Multiple Imputation for Missing Data: Concepts and New Development (Version 9.0)

Yang C. Yuan, SAS Institute Inc., Rockville, MD

Abstract

Multiple imputation provides a useful strategy for dealing with data sets with missing values. Instead of filling in a single value for each missing value, Rubin's (1987) multiple imputation procedure replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute. These 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 imputed data sets is essentially the same. This results in valid statistical inferences that properly reflect the uncertainty due to missing values.

This paper reviews methods for analyzing missing data, including basic concepts and applications of multiple imputation techniques. The paper presents SAS®procedures, PROC MI and PROC MIANALYZE, for creating multiple imputations for incomplete multivariate data and for analyzing results from multiply imputed data sets.

The MI and MIANALYZE procedures, which were introduced as experimental software in Releases 8.1 and 8.2, are production software in Version 9.0. The syntax and examples in this paper apply to Version 9.0. The following enhancements have been made to the MI procedure in Version 9.0:

- A new REGPMM option in the MONOTONE statement and a new PMM option in the MCMC statement request the predicted mean matching method for imputation. This method imputes an observed value which is closest to the predicted value from the simulated regression model for each missing value.
- A flexible model specification in the MONOTONE statement allows a different set of covariates to be specified for each imputed variable.

The following changes and enhancements have been made to the MIANALYZE procedure in Version 9.0:

- A new MODELEFFECTS statement allows you to specify the effects in the data set to be analyzed. This statement replaces the VAR statement, which was used in Releases 8.1 and 8.2.
- A new STDERR statement provides standard errors associated with effects in the MODELEFFECTS statement. The statement can be used for univariate inference when the input DATA= data set contains both parameter estimates and standard errors as variables.
- A new TEST statement tests linear hypotheses about the parameters.

This paper also describes new experimental features in Version 9.0 for specification of classification variables in the MI and MIANALYZE procedures.

Introduction

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 information in the incomplete cases. This approach also ignores the possible systematic difference 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 on 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 may make better use of the available data, but the resulting correlation matrix may not be positive definite.

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

Instead of filling in a single value for each missing value, a multiple imputation procedure (Rubin 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, valid confidence intervals for parameters.

Multiple imputation inference involves three distinct phases:

- The missing data are filled in *m* times to generate *m* complete data sets.
- The *m* complete data sets are analyzed by using standard procedures.
- The results from the *m* complete data sets are combined for the inference.

The MI procedure in the SAS/STAT Software is a multiple imputation procedure that creates multiply imputed data sets for incomplete *p*-dimensional multivariate data. It uses methods that incorporate appropriate variability across the *m* imputations. Once the *m* complete data sets are analyzed by using standard procedures, the MIANALYZE procedure can be used to generate valid statistical inferences about these parameters by combining results from the *m* complete data sets.

Ignorable Missing-Data Mechanism

Let $\mathbf{Y}$ be 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).

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 it is not MAR.

The MAR assumption is not the same as missing completely at random (MCAR), which is a special case of MAR. With MCAR, 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, these SAS procedures also assume that the parameters θ of the data model and the parameters ϕ of 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.

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 that assumes multivariate normality or a nonparametric method that uses propensity scores 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 is said to have a monotone missing pattern when the event that a variable Y_j is missing for the individual *i* implies that all subsequent variables Y_k, *k>j*, are all missing for the individual *i*. When you have 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.

When you have an arbitrary missing data pattern, you can often use the MCMC method, which creates multiple imputations by using 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.

Regression Method

In the regression method, a regression model is fitted for each variable with missing values. Based on the resulting model, a new regression model is then drawn and is used to impute the missing values for the variable (Rubin 1987, pp. 166-167.) Since the data set has a monotone missing data pattern, the process is repeated sequentially for variables with missing values.

That is, for a variable Y_j with missing values, a model

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

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

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

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

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

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

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

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

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 $k+1$ independent random normal variates.

2. The missing values are then replaced by

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

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

Note that the predictive mean matching method can also be used for imputation. It is similar to the regression method except that for each missing value, it imputes an observed value which is closest to the predicted value from the simulated regression model (Rubin 1987, p. 168). The predictive mean matching method ensures that imputed values are plausible and may be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Example: Regression Method

This example uses the regression method to impute missing values for all variables in a data set with a monotone missing pattern. The data set Fish1 used here is a modified version of the Fish data set described in the SAS/STAT documentation for the STEPDISC procedure.

The data set Fish1 data set contains three measurements for a single species of fish: the length from the nose of the fish to the beginning of the tail (Length1), the length from the nose to the notch of the tail (Length2), and the length from the nose to the end of the tail (Length3). Some values have been set to missing, so that the data set has a monotone missing pattern in variables Length1, Length2, and Length3.

```
*---------------Data on Fish Measurements-------------*
| The Fish1 data set contains only one species of     |
| fish and the three length measurements. Some values |
| have been set to missing and the resulting data set |
| has a monotone missing pattern in variables         |
| Length1, Length2, and Length3.                      |
*-----------------------------------------------------*;
```

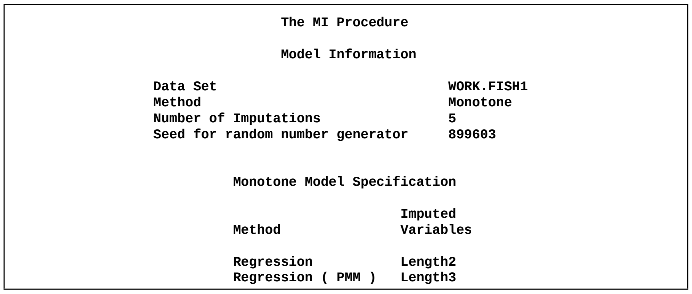
```
data Fish1;
   input Length1 Length2 Length3 @@;
   datalines;
 23.2  25.4  30.0     24.0  26.3  31.2
 23.9  26.5  31.1     26.3  29.0  33.5
 26.5  29.0    .      26.8  29.7  34.7
 26.8    .     .      27.6  30.0  35.0
 27.6  30.0  35.1     28.5  30.7  36.2
 28.4  31.0  36.2     28.7    .     .
 29.1  31.5    .      29.5  32.0  37.3
 29.4  32.0  37.2     29.4  32.0  37.2
 30.4  33.0  38.3     30.4  33.0  38.5
 30.9  33.5  38.6     31.0  33.5  38.7
 31.3  34.0  39.5     31.4  34.0  39.2
 31.5  34.5    .      31.8  35.0  40.6
 31.9  35.0  40.5     31.8  35.0  40.9
 32.0  35.0  40.6     32.7  36.0  41.5
 32.8  36.0  41.6     33.5  37.0  42.6
 35.0  38.5  44.1     35.0  38.5  44.0
 36.2  39.5  45.3     37.4  41.0  45.9
 38.0  41.0  46.5
 ;
```

The following statements invoke the MI procedure and request the regression method for variable Length2 and the predictive mean matching method for variable Length3. The resulting data set is named miout1.

```
proc mi data=Fish1 seed=899603 out=miout1;
    var Length1 Length2 Length3;
    monotone
      reg(Length2)
      regpmm(Length3= Length1 Length2 Length1*Length2);
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. Since no covariates specified for imputed variable Length2, the preceding variable in the VAR statement (Length1) is used as the default covariate for Length2.

The procedure generates the following output:

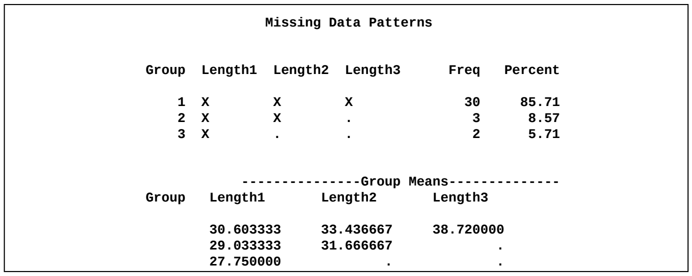


```
                          The MI Procedure

                         Model Information

            Data Set                              WORK.FISH1
            Method                                Monotone
            Number of Imputations                 5
            Seed for random number generator      899603


                   Monotone Model Specification

                                     Imputed
                   Method            Variables

                   Regression        Length2
                   Regression ( PMM )   Length3
```

Figure 1. Model Information

The "Model Information" table describes the method and options used in the multiple imputation process. By default, five imputations are created for the missing data.

When monotone methods are used in the imputation, MONOTONE is displayed as the method. The "Monotone Model Specification" table displays specific monotone methods used in the imputation.



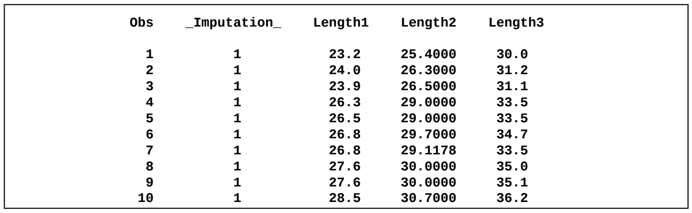
```
                         Missing Data Patterns


   Group  Length1  Length2  Length3      Freq   Percent

       1  X        X        X              30     85.71
       2  X        X        .               3      8.57
       3  X        .        .               2      5.71


                  ---------------Group Means--------------
   Group   Length1       Length2       Length3

           30.603333     33.436667     38.720000
           29.033333     31.666667             .
           27.750000             .             .
```

Figure 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 variable means in each group are also displayed. The table also displays group-specific variable means.

The following statements produce a listing of the first ten observations of data set miout1 with imputed values.

```
proc print data=miout1 (obs=10);
run;
```



```
                Obs    _Imputation_    Length1    Length2    Length3

                  1          1           23.2     25.4000      30.0
                  2          1           24.0     26.3000      31.2
                  3          1           23.9     26.5000      31.1
                  4          1           26.3     29.0000      33.5
                  5          1           26.5     29.0000      33.5
                  6          1           26.8     29.7000      34.7
                  7          1           26.8     29.1178      33.5
                  8          1           27.6     30.0000      35.0
                  9          1           27.6     30.0000      35.1
                 10          1           28.5     30.7000      36.2
```

Figure 3. Output Data Set

Propensity Score Method

The propensity score is the conditional probability of assignment to a particular treatment given a vector of observed covariates (Rosenbaum and Rubin 1983). In the propensity score method, a propensity score is generated for each variable with missing values to indicate the probability of that observation being 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

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

where $X_1, X_2, ..., X_k$ are s set of covariates for Y_j, $p_j = Pr(R_j = 0 | X_1, X_2, ..., X_k)$, 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 (Schafer 1999, p. 11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

MCMC Method

MCMC 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 on the value of the previous one.

In MCMC, one constructs a Markov chain long enough for the distribution of the elements to stabilize to a common distribution. This stationary distribution 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. 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.

Assuming that the data are from a multivariate normal distribution, data augmentation is applied to Bayesian inference with missing data by repeating the following steps:

1. The imputation I-step:
With the 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 $Y_{i(mis)}$ given $Y_{i(obs)}$.

2. The posterior P-step:
The P-step simulates the posterior population mean vector and covariance matrix from the complete sample estimates. These new estimates are then used in the 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). The goal is to have the iterates converge to their stationary distribution and then to simulate an approximately independent draw of the missing values.

That is, with a current parameter estimate $\theta^{(t)}$ at 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

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

which converges in distribution to $p(Y_{mis}, \theta|Y_{obs})$.

Example: MCMC Method

The following Fitness data set has been altered to contain an arbitrary missing pattern:

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

The following statements use the MCMC method to impute missing values for all variables in a data set. The resulting data set is named miout2. These statements also create an iteration plot for the successive estimates of the mean of Oxygen and an autocorrelation function plot for the variable Oxygen.

```
   proc mi data=FitMiss seed=42037921
           nimpute=3 out=miout2;
      mcmc timeplot(mean(Oxygen)) acfplot(mean(Oxygen));
      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                3
          Number of Burn-in Iterations         200
          Number of Iterations                 100
          Seed for random number generator     42037921
```

Figure 4. Model Information

By default, the procedure uses a single chain to create five imputations. It takes 200 burn-in iterations before the first imputation and 100 iterations between imputations. The burn-in iterations are used to make the iterations converge to the stationary distribution before the imputation.

The expectation-maximization (EM) algorithm (Little and Rubin 1987) is a technique that finds maximum likelihood estimates for parametric models for incomplete data. It can also be used to compute posterior modes, the parameter estimates with the highest observed-data posterior density.

By default, the procedure also uses the statistics from the available cases in the data as the initial estimates for the EM algorithm, the correlations are set to zero. The resulting EM estimate provides a good starting value with which to begin the MCMC process. Refer to Schafer (1997, p. 169) for suggested starting values for the algorithm.

```
                          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


                           ---------------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 5. Missing Data Patterns

The “Missing Data Patterns” table lists distinct missing data patterns. It shows that the data set does not have a monotone missing pattern.

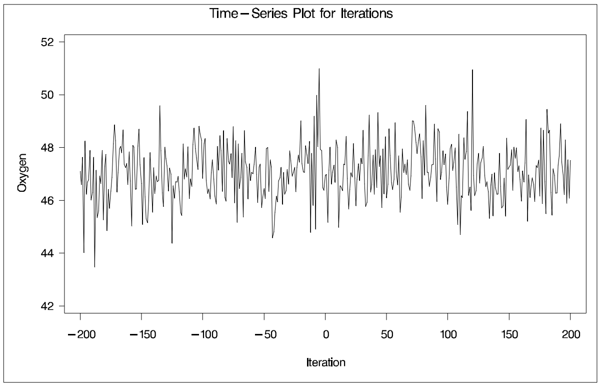


Figure 6. Time-Series Plot for Oxygen

With the TIMEPLOT(MEAN(Oxygen)) option, the procedure displays a time-series plot for the mean of Oxygen. The iterations during the burn-in period are indicated with negative iteration numbers in the plot. The plot shows no apparent trends for the variable Oxygen.

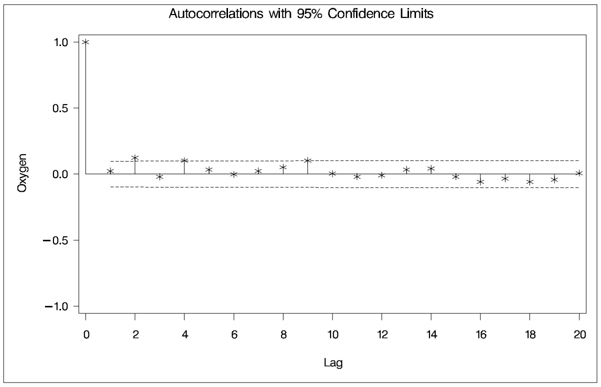


Figure 7. Autocorrelation Function Plot for Oxygen

With the ACFPLOT(MEAN(Oxygen)) option, an autocorrelation plot for the mean of Oxygen is displayed. It shows no significant positive or negative autocorrelation.

Combining Inferences from Imputed Data Sets

With m imputations, you can compute m different sets of the point and variance estimates for a parameter Q. Let $\hat{Q}_i$ and $\hat{U}_i$ be the point and variance estimates from the ith imputed data set, i=1, 2, ..., m. Then the point estimate for Q from multiple imputations is the average of the m complete-data estimates:

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

Let $\overline{U}$ be 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 be 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

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

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

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

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^*.

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

where

$$v_{\hat{obs}} = \frac{v_0 + 1}{v_0 + 3} \, v_0 \, (1 - \gamma)$$

$$\gamma = \frac{(1+m^{-1})B}{T}$$

Similar to the univariate inferences, multivariate inferences based on Wald's tests can also be derived from the m imputed data sets.

Multiple Imputation Efficiency

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, both values r and B are zero. With a large value of m or a small value of r, the degrees of freedom v_m will be large and the distribution will be approximately normal.

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

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

The relative efficiency 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.

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

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

	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

Imputation Model

Multiple imputation inference assumes that the model (variables) used to analyze the multiply imputed data, referred to as the analyst's model, is the same as the model used to impute missing values, referred to as the imputer's model. In practice, the two models need not be the same, as explained by Schafer (1997, pp. 139–143), who discusses the consequences for various scenarios.

In general, you should include as many variables as you can in the imputer's model. To produce high-quality imputations for a particular variable, you should include variables that are potentially related to the imputed variable and variables that are potentially related to the missingness of the imputed variable (Schafer 1997, p. 143). The precision you lose by including unimportant predictors is usually a relatively small price to pay for strengthening the validity of subsequent analysis of the multiply imputed data set (Rubin 1996). However, at the same time, the process of model building and fitting must be kept feasible (Barnard and Meng, 1999, pp. 19–20).

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

It is good practice to supply a description of the imputer's model with the multiply imputed data set (Rubin 1996, p.479). The description should include information about the variables involved in the imputation and relationships among the variables which have been implicitly set to zero.

The MI Procedure

The MI procedure provides three methods to create imputed data sets that can be analyzed using standard procedures.

The following statements are available in PROC MI:

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

BY *variables* **;**
EM < *options* > **;**
FREQ *variable* **;**
MCMC < *options* > **;**
MONOTONE < *options* > **;**
TRANSFORM *transform (variables) ;*
VAR *variables* **;**

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

The MONOTONE statement specifies monotone methods to impute variables for a data set with a monotone missing pattern. Available options for the MONOTONE statement include REGRESSION, REGPMM, and PROPENSITY.

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 in the MI procedure. Available options in the PROC MI statement include:

NIMPUTE=*number*
specifies the number of imputations. The default is NIMPUTE=5.

OUT=*SAS-data-set*
creates an output SAS data set in which to put the imputation results. The data set includes an identification variable, _IMPUTATION_, to identify the imputation number.

SEED=*number*
specifies a positive integer that is used 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.

If the default value is used, the seed information is displayed 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.

Available options for the MCMC statement include:

CHAIN=SINGLE | MULTIPLE
specifies whether a single chain is used for all imputations or a separate chain is used for each imputation (Schafer 1997, pp. 137-138). The default is CHAIN=SINGLE.

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.

PMM $<$**(** *variables***)**$>$
specifies the variables to be imputed with the predicted mean matching method. For each missing value in the variable, the predicted mean matching method imputes an observed value that is closest to the predicted value from the imputed model.

INITIAL=EM
INITIAL=INPUT=*SAS-data-set*
specifies the initial mean and covariance estimates 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. The resulting estimates are used to begin the MCMC process.

You can also specify INITIAL=INPUT=*SAS-data-set* to use a SAS data set from which to obtain the initial estimates of the mean and covariance matrix for each imputation. The default is INITIAL=EM.

The following two options specify the number of iterations before the first imputation and between imputations:

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.

The following two options provide convergence checking for the MCMC process:

ACFPLOT $<$ *(options)* $>$
displays autocorrelation plots of parameters from iterations.

TIMEPLOT $<$ *(options)* $>$
displays time-series plots of parameters from iterations.

Available options for the MONOTONE statement are:

REGRESSION $<$ **(** *imputed* $<$ **=** *effects* $>$ **)** $>$
REGPMM $<$ **(** *imputed* $<$ **=** *effects* $>$ **)** $>$
PROPENSITY $<$ **(** *imputed* $<$ **=** *effects* $>$ **)** $>$
specifies the regression method, the new predictive mean matching method, and the propensity scores method of variables. You can use the new effects specification option to use a different set of covariates for each imputed variable.

Although the MI procedure with a regression or MCMC method assumes multivariate normality, the inference by multiple imputation may be robust to departures from the multivariate normality if the amounts of missing information are 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 the simulation studies described in Schafer (1997) and the original references therein.

The MIANALYZE Procedure

From m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. PROC MIANALYZE combines these results and generates valid statistical inferences about the parameter. Multivariate inferences can also be derived from the m imputed data sets.

The following statements are available in PROC MIANALYZE:

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

> **BY** *variables* **;**
> **MODELEFFECTS** *effects* **;**
> **STDERR** *variables* **;**
> **TEST** *equation1* $<$, . . . , $<$ *equationk* $>$ $>$ **;**

The new MODELEFFECTS statement lists the effects in the data set to be analyzed. Each effect is a variable or a combination of variables, and is specified with a special notation using variable names and operators. This statement replaces the VAR statement, which was used in Releases 8.1 and 8.2 where only variables can be specified.

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

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

The PROC MIANALYZE and MODELEFFECTS statements are required. Available options in the PROC MIANALYZE statement are:

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

EDF=*numbers*
specifies the complete-data degrees of freedom for the parameter estimates. This is used to compute an adjusted degrees of freedom.

MU0=*numbers*
specifies the means under the null hypothesis in the *t*-test for location. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a MODELEFFECTS statement, and the specified numbers correspond to variables in the MODELEFFECTS statement.

MULT | MULTIVARIATE
requests multivariate inference for the variables together.

DATA=*SAS-data-set*
names a specially structured SAS data set to be analyzed by PROC MIANALYZE. The input data set must have a TYPE of COV, CORR, or EST. The parameter estimates and their associated covariance matrix from each imputed data set are read from the data set.

PARMINFO=*SAS-data-set*
names an input SAS data set that contains parameter information associated with variables PRM1, PRM2,..., and so on. These variables are used as variables for parameters in a COVB= data set.

PARMS=*SAS-data-set*
names a SAS data set that contains parameter estimates from imputed data sets.

COVB=*SAS-data-set*
names a SAS data set that contains covariance matrices of the parameter estimates from imputed data sets. If you use the COVB= option, the PARMS= option must also be specified.

XPXI=*SAS-data-set*
names a SAS data set that contains X'X inverse matrices related to the parameter estimates from imputed data sets. If you use the XPXI= option, the PARMS= option must also be specified. In this case, PROC MIANALYZE also reads the standard errors of the estimates from the PARMS= data. The standard errors and X'X inverse matrices are used to derive the covariance matrices.

Input Data Sets

The appropriate input specification depends on the type of inference and the SAS procedure you used to create the data sets. You specify input data sets based on the type of inference you requested. For univariate inference, you can use one of the following options:

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

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

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

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

Examples

The following statements generate five imputed data sets to be used in this section.

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

Example 1. Reading Means and Standard Errors from Variables in a DATA= Data Set

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

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

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

The following statements display the output data set from PROC UNIVARIATE:

```
proc print data=outuni;
   title ’UNIVARIATE Means and Standard Errors’;
run;
```

```
                      UNIVARIATE Means and Standard Errors

                                               Run                 SRun      SRun
Obs   _Imputation_    Oxygen   RunTime    Pulse   SOxygen     Time     Pulse

 1          1        47.0120   10.4441   171.216   0.95984   0.28520   1.59910
 2          2        47.0451   10.5401   170.005   0.93008   0.26338   1.77583
 3          3        46.9630   10.4624   172.652   0.98626   0.28930   1.74034
 4          4        46.8488   10.6020   172.730   0.95865   0.25038   1.74767
 5          5        46.9786   10.4918   171.826   0.99090   0.26286   1.92357
```

Figure 8. UNIVARIATE Output Data Set

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

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

```
                          The MIANALYZE Procedure

                             Model Information

                  Data Set                    WORK.OUTUNI
                  Number of Imputations       5


                 Multiple Imputation Variance Information

                  -----------------Variance-----------------
       Parameter          Between          Within           Total        DF

       OXYGEN            0.005557        0.931989        0.938657    27.972
       RUNTIME           0.004072        0.073238        0.078124    25.754
       RUNPULSE          1.272250        3.098789        4.625490    12.469

                 Multiple Imputation Variance Information

                                 Relative        Fraction
                                 Increase         Missing
                 Parameter     in Variance     Information

                 OXYGEN           0.007155        0.007129
                 RUNTIME          0.066724        0.064379
                 RUNPULSE         0.492676        0.363798
```

Figure 9. Multiple Imputation Variance Information

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

The “Multiple Imputation Variance Information” table displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missing values and the fraction of missing information for each variable are also displayed.

```
                          The MIANALYZE Procedure

                 Multiple Imputation Parameter Estimates

Parameter         Estimate       Std Error     95% Confidence Limits         DF

OXYGEN           46.969511        0.968843      44.9848       48.9542    27.972
RUNTIME          10.508089        0.279507       9.9333       11.0829    25.754
RUNPULSE        171.685634        2.150695     167.0191      176.3521    12.469

                 Multiple Imputation Parameter Estimates

                 Parameter          Minimum         Maximum

                 OXYGEN           46.848766       47.045131
                 RUNTIME          10.444149       10.602007
                 RUNPULSE        170.004535      172.729655

                 Multiple Imputation Parameter Estimates

                                                t for H0:
          Parameter           Theta0    Parameter=Theta0    Pr > |t|

          OXYGEN                   0               48.48      <.0001
          RUNTIME                  0               37.60      <.0001
          RUNPULSE                 0               79.83      <.0001
```

Figure 10. Multiple Imputation Parameter Estimates

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

Example 2. Reading Regression Results from a DATA= EST Data Set

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

The following statements use the REG procedure to generate regression coefficients:

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

The following statements display regression coefficients and their covariance matrices.

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

```
                 REG Model Coefficients and Covariance matrices

Obs    _Imputation_    _TYPE_     _NAME_       Intercept      RunTime     RunPulse

  1          1         PARMS                      86.544     -2.82231     -0.05873
  2          1         COV       Intercept       100.145     -0.53519     -0.55077
  3          1         COV       RunTime          -0.535      0.10774     -0.00345
  4          1         COV       RunPulse         -0.551     -0.00345      0.00343
  5          2         PARMS                      92.451     -2.89662     -0.08750
  6          2         COV       Intercept        64.527     -0.37466     -0.35512
  7          2         COV       RunTime          -0.375      0.10754     -0.00446
  8          2         COV       RunPulse         -0.355     -0.00446      0.00237
```

Figure 11. EST Type Data Set

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

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

```
                          The MIANALYZE Procedure

                  Multiple Imputation Variance Information

                   -----------------Variance-----------------
      Parameter         Between         Within          Total       DF

      INTERCEP         7.405948      80.807859      89.694996   22.306
      RUNTIME          0.033768       0.114730       0.155252   14.558
      RUNPULSE         0.000182       0.002720       0.002938   23.468

                  Multiple Imputation Variance Information

                                   Relative       Fraction
                                   Increase        Missing
                  Parameter     in Variance    Information

                  INTERCEP         0.109979       0.103472
                  RUNTIME          0.353194       0.284956
                  RUNPULSE         0.080115       0.076709
```

Figure 12. Variance Information

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

```
                  Multiple Imputation Parameter Estimates

Parameter        Estimate      Std Error    95% Confidence Limits        DF

INTERCEP        91.396566       9.470744     71.77107     111.0221   22.306
RUNTIME         -2.980954       0.394020     -3.82302      -2.1389   14.558
RUNPULSE        -0.076286       0.054202     -0.18829       0.0357   23.468

                  Multiple Imputation Parameter Estimates

                  Parameter         Minimum        Maximum

                  INTERCEP        86.544034      92.930137
                  RUNTIME         -3.199326      -2.822311
                  RUNPULSE        -0.091658      -0.058729

                  Multiple Imputation Parameter Estimates

                                             t for H0:
          Parameter          Theta0   Parameter=Theta0   Pr > |t|

          INTERCEP                0               9.65     <.0001
          RUNTIME                 0              -7.57     <.0001
          RUNPULSE                0              -1.41     0.1724
```

Figure 13. Parameter Estimates

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*-distribution.

Example 3. Reading Mixed Model Results from PARMS= and COVB= Data Sets

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

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

```
proc mixed data=outmi;
   model Oxygen= RunTime RunPulse
                 RunTime*RunPulse/solution covb;
   by _Imputation_;
   ods output SolutionF=mixparms CovB=mixcovb;
run;
```

The following statements display parameter estimates from the first two imputed data sets.

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

```
                          MIXED Model Coefficients

      Obs    _Imputation_    Effect              Estimate      StdErr

        1          1         Intercept             148.09     81.5231
        2          1         RunTime              -8.8115      7.8794
        3          1         RunPulse             -0.4123      0.4684
        4          1         RunTime*RunPulse     0.03437     0.04517
        5          2         Intercept             164.64     55.4039
        6          2         RunTime             -10.0189      5.4193
        7          2         RunPulse             -0.5068      0.3221
        8          2         RunTime*RunPulse     0.04130     0.03137
```

Figure 14. MIXED Model Coefficients

The following statements display the covariance matrices associated with the parameter estimates from the first two imputed data sets. Note that the variables Col1, Col2, Col3, and Col4 are used to identify the effects Intercept, RunTime, RunPulse, and RunTime*RunPulse through the variable Row.

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

```
                           Covariance Matrices

Obs _Imputation_  Row Effect                Col1     Col2     Col3     Col4

  1       1         1 Intercept          6646.01  -637.40 -38.1515   3.6542
  2       1         2 RunTime            -637.40  62.0842   3.6548  -0.3556
  3       1         3 RunPulse          -38.1515   3.6548   0.2194 -0.02099
  4       1         4 RunTime*RunPulse    3.6542  -0.3556 -0.02099 0.002040
  5       2         1 Intercept          3069.59  -296.99 -17.8108   1.7201
  6       2         2 RunTime            -296.99  29.3690   1.7186  -0.1697
  7       2         3 RunPulse          -17.8108   1.7186   0.1038 -0.00999
  8       2         4 RunTime*RunPulse    1.7201  -0.1697 -0.00999 0.000984
```

Figure 15. MIXED Covariance Matrices

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

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

```
                          The MIANALYZE Procedure

                 Multiple Imputation Variance Information

                      -----------------Variance-----------------
    Parameter              Between          Within           Total       DF

    INTERCEPT          2188.564843     4757.824063     7384.101874   11.004
    RUNTIME              22.088771       45.308732       71.815257   10.574
    RUNPULSE              0.067241        0.157601        0.238290   11.575
    RUNTIME*RUNPULSE      0.000682        0.001493        0.002312    11.06

                 Multiple Imputation Variance Information

                                     Relative        Fraction
                                     Increase         Missing
                Parameter         in Variance     Information

                INTERCEPT            0.551991        0.392889
                RUNTIME              0.585020        0.408084
                RUNPULSE             0.511988        0.373534
                RUNTIME*RUNPULSE     0.547921        0.390968
```

Figure 16. Multiple Imputation Variance Information

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

```
                          The MIANALYZE Procedure

                  Multiple Imputation Parameter Estimates

Parameter             Estimate      Std Error    95% Confidence Limits        DF

INTERCEPT           158.762699      85.930797     -30.3612     347.8866   11.004
RUNTIME              -9.596869       8.474388     -28.3409       9.1472   10.574
RUNPULSE             -0.463078       0.488150      -1.5310       0.6049   11.575
RUNTIME*RUNPULSE      0.037952       0.048080      -0.0678       0.1437    11.06

                  Multiple Imputation Parameter Estimates

                 Parameter                Minimum         Maximum

                 INTERCEPT              85.320006      206.585760
                 RUNTIME               -14.253798       -2.127674
                 RUNPULSE               -0.728034       -0.050394
                 RUNTIME*RUNPULSE       -0.004085        0.063710

                  Multiple Imputation Parameter Estimates

                                                  t for H0:
          Parameter                Theta0    Parameter=Theta0    Pr > |t|

          INTERCEPT                     0                1.85      0.0917
          RUNTIME                       0               -1.13      0.2825
          RUNPULSE                      0               -0.95      0.3622
          RUNTIME*RUNPULSE              0                0.79      0.4465
```

Figure 17. Multiple Imputation Parameter Estimates

The “Multiple Imputation Parameter Estimates” table displays the estimated mean and standard error of the regression coefficients.

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

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

Example 4. Reading GENMOD Results from PARMS=, PARMINFO=, and COVB= Data Sets

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

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

```
   proc genmod data=outmi;
      model Oxygen= RunTime RunPulse/covb;
      by _Imputation_;
      ods output ParameterEstimates=gmparms
                 ParmInfo=gmpinfo
                 CovB=gmcovb;
   run;
```

The following statements print parameter estimates and associated standard errors from the first two imputed data sets.

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

```
                      GENMOD Model Coefficients

         Obs    _Imputation_    Parameter    Estimate      StdErr

           1          1         Intercept     86.5440      9.5107
           2          1         RunTime       -2.8223      0.3120
           3          1         RunPulse      -0.0587      0.0556
           4          1         Scale          2.6692      0.3390
           5          2         Intercept     92.4506      7.6343
           6          2         RunTime       -2.8966      0.3117
           7          2         RunPulse      -0.0875      0.0462
           8          2         Scale          2.4033      0.3052
```

Figure 18. GENMOD Model Coefficients

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

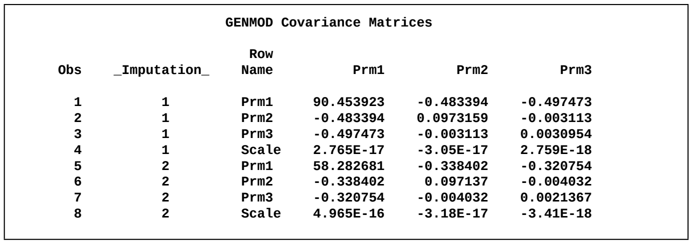
```
   proc print data=gmpinfo (obs=6);
     title ’GENMOD Parameter Information’;
   run;
```

```
                     GENMOD Parameter Information

              Obs    _Imputation_    Parameter    Effect

               1          1            Prm1       Intercept
               2          1            Prm2       RunTime
               3          1            Prm3       RunPulse
               4          2            Prm1       Intercept
               5          2            Prm2       RunTime
               6          2            Prm3       RunPulse
```

Figure 19. GENMOD Parameter Information

The following statements display the covariance matrices from the first two imputed data sets. Note that the GENMOD procedure computes maximum likelihood estimates for each covariance matrix.

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



```
                         GENMOD Covariance Matrices

                                  Row
   Obs    _Imputation_    Name         Prm1         Prm2         Prm3

     1          1         Prm1     90.453923    -0.483394    -0.497473
     2          1         Prm2     -0.483394    0.0973159    -0.003113
     3          1         Prm3     -0.497473    -0.003113    0.0030954
     4          1         Scale    2.765E-17    -3.05E-17    2.759E-18
     5          2         Prm1     58.282681    -0.338402    -0.320754
     6          2         Prm2     -0.338402     0.097137    -0.004032
     7          2         Prm3     -0.320754    -0.004032    0.0021367
     8          2         Scale    4.965E-16    -3.18E-17    -3.41E-18
```

Figure 20. GENMOD Covariance Matrices

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

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

Since the GENMOD procedure computes maximum likelihood estimates for the covariance matrix, the EDF= option is not used. The resulting model coefficients are identical to the estimates from the previous regression example but the standard errors are slightly different because in this example, maximum likelihood estimates for the standard errors are combined without the EDF= option, whereas in the previous regression example, unbiased estimates for the standard errors are combined with the EDF= option.

V9 Experimental Features

In Version 9.0, a CLASS statement has been added in the MI procedure. This is an experimental feature. You can use the CLASS statement to specify classification variables for data sets with monotone missing patterns, which can be used either as covariates for imputed variables or as imputed variables. You must use the CLASS statement in conjunction with the VAR and the MONOTONE statements. To impute missing values of a CLASS variable, the MI procedure provides a logistic regression method and a discriminant function method.

Likewise, an experimental CLASS statement has been added to the MIANALYZE procedure in Version 9.0. You can use this statement to specify classification variables in the MODELEFFECTS statement. Consequently, you can use the MIANALYZE procedure to combine results from other SAS procedures which analyze classification variables.

Acknowledgments

I am grateful to Maura Stokes, Bob Rodriguez, and Virginia Clark of the Applications Division at SAS Institute for their valuable assistance in the preparation of this manuscript.

The development of the MI and MIANALYZE procedures has benefited significantly from the advice and expertise of Paul D. Allison, University of Pennsylvania; John Barnard Jr., deCODE genetics; Nicholas J. Horton, Boston University; Donald B. Rubin, Harvard University; and Joseph L. Schafer, Penn State University. We gratefully acknowledge the time and effort that these researchers have contributed.

References

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

Barnard, J., and Meng, X.L. (1999), "Applications of Multiple Imputation in Medical Studies: From AIDS to NHANES," *Statistical Methods in Medical Research*, 8, 17–36.

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

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

Lavori, P.W., Dawson, R., and Shera, D. (1995), "A Multiple Imputation Strategy for Clinical Trials with Truncation of Patient Data," *Statistics in Medicine*, 14, 1913–1925.

Little, R.J.A. and Rubin, D.B. (1987), *Statistical Analysis with Missing Data*, New York: John Wiley & Sons, Inc.

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

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

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

Contact Information

Your comments and questions are valued and encouraged. Contact the author at

Yang C. Yuan, SAS Institute Inc., 1700 Rockville Pike, Suite 600, Rockville, MD 20852. Phone (301) 881-8840 ext 3355. FAX (301) 881-8477. E-mail Yang.Yuan@sas.com

Version 3.0