 9.6.2. Autocorrelation Function Plot for Oxygen

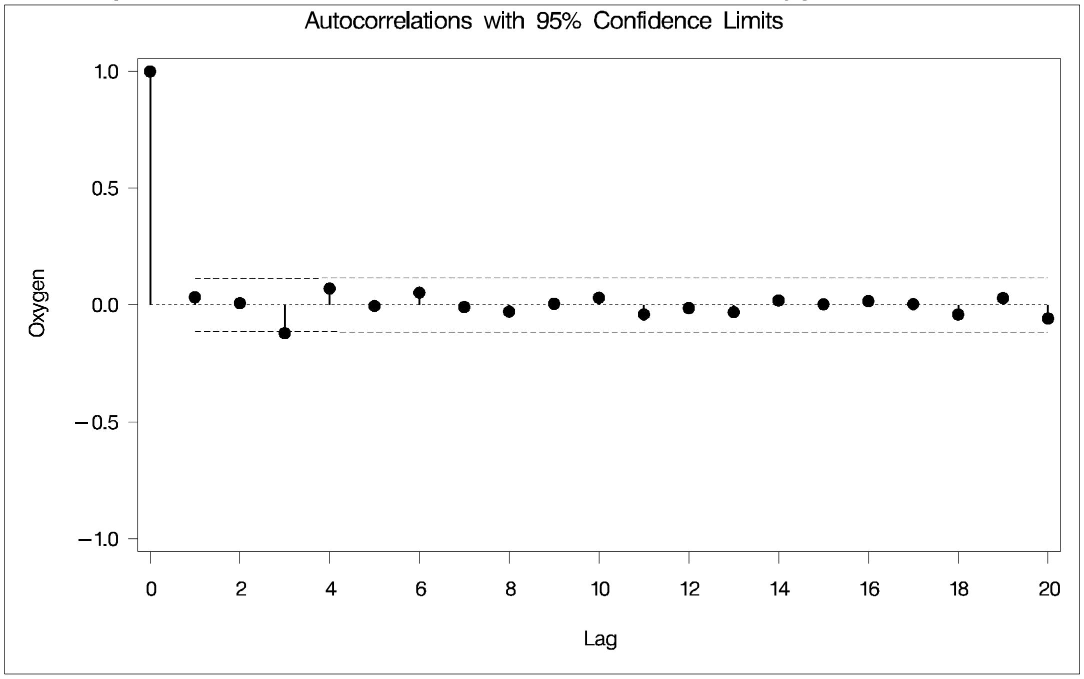


By default, the MI procedure uses the star sign (*) as the plot symbol to display the points in the plot, a solid line to display the reference line of zero autocorrelation, and a pair of dashed lines to display approximately 95% confidence limits for the autocorrelations. The autocorrelation function plot shows no significant positive or negative autocorrelation.

The following statements use display options to modify the autocorrelation function plot for Oxygen.

```
proc mi data=FitMiss seed=37921 noprint nimpute=2;
   mcmc acfplot(mean(Oxygen) / symbol=dot lref=2);
   var Oxygen RunTime RunPulse;
run;
```

Output 9.6.3. Modified 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 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 time-series analysis of the iterations with other procedures, such as PROC AUTOREG and PROC ARIMA in the *SAS/ETS User's Guide, Version 8*.

Example 9.7. Transformation to Normality

This example applies the MCMC method to the FitMiss 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=FitMiss seed=37921 mu0=50 10 180 out=outmi;
   transform log(Oxygen);
   mcmc chain=multiple displayinit;
   var Oxygen RunTime RunPulse;
run;
```

The following "Missing Data Patterns" table lists distinct missing data patterns with corresponding statistics for the FitMiss data. Note that the values of Oxygen shown in the tables are transformed values.

Output 9.7.1. Missing Data Pattern

```
                              The MI Procedure

                            Missing Data Patterns

                              Run     Run
          Group    Oxygen     Time    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 following “Variable Transformations” table lists the variables that have been transformed.

Output 9.7.2. Missing Data Pattern

```
                              The MI Procedure

                          Variable Transformations

                          Variable    _Transform_

                          Oxygen      LOG
```

The following “Initial Parameter Estimates for MCMC” table displays the starting mean and covariance estimates used in the MCMC process.

Output 9.7.3. Initial Parameter Estimates

```
                              The MI Procedure

                    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        3.011179
       COV       RunPulse        -0.328772        3.011179       82.747608

                       Transformed Variables: Oxygen
```

The following table displays variance information from the multiple imputation.

Output 9.7.4. Variance Information

```
                                The MI Procedure

                    Multiple Imputation Variance Information

                     -----------------Variance-----------------
         Variable         Between         Within          Total       DF

       * Oxygen       0.000004541       0.000398       0.000404   27.766
         RunTime         0.000814       0.063128       0.064105   27.708
         RunPulse        0.182700       3.498974       3.718214   25.923

                            * Transformed Variables

                    Multiple Imputation Variance Information

                                  Relative       Fraction
                                  Increase        Missing
                     Variable     in Variance    Information

                   * Oxygen          0.013685       0.013590
                     RunTime         0.015478       0.015356
                     RunPulse        0.062658       0.060595

                            * Transformed Variables
```

The following table displays parameter estimates from the multiple imputation. Note that the parameter value of Mu0 has also been transformed using the logarithmic transformation.

Output 9.7.5. Parameter Estimates

```
                                The MI Procedure

                     Multiple Imputation Parameter Estimates

    Variable             Mean      Std Error    95% Confidence Limits        DF

  * Oxygen           3.845991       0.020091       3.8048       3.8872   27.766
    RunTime         10.586242       0.253190      10.0674      11.1051   27.708
    RunPulse       170.849654       1.928267     166.8855     174.8138   25.923

                             * Transformed Variables

                    Multiple Imputation Parameter Estimates

                                                              t for H0:
  Variable          Minimum        Maximum            Mu0     Mean=Mu0   Pr > |t|

* Oxygen           3.843860       3.848775       3.912023        -3.29     0.0028
  RunTime         10.547440      10.616746      10.000000         2.32     0.0282
  RunPulse       170.315955     171.324638     180.000000        -4.75     <.0001

                            * Transformed Variables
```

The following statements list the first ten observations of the data set outmi. Note that the values for Oxygen are in the original scale.

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

Output 9.7.6. Imputed Data Set in Original Scale

```
                 First 10 Observations of the Imputed Data Set

                                                                 Run
           Obs    _Imputation_     Oxygen    RunTime     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     8.4840    155.503
             5          1         49.8740     9.2200    166.031
             6          1         44.8110    11.6300    176.000
             7          1         43.4130    11.9500    176.000
             8          1         44.6435    10.8500    173.761
             9          1         39.4420    13.0800    174.000
            10          1         60.0550     8.6300    170.000
```

The preceding results can also be produced from the following statements without using a TRANSFORM statement.

```
data temp;
   set FitMiss;
   LogOxygen= log(Oxygen);
run;

proc mi data=temp seed=37921 mu0=3.91202 10 180 out=outtemp;
   mcmc chain=multiple displayinit;
   var LogOxygen RunTime RunPulse;
run;

data outmi;
   set outtemp;
   Oxygen= exp(LogOxygen);
run;
```

Note that a transformed value of log(50)=3.91202 is used in the MU0= option.

Example 9.8. Saving and Using Parameters for MCMC

This example uses the MCMC method with multiple chains as specified in Example 9.4. It saves the parameter values used for each imputation in an output data set of type EST. 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=FitMiss seed=55417 nimpute=3 mu0=50 10 180 noprint;
   mcmc chain=multiple outest=miest;
   var Oxygen RunTime RunPulse;
run;
```

The following statements list the parameters used for the imputations. Note that the data set includes observations with _TYPE_=‘SEED’ containing the seed to start the next random number generator.

```
proc print data=miest;
   title 'Parameters for the Imputations';
run;
```

Output 9.8.1. OUTEST Data Set

```
                          Parameters for the Imputations

Obs _Imputation_ _TYPE_  _NAME_          Oxygen        RunTime       RunPulse

  1       1       SEED             2099769086.00  2099769086.00  2099769086.00
  2       1       PARM                     49.31          10.00         172.19
  3       1       COV    Oxygen            32.05          -7.47         -28.32
  4       1       COV    RunTime           -7.47           2.41           6.75
  5       1       COV    RunPulse         -28.32           6.75         128.61
  6       2       SEED              419117425.00   419117425.00   419117425.00
  7       2       PARM                     47.49          10.43         171.58
  8       2       COV    Oxygen            41.02          -8.60         -34.29
  9       2       COV    RunTime           -8.60           2.25           7.61
 10       2       COV    RunPulse         -34.29           7.61         142.94
 11       3       SEED              535522494.00   535522494.00   535522494.00
 12       3       PARM                     45.98          10.82         172.45
 13       3       COV    Oxygen            43.24          -9.90           8.14
 14       3       COV    RunTime           -9.90           2.75          -2.72
 15       3       COV    RunPulse           8.14          -2.72         218.32
```

The following statements invoke the MI procedure and use the INEST= option in the MCMC statement.

```
proc mi data=FitMiss;
   mcmc inest=miest;
   var Oxygen RunTime RunPulse;
run;
```

Output 9.8.2. Model Information

```
                         The MI Procedure

                        Model Information

           Data Set                           WORK.FITMISS
           Method                             MCMC
           INEST Data Set                     WORK.MIEST
           Number of Imputations              3
```

The remaining tables for the example are identical to the tables in Example 9.4.

References

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

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

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

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.

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.

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. (1987), *Statistical Analysis with Missing Data*, New York: John Wiley & Sons, Inc.

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, Inc.

Rosenbaum, P.R. and Rubin, D.B. (1983), “The Central Role of the Propensity Score in Observational Studies for Causal Effects,” *Biometrika*, 70, 41–55.

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

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

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

SAS Institute Inc. (1999), *SAS/ETS User’s Guide, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS/GRAPH Software: Reference, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Language Reference: Concepts, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS/STAT User’s Guide, Version 8*, Cary, NC: SAS Institute Inc.

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

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.