

Chapter 11

The MI Procedure

Chapter Table of Contents

OVERVIEW	3
GETTING STARTED	4
SYNTAX	8
BY Statement	11
FREQ Statement	12
MULTINORMAL Statement	12
VAR Statement	15
DETAILS	15
Descriptive Statistics	15
Statistical Assumptions for Using MI	16
Imputation Mechanisms	17
Regression Method for Monotone Missing Data	18
Propensity Score Method for Monotone Missing Data	19
MCMC Method for Arbitrary Missing Data	20
Input Data Sets	26
Output Data Sets	27
Combining Inferences from Multiply-Imputed Data Sets	28
Multiple Imputation Efficiency	29
Imputer's Model Versus Analyst's Model	30
Parameter Simulation Versus Multiple Imputation	31
ODS Table Names	31
EXAMPLES	32
Example 1. Propensity Score Method	32
Example 2. Regression Method	36
Example 3. MCMC Method	38
Example 4: Saving and Using Parameter Values for MCMC	40
Example 5. Using EM on Bootstrap Resampling for MCMC	42
Example 6: Checking Convergence in A Single Chain	43
REFERENCES	46

Chapter 11

The MI Procedure

Overview

The MI procedure is a new, experimental procedure for multiple imputation of missing data, which is available in Release 8.1 of the SAS System. Missing values are an issue in a substantial number of statistical analyses. Most SAS statistical procedures exclude observations with any missing variable values from the analysis. These observations are called incomplete cases. While using only complete cases has its simplicity, you lose the information that is contained in the incomplete cases. This approach also ignores possible systematic differences between the complete cases and 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, PROC CORR 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, in which you substitute a value for each missing value. Standard statistical procedures for complete data analysis can then be used with the filled-in data set. For example, each missing value can be imputed 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. 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 by 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. Once the m complete data sets are analyzed by 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 8.1 of the SAS System.

Often, as few as three to five imputations are adequate in multiple imputation (Rubin 1996, p. 480). While this seems counterintuitive, 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 29.

Multiple imputation inferences assume 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 (imputer’s model). But in practice, the two models may not be the same. The consequence for different scenarios (Schafer 1997, pp. 139-143) will be discussed in the “Imputer’s Model Versus Analyst’s Model” section on page 30.

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 produce a simulation-based method of parameter simulation, the choice between the two methods (Schafer 1997, pp. 89-90, 135-136) will be examined in the “Parameter Simulation Versus Multiple Imputation” section on page 31.

Getting Started

Consider the following Fitness data set which 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      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      .       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
;

```

Suppose that the data are multivariate normally distributed and the missing data are missing at random. The following statements invoke the MI procedure and impute missing values for the FitMiss data set.

```

proc mi data=FitMiss noprint out=outmi seed=37851;
  var Oxygen RunTime RunPulse;
run;

```

Output 11.0.0. Model Information

The MI Procedure	
Model Information	
Data Set	WORK.FITMISS
Method	MCMC
Multiple Imputation Chain	Multiple Chains
Initial Estimates for MCMC	EM
Start	Starting Value
Prior	Jeffreys
Number of Imputations	5
Number of Burn-in Iterations	200
Seed for random number generator	37851

The “Model Information” table describes the method used in the multiple imputation process. By default, the procedure uses the MCMC method with multiple chains to create five imputations. Also, the procedure takes 200 burn-in iterations before the imputation. The burn-in iterations are used to establish convergence to the stationary distribution.

Output 11.0.1. Missing Data Patterns

Missing Data Patterns					
Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	.	5	16.13
3	X	.	.	3	9.68
4	.	X	X	2	6.45

Missing Data Patterns			
-----Group Means-----			
Group	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.505800	10.280000	.
3	52.461667	.	.
4	.	10.435000	161.000000

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.

Output 11.0.2. Initial Parameter Estimates

Initial Parameter Estimates for MCMC					
IMPUTATION	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	MEAN		47.259052	10.560270	170.146102
1	COV	Oxygen	25.022733	-5.743303	-21.939931
1	COV	RunTime	-5.743303	1.764589	4.585643
1	COV	RunPulse	-21.939931	4.585643	103.355140

The “Initial Parameter Estimates for MCMC” table displays the starting mean and covariance estimates used in each imputation. By default, the MI procedure uses the parameter estimates for a posterior mode, that is, the parameter estimates with the highest observed-data posterior density.

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 in the “Combining Inferences from Multiply-Imputed Data Sets” section on page 28.

Output 11.0.3. Variance Information

Multiple-Imputation Variance Information				
Variable	-----Variance-----			DF
	Between	Within	Total	
Oxygen	0.018986	0.955902	0.978685	27.424
RunTime	0.001868	0.072048	0.074289	27.163
RunPulse	3.087150	4.035638	7.740219	7.9793

Multiple-Imputation Variance Information			
Variable	Relative Increase in Variance	Fraction Missing Information	
		Oxygen	0.023834
RunTime	0.031109	0.030611	
RunPulse	0.917966	0.529576	

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.

Output 11.0.4. Parameter Estimates

Multiple-Imputation Parameter Estimates					
Variable	Mean	Std Error	95% Confidence Limits		DF
		Mean			
Oxygen	47.365836	0.989285	45.3375	49.3942	27.424
RunTime	10.514422	0.272560	9.9553	11.0735	27.163
RunPulse	170.407932	2.782125	163.9894	176.8264	7.9793

Multiple-Imputation Parameter Estimates					
Variable	Minimum	Maximum	Mu0	t for H0:	
				Mean=Mu0	Pr > t
Oxygen	47.221641	47.540153	0	47.88	<.0001
RunTime	10.439644	10.551335	0	38.58	<.0001
RunPulse	167.919562	172.014589	0	61.25	<.0001

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 variable `_Imputation_` indicating the imputation numbers. The data set can now be analyzed by using standard statistical procedures with the BY variable `_Imputation_`.

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;
```

Output 11.0.5. Imputed Data Set

First 10 Observations of the Imputed Data Set				
Obs	_Imputation_	Oxygen	RunTime	Run Pulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	7.5363	172.631
5	1	49.8740	9.2200	181.680
6	1	44.8110	11.6300	176.000
7	1	42.9519	11.9500	176.000
8	1	49.0910	10.8500	176.028
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

Syntax

The following statements are available in PROC MI.

```
PROC MI < options > ;
  BY variables ;
  FREQ variable ;
  MULTINORMAL < options > ;
  VAR variables ;
```

The rest of this section gives detailed syntax information for each of these statements, beginning with the PROC MI statement. The remaining statements are in alphabetical order.

The PROC MI statement is the only required statement for the MI procedure.

```
PROC MI < options > ;
```

The following tables summarize the options available in the PROC MI statement.

Table 11.1. Summary of PROC MI Options

Tasks	Options
Specify data sets	
input data set	DATA=
output data set with imputed values	OUT=
input parameter estimates for imputations	INEST=
output parameter estimates used in imputations	OUTEST=
output parameter estimates used in iterations	OUTITER=
Specify imputation details	
number of imputations	NIMPU=
seed to begin random number generator	SEED=
singularity tolerance	SINGULAR=
units to round imputed variable values	ROUND=
minimum values for imputed variable values	MINIMUM=
maximum values for imputed variable values	MAXIMUM=
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=*p*

specifies that confidence limits be constructed for the mean estimates with confidence level $100(1 - p)\%$, where $0 < p < 1$. The default is $p = .05$.

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.

INEST=*SAS-data-set*

names the 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 section “Input Data Sets”.

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 need to 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 for each variable.

The `MAXIMUM=` option is related to the `MINIMUM=` and `ROUND=` options, which are used to make the imputed values more consistent with the observed variable values. These options are not valid for `METHOD=PROPENSITY` in the `MULTINORMAL` 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 examples, the option “`MAXIMUM= 100 .`” sets a maximum of 100 for the first analysis variable only and no maximum for the remaining variables. The option “`MAXIMUM= . 100`” 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 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 for each variable.

MU0=numbers

LOCATION=numbers

specifies the parameter values μ_0 under the null hypothesis $\mu = \mu_0$ for the population means corresponding to the analysis variables. Each hypothesis is tested with a t test. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a `VAR` statement, and the specified numbers correspond to variables in the `VAR` statement. The default is `MU0=0`.

NIMPU=number

IMPU=number

specifies the number of imputations. The default is `NIMPU=5`.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 16, “Using the Output Delivery System” for more information.

OUT=SAS-data-set

creates an output SAS data set for imputation results. The data set includes an identification 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.

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*.

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.

OUTITER=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.

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 correspond to variables in the VAR statement. A missing or zero value indicates no rounding for imputed variables.

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 be able to duplicate the result 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 exactly the same seed number in the future to reproduce the same 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 15.

SINGULAR= p

specifies the criterion for determining the singularity of a covariance matrix, where $0 < p < 1$. The default is SINGULAR=1E-4.

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 , $j = 1, \dots, v$, 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*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

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's name in a FREQ statement. PROC MI then treats the data set as if each observation appeared 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.

MULTINORMAL Statement

MULTINORMAL < *options* > ;

The MULTINORMAL statement specifies the imputation method. You can specify the following options in a MULTINORMAL statement.

METHOD=REGRESSION

METHOD=PROPENSITY < (**NGROUPS** | **GROUPS = number**) >

METHOD=MCMC < (*options*) >

The default is METHOD=MCMC. If no MULTINORMAL statement is included, METHOD=MCMC is used. 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

GROUPS=number

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

The following table summarizes the options available for METHOD=MCMC.

Table 11.2. Summary of Options for METHOD=MCMC

Tasks	Options
Specify imputation details	
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=
starting parameters	START=
prior parameter information	PRIOR=

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

CHAIN=SINGLE | MULTIPLE

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

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.

With INITIAL=EM, PROC MI uses the means and standard deviations from available cases as the initial estimates for the expectation-maximization (EM) algorithm. The correlations are set to zero. 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 , $0 < p \leq 1$, 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 gives an overdispersed initial estimate that provides possible different starting values for the MCMC. If you specify the BOOTSTRAP option without the number of observations, $p=0.75$ 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. This convergence criterion is used in parameter estimation for a single model fit.

MAXITER=*number*

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

FUNCTION=MLE**FUNCTION=POSTERIOR**

specifies the objective function used in the EM algorithm. You can use either FUNCTION=MLE to compute the maximum likelihood estimates for parameters or FUNCTION=POSTERIOR to compute parameter estimates for a posterior mode, the highest observed-data posterior density. The default is FUNCTION=POSTERIOR.

NBITER=*number***BITER=*number***

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

NITER=*number***ITER=*number***

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

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 22 and the “Posterior Step” section on page 23. If you do not specify the PRIOR= option, the default is PRIOR=JEFFREYS.

The PRIOR=INPUT= option specifies a SAS data set from which the prior information of the mean vector and the covariance matrix is read. The data set must be of TYPE=COV. See the “Input Data Sets” section on page 26 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.

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 have more than one number specified 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. Suppose n_0 is the number of observations fully observed, and n_j is the number of observations with observed values for variable Y_j .

With complete cases, the sample mean vector is

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

and the CSSCP matrix is

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

where each summation is over the fully observed observations.

The sample covariance matrix is

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

and is an unbiased estimate of the covariance matrix.

The correlation matrix \mathbf{R} 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 sample mean for variable Y_j is

$$\bar{y}_j = \frac{1}{n_j} \sum y_{ji}$$

and its corrected sum of squares is

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

where each summation is over observations with observed values for variable Y_j .

The variance is

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

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

Statistical Assumptions for Using MI

The MI procedure assumes that the data are from a multivariate continuous distribution and contain missing values that may 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 multiple imputation procedures assume that the missing data are missing at random (MAR), that is, the probability that an observation is missing may 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 \mathbf{Y} are missing or observed. Then the MAR assumption is that the distribution of \mathbf{R} may depend on \mathbf{Y}_{obs} but not on \mathbf{Y}_{mis} .

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

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 may 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, MI and MIANALYZE procedures assume that the parameters θ of the data model and the parameters ϕ of the model for the missing data indicators are distinct. That is, knowing the values of θ does not provide any additional information about ϕ , and vice versa. If both 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).

Imputation Mechanisms

This section describes three methods that are available in the MI procedure. The method of choice depends on the type of missing data pattern.

- For monotone missing data 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 an arbitrary missing data pattern, a Markov Chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality can be used.

A data set with variables Y_1, Y_2, \dots, 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. With a monotone missing data pattern, you have greater flexibility in your choice of strategies. For example, you can implement a regression model without involving iterations as in MCMC.

With an arbitrary missing data pattern, you can often use the MCMC method, which creates multiple imputations by drawing simulations from a Bayesian prediction 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 still being researched.

Although the regression or MCMC method assumes 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 nonnormal, as supported by simulation studies described in Schafer (1997) and the original references therein.

Li (1988) presented an argument for convergence of the MCMC in the continuous case and used it to create imputation for incomplete multivariate continuous data. PROC MI generates statistics that you can use to check for convergence. The details are described later in the section “Convergence in MCMC”.

Regression Method for Monotone Missing Data

In this 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

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

is fitted using nonmissing observations.

The fitted model has the regression parameter estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_{(j-1)})$ 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 $Y_1, Y_2, \dots, Y_{(j-1)}$.

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

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

where $y_1, y_2, \dots, y_{(j-1)}$ 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, propensity scores are generated for all observations to estimate the probabilities that each 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).

With a monotone missing pattern, the following steps are used 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

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

where $p_j = Pr(R_j = 0 | Y_1, Y_2, \dots, Y_{(j-1)})$ 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 assume 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 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, but it is not appropriate for analyses involving relationship among variables. 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 pseudorandom draws from multidimensional and otherwise intractable probability distributions via Markov chains. A Markov chain is a sequence of random variables in which the distribution of each element depends only on the value of the previous 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, it 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

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

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 $p(\theta|Y_{obs})$ 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 $p(\theta|Y_{obs}, Y_{mis})$ 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 may 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 $\theta^{(t)}$ at the t th iteration, the I-step draws $Y_{mis}^{(t+1)}$ from $p(Y_{mis}|Y_{obs}, \theta^{(t)})$ and the P-step draws $\theta^{(t+1)}$ from $p(\theta|Y_{obs}, Y_{mis}^{(t+1)})$.

This creates a Markov chain

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

which converges in distribution to $p(Y_{mis}, \theta|Y_{obs})$. 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 verify them by repeating the process with different random number generators and starting values based on different initial parameter estimates.

Imputation Step

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

Suppose $\mu = [\mu'_1, \mu'_2]'$, is the partitioned mean vector of two sets of variables, Y_{obs} and Y_{mis} , where μ_1 is the mean vector for variables Y_{obs} and μ_2 is the mean vector for variables Y_{mis} .

Also suppose

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

is the partitioned covariance matrix for these variables, where Σ_{11} is the covariance matrix for variables Y_{obs} , Σ_{22} is the covariance matrix for variables Y_{mis} , and Σ_{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 Σ_{11} submatrix, the matrix becomes

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

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

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

$$\mu_{2.1} = \mu_2 + \Sigma'_{12}\Sigma_{11}^{-1}(\mathbf{y}_1 - \mu_1)$$

and the conditional covariance matrix

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

Bayesian Estimation of the Mean Vector and Covariance Matrix

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

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

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

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

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

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

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

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

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

$$\begin{aligned} \mathbf{\Sigma} &\sim W^{-1}(m, \mathbf{\Psi}) \\ \mu | \mathbf{\Sigma} &\sim N(\mu_0, \frac{1}{\tau} \mathbf{\Sigma}) \end{aligned}$$

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

$$\begin{aligned} \mathbf{\Sigma} | \mathbf{Y} &\sim W^{-1}(n + m, (n - 1)\mathbf{S} + \mathbf{\Psi} + \frac{n\tau}{n + \tau}(\bar{\mathbf{y}} - \mu_0)(\bar{\mathbf{y}} - \mu_0)') \\ \mu | (\mathbf{\Sigma}, \mathbf{Y}) &\sim N(\frac{1}{n + \tau}(n\bar{\mathbf{y}} + \tau\mu_0), \frac{1}{n + \tau}\mathbf{\Sigma}) \end{aligned}$$

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

Posterior Step

In each iteration, the posterior step simulates the posterior population mean vector μ and covariance matrix Σ from prior information for μ and Σ , 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 Σ in the data set. Optionally, it also provides a prior information for μ in the data set.
- PRIOR=RIDGE=, which uses a ridge prior.

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

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

2. An Informative Prior for μ and Σ

When prior information is available for the parameters μ and Σ , you can provide it with a SAS data set which you specify with the PRIOR=INPUT= option.

$$\begin{aligned}\Sigma &\sim W^{-1}(d^*, d^*\mathbf{S}^*) \\ \mu|\Sigma &\sim N(\mu_0, \frac{1}{n_0}\Sigma)\end{aligned}$$

To obtain the prior distribution for Σ , 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 μ_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 previous section “Bayesian Estimation of the Mean Vector and Covariance Matrix”, is given by

$$\begin{aligned}\Sigma^{(t+1)}|\mathbf{Y} &\sim W^{-1}(n+d^*, (n-1)\mathbf{S} + d^*\mathbf{S}^* + \mathbf{S}_m) \\ \mu^{(t+1)}|(\Sigma^{(t+1)}, \mathbf{Y}) &\sim N\left(\frac{1}{n+n_0}(n\bar{\mathbf{y}} + n_0\mu_0), \frac{1}{n+n_0}\Sigma^{(t+1)}\right)\end{aligned}$$

where

$$\mathbf{S}_m = \frac{nn_0}{n+n_0}(\bar{\mathbf{y}} - \mu_0)(\bar{\mathbf{y}} - \mu_0)'$$

3. An Informative Prior for Σ

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

To obtain the prior distribution for Σ , 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 Σ will be used.

Corresponding to the prior for Σ

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

the posterior distribution for Σ (Anderson 1984, p. 269) is

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

Thus, an estimate of Σ is given by the weighted average

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

and the posterior distribution for Σ and μ becomes

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

4. A Ridge Prior

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

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

The posterior is then given by

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

MCMC Method Specifications

With MCMC, you can use either a single chain for all imputations (CHAIN=SINGLE) or a separate chain for each imputation (CHAIN=MULTIPLE). 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 for each chain with the NBITER= option. For a single chain, you can also specify the number of iterations between imputations with the NITER= option.

You can explicitly specify initial parameter estimates for the MCMC process with the INITIAL=INPUT= data set option. Or you can use the EM algorithm to derive a set of initial parameter estimates 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) in the first iteration step of each chain.

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. The data set can be used as the reference distribution to impute values in another data set with the INEST= option.

By default, the MCMC method uses five separate chains and completes 200 burn-in iterations before each imputation. The method uses the statistics from the available cases in the data as the initial estimates for the EM algorithm to compute the starting values for the MCMC process. A noninformative prior is used to derive the posterior distribution of the parameters.

EM Specifications

The EM algorithm (Little and Rubin 1987) is a technique that finds maximum likelihood estimates for parametric models for incomplete data. You can use either the maximum likelihood estimates (FUNCTION=MLE) or a posterior mode (FUNCTION=POSTERIOR), the parameter estimates with the highest observed-data posterior density. The resulting EM estimate provides a good starting value with which to begin the MCMC process.

With INITIAL=EM, 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. Refer to Schafer (1997, p. 169) for a discussion of suggested starting values for the algorithm.

You can also use overdispersed starting values for MCMC with the BOOTSTRAP= option. With the BOOTSTRAP option, PROC MI uses a simple random sample with replacement from the input data set to derive the initial estimates for each chain (Schafer 1997, p. 128).

You can specify the convergence criterion with the CONVERGE= option. 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.

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 the convergence is not a simple matter and can not 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. 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).

You can use EM on bootstrap resampling 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.

Input Data Sets
DATA= data set

The input `DATA=` data set is an ordinary SAS data set containing multivariate data with missing values.

INEST= 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=` dataset.

INITIAL=INPUT= 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.

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'`.

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).

PRIOR=INPUT= 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 Σ of the form

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

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

$$\mu \sim N\left(\mu_0, \frac{1}{n_0} \Sigma\right)$$

PROC MI reads the mean vector μ_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**OUT= data set**

The OUT= data set contains all the variables in the original data set and a new variable named `_Imputation_` which 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.

OUTEST= 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 identification 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.

OUTITER= data set

The OUTITER= data set 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 observations with `_TYPE_='MEAN'` contain the mean parameters, the observations with `_TYPE_='STD'` contain the standard deviations, and the observations with `_TYPE_='COV'` contain the covariance parameters used in the imputation step.

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. Let \hat{Q}_i and \hat{U}_i be the point and variance estimates from the i -th imputed data set, $i=1, 2, \dots, m$. Then the combined point estimate for Q from multiple imputation is the average of the m complete-data estimates:

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

Let \bar{U} be the within-imputation variance, which is the average of the m complete-data estimates:

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

and B be the between-imputation variance

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

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

$$T = \bar{U} + \left(1 + \frac{1}{m}\right)B$$

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

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

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^* , in the inferences.

The degrees of freedom v_m depends on m and the ratio

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

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 $(Q - \bar{Q})T^{-(1/2)}$ 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).

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

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 11.3. Relative Efficiency

m	λ				
	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) comprehensively discusses this topic, and the following discussion is largely based on his work.

Multiple imputation inferences assume 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 (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 may 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 $Y_3 = Y_1 Y_2$, but the analyst later fits a model $Y_3 = Y_1$. 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, one simulates random values of parameters from its observed-data posterior distribution and makes simple inferences about these parameters (Schafer 1997, p. 89).

When a set of well defined population parameters θ are of interest, parameter simulation can be used to directly examine and summarize simulated values of θ . It usually requires a large number of iterations, and involves calculating appropriate summaries of the resulting dependent sample of the iterates of the θ . If only a small set of parameters are involved, parameter simulation may 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 may be cheaper than generating and storing a large number 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 θ 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, see Chapter 16, “Using the Output Delivery System.”

Table 11.4. ODS Tables Produced in PROC MI

ODS Table Name	Description	Option
ModelInfo	Model information	
MissingPatn	Missing data patterns	
UnivAC	Univariate statistics for available cases	SIMPLE
CorrAC	Pairwise correlations for available cases	SIMPLE
InitParm	Initial parameter estimates for 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 examples with the regression method and the propensity score method. 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
;
```

The FitMiss data set created in the "Getting Started" section is used in examples with the MCMC method.

Example 1. 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;
  multinormal method=propensity;
  var Oxygen RunTime RunPulse;
run;
```

The procedure generates the following output:

Output 11.0.6. 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 11.0.7. Missing Data Patterns

Missing Data Patterns					
Group	Oxygen	Run Time	Run 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	-----Group Means-----		
	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 variable means in each group are also displayed. 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 (Available Cases)” table and correlations from available pairwise cases in the “ Pairwise Correlations (Available Cases)” table.

Output 11.0.8. Univariate Statistics

Univariate Statistics (Available Cases)					
Variable	N	Mean	Std Dev	Minimum	Maximum
Oxygen	31	47.37581	5.32723	37.38800	60.05500
RunTime	28	10.68821	1.37988	8.63000	14.03000
RunPulse	23	170.73913	11.28240	146.00000	186.00000

Output 11.0.9. Pairwise Correlations

Pairwise Correlations (Available Cases)			
	Oxygen	RunTime	RunPulse
Oxygen	1.000000000	-0.846153138	-0.437864647
RunTime	-0.846153138	1.000000000	0.339593750
RunPulse	-0.437864647	0.339593750	1.000000000

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.

Output 11.0.10. Variance Information

Multiple-Imputation Variance Information				
Variable	-----Variance-----			DF
	Between	Within	Total	
RunTime	0.001068	0.059100	0.060382	27.498
RunPulse	1.147555	4.686646	6.063711	17.006
Multiple-Imputation Variance Information				
Variable	Relative Increase in Variance	Fraction Missing Information		
RunTime	0.021688	0.021448		
RunPulse	0.293828	0.246288		

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.

Output 11.0.11. Parameter Estimates

Multiple-Imputation Parameter Estimates					
Variable	Mean	Std Error Mean	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					
Variable	Minimum	Maximum	Mu0	t for H0: 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 11.0.12. Imputed Data Set

First 10 Observations of the Imputed Data Set				
Obs	_Imputation_	Oxygen	Run Time	Run 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 2. 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;
      multinormal 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 m imputations, 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 missing and the fraction of missing information for each variable are also displayed. A detailed description of these statistics is in the “Combining Inferences from Multiply-Imputed Data Sets” section.

Output 11.0.13. Variance Information

Multiple-Imputation Variance Information				
Variable	-----Variance-----			DF
	Between	Within	Total	
RunTime	0.023347	0.082701	0.110717	15.745
RunPulse	9.309053	7.878363	19.049227	5.8218
Multiple-Imputation Variance Information				
Variable	Relative Increase in Variance	Fraction Missing Information		
RunTime	0.338768	0.275863		
RunPulse	1.417917	0.642953		

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 11.0.14. Parameter Estimates

Multiple-Imputation Parameter Estimates					
Variable	Mean	Std Error Mean	95% Confidence Limits		DF
RunTime	10.590387	0.332742	9.8841	11.2967	15.745
RunPulse	170.516129	4.364542	159.7567	181.2755	5.8218

Multiple-Imputation Parameter Estimates					
Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > t
RunTime	10.420968	10.825806	10.000000	1.77	0.0953
RunPulse	167.838710	174.032258	150.000000	4.70	0.0036

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 11.0.15. Imputed Data Set

First 10 Observations of the Imputed Data Set				
Obs	_Imputation_	Oxygen	Run Time	Run 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	4.67	188
5	1	49.874	9.22	182
6	1	44.811	11.63	176
7	1	45.681	11.95	176
8	1	49.091	10.85	177
9	1	39.442	13.08	174
10	1	60.055	8.63	170

Example 3. 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 nimpu=3;
  multinormal method=mcmc;
  var Oxygen RunTime RunPulse;
run;
```

Output 11.0.16. Model Information

The MI Procedure	
Model Information	
Data Set	WORK.FITMISS
Method	MCMC
Multiple Imputation Chain	Multiple Chains
Initial Estimates for MCMC	EM
Start	Starting Value
Prior	Jeffreys
Number of Imputations	5
Number of Burn-in Iterations	200
Seed for random number generator	55417

By default, the procedure uses multiple chains and completes 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 the statistics from the available cases in the data as the initial estimates for the EM algorithm. That is, it uses the available cases for the means and standard deviations. The correlations are set to zero, and the covariance matrix is constructed from these standard deviations and zero correlations.

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

Output 11.0.17. Missing Data Pattern

Missing Data Patterns					
Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	.	5	16.13
3	X	.	.	3	9.68
4	.	X	X	2	6.45

Missing Data Patterns			
Group	-----Group Means-----		
	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.505800	10.280000	.
3	52.461667	.	.
4	.	10.435000	161.000000

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 11.0.18. Initial Parameter Estimates

Initial Parameter Estimates for MCMC					
<u>_IMPUTATION_</u>	<u>_TYPE_</u>	<u>_NAME_</u>	Oxygen	RunTime	RunPulse
1	MEAN		47.259052	10.560270	170.146102
1	COV	Oxygen	25.022733	-5.743303	-21.939931
1	COV	RunTime	-5.743303	1.764589	4.585643
1	COV	RunPulse	-21.939931	4.585643	103.355140

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

Output 11.0.19. Variance Information

Multiple-Imputation Variance Information				
Variable	-----Variance-----			DF
	Between	Within	Total	
Oxygen	0.006092	0.940079	0.947389	27.953
RunTime	0.001633	0.063477	0.065436	27.172
RunPulse	3.684419	4.441733	8.863035	7.5179

Multiple-Imputation Variance Information			
Variable	Relative Increase in Variance	Fraction Missing Information	
Oxygen	0.007776		0.007746
RunTime	0.030865		0.030375
RunPulse	0.995400		0.551396

Output 11.0.20. Parameter Estimates

Multiple-Imputation Parameter Estimates					
Variable	Mean	Std Error		95% Confidence Limits	DF
		Mean			
Oxygen	47.203772	0.973339		45.2098 49.1977	27.953
RunTime	10.589528	0.255805		10.0648 11.1142	27.172
RunPulse	170.154616	2.977085		163.2120 177.0972	7.5179

Multiple-Imputation Parameter Estimates					
Variable	Minimum	Maximum	t for H0:		
			Mu0	Mean=Mu0	Pr > t
Oxygen	47.107956	47.305244	0	48.50	<.0001
RunTime	10.526707	10.626820	0	41.40	<.0001
RunPulse	167.611862	172.641256	0	57.15	<.0001

Example 4: Saving and Using Parameter Values for MCMC

This example uses the MCMC method with the same specification as in the previous example. It saves the parameter values used for each imputation in an output type EST data set. This output data set is then used to impute missing values in other similar input data sets.

The following statements invoke the MI procedure and specify the MCMC method with three imputations.

```
proc mi data=FitMiss seed=55417 nimpu=3 outest=miest noprint;
  multinormal method=mcmc;
  var Oxygen RunTime RunPulse;
run;
```

The following statements list the parameters used for the first two 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;
  where _Imputation_ < 3;
  title 'Parameters for the First 2 Imputations';
run;
```

Output 11.0.21. OUTEST Data Set

Parameters for the First 2 Imputations						
Obs	_Imputation_	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	1	SEED		1320040631.00	1320040631.00	1320040631.00
2	1	N		31.00	31.00	31.00
3	1	PARM		48.33	10.40	169.01
4	1	COV	Oxygen	22.48	-4.77	-21.85
5	1	COV	RunTime	-4.77	1.43	3.95
6	1	COV	RunPulse	-21.85	3.95	165.12
7	2	SEED		1815608871.00	1815608871.00	1815608871.00
8	2	N		31.00	31.00	31.00
9	2	PARM		47.53	10.62	173.27
10	2	COV	Oxygen	38.82	-10.46	-62.66
11	2	COV	RunTime	-10.46	3.44	19.83
12	2	COV	RunPulse	-62.66	19.83	252.27

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

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

Output 11.0.22. Model Information

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 3.

Example 5. Using EM on Bootstrap Resampling for MCMC

This example uses a version of the MCMC method that derives an EM estimate on a bootstrap sample to begin the MCMC process. It also uses a ridge prior to derive posterior distributions for the parameters.

The following statements request the MCMC method with a ridge prior.

```
proc mi data=FitMiss seed=55417 nimpu=2;
  multinormal method=mcmc(initial=em(bootstrap) prior=ridge=.2);
  var Oxygen RunTime RunPulse;
run;
```

Output 11.0.23. Model Information

The MI Procedure	
Model Information	
Data Set	WORK.FITMISS
Method	MCMC
Multiple Imputation Chain	Multiple Chains
Initial Estimates for MCMC	EM
Start	Starting Value
Prior	Ridge=0.2
Number of Imputations	2
Number of Burn-in Iterations	200
Seed for random number generator	55417

The following “Initial Parameter Estimates for MCMC” table displays the starting mean and covariance estimates used in MCMC. With the BOOTSTRAP option, the EM algorithm is applied to each bootstrap resampling separately to find starting estimates for the MCMC process.

Output 11.0.24. Initial Parameter Estimates

Initial Parameter Estimates for MCMC					
IMPUTATION	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	MEAN		47.986217	10.448728	174.060815
1	COV	Oxygen	39.346150	-6.752048	-16.237315
1	COV	RunTime	-6.752048	2.197427	2.517385
1	COV	RunPulse	-16.237315	2.517385	94.334662
Initial Parameter Estimates for MCMC					
IMPUTATION	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
2	MEAN		48.882537	10.319348	163.742226
2	COV	Oxygen	28.221148	-3.976001	-23.428993
2	COV	RunTime	-3.976001	1.121231	4.341911
2	COV	RunPulse	-23.428993	4.341911	109.275100

Example 6: Checking Convergence in A Single Chain

This example uses a version of the MCMC method with a single chain. It also uses time-series plots and autocorrelations to check the convergence of the single chain.

The following statements request the MCMC method with a single chain. The mean and covariance estimates from each iteration are saved in a data set named `outiter`.

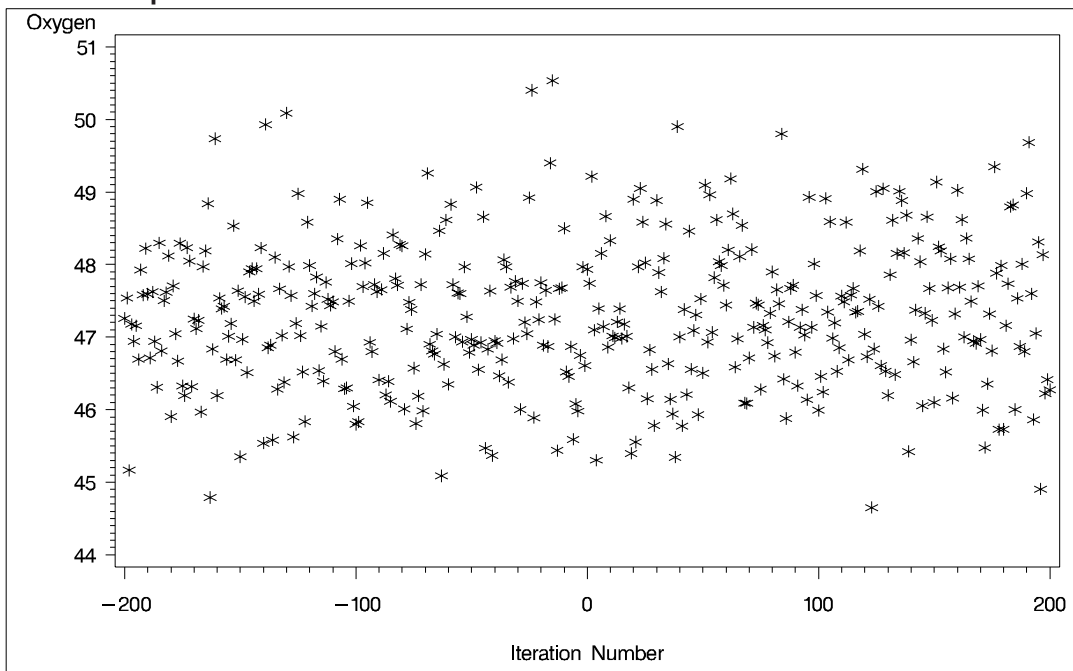
```
proc mi data=FitMiss seed=37921 noprint outiter=outit nimpu=2;
  multinormal method=mcmc(chain=single);
  var Oxygen RunTime RunPulse;
run;
```

The following statements save the mean estimates in a separate data set and plot the estimated means for `Oxygen` against iteration number. Note that iterations in the burn-in period have the iteration number less than zero.

```
data outmean;
  set outit;
  if _type_='MEAN';
run;

proc gplot data=outmean;
  plot Oxygen*_Iteration_;
  symbol v=star;
run;
```

Output 11.0.25. Time-Series Plot



The time-series plot shows no distinct trends for the variable `Oxygen`.

The following statements save the mean estimates in a separate data set named `outmean1` and then check for autocorrelations. With the statement “`if _Iteration_ > 0`”, iterations in the initial burn-in period are excluded.

```
data outmean1;
  set outmean;
  if _Iteration_ > 0;
run;

proc autoreg data=outmean1;
  model Oxygen=/nlag=18;
run;
```

Output 11.0.26. Estimates of Autocorrelations

The AUTOREG Procedure																									
Estimates of Autocorrelations																									
Lag	Covariance	Correlation	-1	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	1		
0	1.0636	1.000000												*****											
1	0.00869	0.008168													*										
2	-0.0717	-0.067444													*										
3	-0.0620	-0.058290													*										
4	-0.0639	-0.060114													*										
5	-0.0379	-0.035587													*										
6	-0.0404	-0.037982													*										
7	0.1167	0.109717													**										
8	-0.0252	-0.023733																							
9	0.0915	0.086075																							
10	-0.00504	-0.004740																							
11	-0.1351	-0.126988													***										
12	-0.0698	-0.065585													*										
13	-0.0908	-0.085350													**										
14	0.0405	0.038086													*										
15	-0.0400	-0.037650													*										
16	0.0577	0.054218													*										
17	0.00464	0.004363																							
18	-0.0910	-0.085542													**										

The “Estimates of Autocorrelations” table shows that autocorrelations are small after lag 11.

Output 11.0.27. Tests for Autocorrelations

The AUTOREG Procedure			
Ordinary Least Squares Estimates			
SSE	212.720292	DFE	199
MSE	1.06895	Root MSE	1.03390
SBC	585.205882	AIC	581.907564
Regress R-Square	0.0000	Total R-Square	0.0000
Durbin-Watson	1.9774	Pr < DW	0.4363
Pr > DW	0.5637		

NOTE: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

The “Ordinary Least Squares Estimates” table shows no significant positive or negative autocorrelation. The example shows convergence for the variable Oxygen. You can also apply the above process to other parameter estimates.

References

- Anderson, T.W. (1984), *An Introduction to Multivariate Statistical Analysis*, 2nd 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.
- 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.
- 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.