P267-25

Multiple Imputation for Missing Data: Concepts and New Development

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 also presents new SAS® procedures for creating multiple imputations for incomplete multivariate data and for analyzing results from multiply imputed data sets.

These procedures are still under development and will be available in experimental form in Release 8.1 of the SAS System.

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 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 from the variable mean of the complete cases, or it can be imputed from the mean conditional on observed values of other variables. 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.

A new SAS/STAT® procedure, PROC MI, 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, another new procedure, PROC MIANALYZE, 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 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. This approach is still being researched.

Regression Method

In the regression method, a regression model is fitted for each variable with missing values, with the previous variables as covariates. Based on the resulting model, a new regression model is then fitted and is used to impute the missing values for each 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

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

is fitted with the nonmissing observations.

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

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

The missing values are then replaced by

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

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

The following Fitness data set has a monotone missing data pattern and is used with the regression method to generate imputed data sets. Note that the original data set has been altered for this example.

```
*-------------------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 (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.           |
|                                                         |
| Certain values were changed to missing for the analysis |
*---------------------------------------------------------*;
  data Fitness1;
     input Oxygen RunTime RunPulse @@;
     datalines;
   44.609  11.37  178     45.313  10.07  185
   54.297   8.65  156     59.571    .      .
   49.874   9.22    .     44.811  11.63  176
   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 following statements invoke the MI procedure and request the regression method. The resulting data set is named miout1.

```
   proc mi data=Fitness1 seed=37851 out=miout1;
      multinormal method=regression;
      var Oxygen RunTime RunPulse;
   run;
```

The procedure generates the following output:

```
                              The MI Procedure

                             Model Information

            Data Set                             WORK.FITNESS1
            Method                               Regression
            Number of Imputations                5
            Seed for random number generator     37851


                            Missing Data Patterns

                                 Run     Run
             Group    Oxygen     Time    Pulse        Freq      Percent

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

                           Missing Data Patterns

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

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

Figure 1. Output from PROC MI

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.

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;
```

```
                                                    Run
            Obs    _Imputation_    Oxygen    RunTime     Pulse

              1          1         44.609    11.3700    178.000
              2          1         45.313    10.0700    185.000
              3          1         54.297     8.6500    156.000
              4          1         59.571     9.8709    170.676
              5          1         49.874     9.2200    137.623
              6          1         44.811    11.6300    176.000
              7          1         45.681    11.9500    176.000
              8          1         49.091    10.8500    121.146
              9          1         39.442    13.0800    174.000
             10          1         60.055     8.6300    170.000
```

Figure 2. 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 of

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

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

3. Create a propensity score for each observation to indicate the probability of its being missing.

4. Divide the observations into a fixed number of groups based on these propensity scores.

5. Apply an approximate Bayesian bootstrap imputation to each group. In group k, let Y_{obs} denote the n_1 observations with nonmissing Y_j values and Y_{mis} denote the n_0 observations with missing Y_j. 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 the drawing of parameters from the posterior predictive distribution of the missing data. The process then draws the n_0 values for Y_{mis} randomly with replacement from Y^*_{obs}.

The process is repeated sequentially for each variable with missing values.

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.

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 $p(Y_{mis}, \theta|Y_{obs})$.

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.       |
| 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.           |
|                                                         |
| Certain values were changed to missing for the analysis |
*---------------------------------------------------------*;
   data Fitness2;
      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    .
      .     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
    ;
```

The following statements invoke the MI procedure and specify the MCMC method. The option NIMPU=3 requests three imputations.

```
proc mi data=Fitness nimpu=3 out=mioutmc;
   multinormal method=mcmc;
   var Oxygen RunTime RunPulse;
run;
```

```
                           The MI Procedure

                          Model Information

         Data Set                              WORK.FITNESS2
         Method                                MCMC
         Multiple Imputation Chain             Multiple Chains
         Initial Estimates for MCMC            EM
         Start                                 Starting Value
         Prior                                 Jeffreys
         Number of Imputations                 3
         Number of Burn-in Iterations          50
         Seed for random number generator      36611


                        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               2        6.45
              5    .       X        .               1        3.23

                       Missing Data Patterns

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

               1        46.353810       10.809524      171.666667
               2        47.688750       10.287500               .
               3        52.461667               .               .
               4                .       10.435000      161.000000
               5                .       10.250000               .
```

Figure 3. Output from PROC MI, MCMC

By default, the procedure uses multiple chains to create five imputations. It takes 50 burn-in iterations before the first imputation. The burn-in iterations are used to make the iterations converge to the stationary distribution before the imputation.

By default, the procedure also 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 generated from these standard deviations and zero correlations. Refer to Schafer (1997, p. 169) for suggested starting values for the algorithm.

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. The resulting EM estimate provides a good starting value with which to begin the MCMC process.

The following "Initial Parameter Estimates for MCMC" tables display the starting mean and covariance estimates used in each imputation. The same starting estimates are used to begin the MCMC process for each imputation because the same EM algorithm is used. You can specify different initial estimates for different imputations explicitly or use the bootstrap to generate different initial estimates for the EM algorithm to find possible different starting values for the MCMC process.

```
                             The MI Procedure

                   Initial Parameter Estimates for MCMC

_IMPUTATION_   _TYPE_   _NAME_          Oxygen        RunTime       RunPulse

          1    MEAN                  47.314978      10.562925     170.089851
          1    COV      Oxygen       25.284270      -5.744658     -22.268666
          1    COV      RunTime      -5.744658       1.757043       4.586548
          1    COV      RunPulse    -22.268666       4.586548     103.588249

                   Initial Parameter Estimates for MCMC

_IMPUTATION_   _TYPE_   _NAME_          Oxygen        RunTime       RunPulse

          2    MEAN                  47.314978      10.562925     170.089851
          2    COV      Oxygen       25.284270      -5.744658     -22.268666
          2    COV      RunTime      -5.744658       1.757043       4.586548
          2    COV      RunPulse    -22.268666       4.586548     103.588249

                   Initial Parameter Estimates for MCMC

_IMPUTATION_   _TYPE_   _NAME_          Oxygen        RunTime       RunPulse

          3    MEAN                  47.314978      10.562925     170.089851
          3    COV      Oxygen       25.284270      -5.744658     -22.268666
          3    COV      RunTime      -5.744658       1.757043       4.586548
          3    COV      RunPulse    -22.268666       4.586548     103.588249
```

Figure 4. Initial Parameter Estimates

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

Imputer's Model versus Analyst's Model

Note: This section is mainly based on Section 4.5.4 of Schafer (1997). You should consult this book for a comprehensive discussion on choosing an imputation model.

Multiple imputation inferences assume that the analyst's model is the same as the imputer's model. But in practice, the two models may not be the same.

For example, consider the same trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 with missing values. An imputer creates multiple imputations with the model

$$Y_3 = Y_1 \; Y_2$$

However, the analyst may later model $Y_3 = Y_1$ only. 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 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 it may be somewhat conservative because it reflects the additional uncertainty of estimating the relationship between Y_3 and Y_2.

On the other hand, suppose that the analyst model $Y_3 = Y_1$ only and there is a relationship between variables Y_3 and Y_2. Then the model of $Y_3 = Y_1$ will be biased and the analyst's model 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, an imputer creates multiple imputations with the model

$$Y_3 = Y_1$$

but the analyst later fits a model $Y_3 = Y_1 \; Y_2$. When the assumption is true, the imputer's model is a correct model and the inferences still hold.

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

Thus, generally you want to 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.

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* **;**
> **FREQ** *variable* **;**
> **MULTINORMAL** < *options* > **;**
> **VAR** *variables* **;**

The PROC MI statement is the only required statement in the MI procedure. Available options in the PROC MI statement include:

NIMPU=*number*
specifies the number of imputations. The default is NIMPU=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. For each imputation, the data set contains all variables in the input data set, with missing values being replaced by the imputed values.

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.

The following options are used to make the imputed values consistent with the observed variable values:

MAXIMUM=*numbers*
MINIMUM=*numbers*
specifies the maximum/minimum values for imputed variables. 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 maximum/minimum for the corresponding variable.

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.

The following statements generate imputed values rounded to the desired units:

```
proc mi data=Fitness1 seed=37851 out=miout2
        round=.001 .01 1;
   multinormal method=regression;
   var Oxygen RunTime RunPulse;
run;

proc print data=miout2 (obs=10);
run;
```

```
                                         Run      Run
 Obs    _Imputation_    Oxygen     Time    Pulse

   1          1         44.609    11.37     178
   2          1         45.313    10.07     185
   3          1         54.297     8.65     156
   4          1         59.571     9.87     171
   5          1         49.874     9.22     138
   6          1         44.811    11.63     176
   7          1         45.681    11.95     176
   8          1         49.091    10.85     121
   9          1         39.442    13.08     174
  10          1         60.055     8.63     170
```

Figure 5. Output Data Set with a ROUND option

You 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 one variable in your input data set represents the frequency of occurrence for other values in the observation, you can 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.

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 MULTINORMAL statement specifies the imputation method. Available options are:

METHOD=REGRESSION
METHOD=PROPENSITY $<$**(NGROUPS=*number*)**$>$
METHOD=MCMC $<$**(** *options* **)**$>$

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

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

Available options for METHOD=MCMC 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=MULTIPLE.

INITIAL=EM $<$ **(BOOTSTRAP** $<$ **=** *p* $>$**)** $>$
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 specify a BOOTSTRAP option to use a simple random sample with replacement of $[np]$ observations from the input data set to compute the initial estimates for each chain (Schafer 1997, p. 128), where *n* is the number of observations in the data set, $[np]$ is the integer part of np, and $0 < p <= 1$. This gives an overdispersed initial estimate that provides possible different starting values for the MCMC. If a BOOTSTRAP option is specified without the *p* value, *p*=0.75 is used.

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.

NBITER=*number*
specifies the number of burn-in iterations before the first imputation in each chain. The default is NBITER=50.

NITER=*number*
specifies the number of iterations between imputations in a single chain. The default is NITER=30.

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* **;**
VAR *variables* **;**

The PROC MIANALYZE and VAR 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 VAR statement, and the specified numbers correspond to variables in the VAR 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.

PARMS=*SAS-data-set*
names a SAS data set that contains parameter estimates from imputed data sets. If you use the PARMS= option, then either the COVB= or XPXI= option must be specified.

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

If you do not specify an input data set with the DATA=, COVB=, or XPXI= option, then the most recently created SAS data set is used as a DATA= input data set.

You can specify input data sets using one of the following combinations of options:

- DATA=, which provides both parameter estimates and the associated covariance matrix in a single input data set. See Example 1 in the next section.
- PARMS= and COVB=, which provide parameter estimates and the associated covariance matrix in separate data sets, respectively.
- PARMS= and XPXI=, which provide parameter estimates and the associated standard errors in a PARMS= data set, and the associated X'X inverse matrix in sa XPXI= data set.

The combination you use depends on the SAS procedure you used to create the parameter estimates and associated covariance matrix. 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. The next section illustrates these combinations.

Examples

The following statements generate five imputed data sets:

```
proc mi data=Fitness seed=37851 noprint out=miout;
   var Oxygen RunTime RunPulse;
run;
```

Example 1. REG: DATA= data set
The following statements generate regression coefficients:

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

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

```
                     Parameter Estimates from Imputed Data Sets

Obs    _Imputation_    _TYPE_     _NAME_       Intercept      RunTime     RunPulse

  1          1         PARMS                     102.650     -3.17736     -0.12495
  2          1         COV       Intercept        74.402     -0.85060     -0.38264
  3          1         COV       RunTime          -0.851      0.22194     -0.00881
  4          1         COV       RunPulse         -0.383     -0.00881      0.00280
  5          2         PARMS                      91.828     -2.95717     -0.07932
  6          2         COV       Intercept        56.905     -0.23604     -0.32023
  7          2         COV       RunTime          -0.236      0.12399     -0.00632
  8          2         COV       RunPulse         -0.320     -0.00632      0.00229
```

Figure 6. Output Data Set from PROC REG

The following statements combine the inferences from the imputed data sets:

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

```
                          The MIANALYZE Procedure

                  Multiple-Imputation Variance Information

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

    Intercept       20.626871      54.139158      78.891402          12
    RunTime          0.010739       0.149768       0.162656          23
    RunPulse         0.000378       0.002055       0.002509          18

                  Multiple-Imputation Variance Information

                                  Relative       Fraction
                                  Increase        Missing
                  Variable      in Variance    Information

                  Intercept        0.457197       0.345206
                  RunTime          0.086048       0.082107
                  RunPulse         0.220915       0.194029
```

Figure 7. Variance Information

The "Multiple-Imputation Variance Information" table displays the between-imputation, within-imputation, and total variances 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 parameter estimate are also displayed.

```
                          The MIANALYZE Procedure

                  Multiple-Imputation Parameter Estimates

                             Std Error
 Variable          Mean           Mean    95% Confidence Limits          DF

 Intercept      95.565478       8.882083     76.21308     114.9179          12
 RunTime        -3.060741       0.403306     -3.89504      -2.2264          23
 RunPulse       -0.092776       0.050087     -0.19801       0.0125          18

                  Multiple-Imputation Parameter Estimates

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

             Intercept              0      10.759354     <.0001
             RunTime                0      -7.589132     <.0001
             RunPulse               0      -1.852276     0.0805
```

Figure 8. 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. The table also displays a 95% mean confidence interval and a *t*-test with the associated *p*-value for the hypothesis that the variable mean is equal to mu0 as specified in the MU0= option for each variable.

When the MULT option is specified, the following table displays the within-imputation, between-imputation, and total covariance matrices:

```
                       The MIANALYZE Procedure

                 Within-Imputation Covariance Matrix

                      Intercept           RunTime          RunPulse

    Intercept       54.13915751       -0.55774289       -0.28357515
    RunTime         -0.55774289        0.14976830       -0.00606467
    RunPulse        -0.28357515       -0.00606467        0.00205481

                Between-Imputation Covariance Matrix

                      Intercept           RunTime          RunPulse

    Intercept       20.62687065       -0.43718694       -0.08759224
    RunTime         -0.43718694        0.01073940        0.00176325
    RunPulse        -0.08759224        0.00176325        0.00037828

                       Total Covariance Matrix

                      Intercept           RunTime          RunPulse

    Intercept       68.94083847       -0.71023016       -0.36110478
    RunTime         -0.71023016        0.19071506       -0.00772276
    RunPulse        -0.36110478       -0.00772276        0.00261659
```

Figure 9. Covariance Matrices

The "Multiple-Imputation Multivariate Inference" table displays the multivariate inference assuming proportionality of between and within covariance matrices.

```
                       The MIANALYZE Procedure

             Multiple-Imputation Multivariate Inference
        Assuming Proportionality of Between/Within Covariance Matrices

     Avg Relative
         Increase                               F for H0:
      in Variance     Num DF      Den DF         Mean=Mu0      Pr > F

         0.273401          3         135      2355.946171      <.0001
```

Figure 10. Multivariate Inference

Example 2. MIXED: PARMS= and COVB= data sets
PROC MIXED generates fixed-effect parameter estimates and associated covariance matrix for each model fit:

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

proc print data=mixparms (obs=6);
  var _Imputation_ Effect Estimate;
  title 'MIXED Model Coefficients';
run;

proc print data=mixcovb (obs=6);
  var _Imputation_ Effect Col1 Col2 Col3;
  title 'MIXED Covariance Matrices';
run;
```

```
          MIXED Model Coefficients from Imputed Data Sets

            Obs    _Imputation_     Effect       Estimate

              1          1         Intercept      102.65
              2          1         RunTime       -3.1774
              3          1         RunPulse      -0.1250
              4          2         Intercept     91.8277
              5          2         RunTime       -2.9572
              6          2         RunPulse     -0.07932
```

Figure 11. MIXED Model Coefficients

```
             Covariance Matrices from Imputed Data Sets

 Obs    _Imputation_     Effect          Col1         Col2         Col3

   1          1         Intercept     74.4024      -0.8506      -0.3826
   2          1         RunTime       -0.8506       0.2219     -0.00881
   3          1         RunPulse      -0.3826     -0.00881     0.002798
   4          2         Intercept     56.9053      -0.2360      -0.3202
   5          2         RunTime       -0.2360       0.1240     -0.00632
   6          2         RunPulse      -0.3202     -0.00632     0.002285
```

Figure 12. MIXED Covariance Matrices

The following MIANALYZE procedure uses PARMS= and COVB= options and produces the same results as in the previous regression example:

```
proc mianalyze parms=mixparms covb=mixcovb edf=28;
   var Intercept RunTime RunPulse;
run;
```

Example 3. GENMOD: PARMS= and COVB= data sets
PROC GENMOD generates parameter estimates and covariance matrix for each model fit:

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

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

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

```
          GENMOD Model Coefficients from Imputed Data Sets

            Obs    _Imputation_    Parameter    Estimate

              1          1         Intercept    102.6503
              2          1         RunTime       -3.1774
              3          1         RunPulse      -0.1250
              4          1         Scale          3.1099
              5          2         Intercept     91.8277
              6          2         RunTime       -2.9572
              7          2         RunPulse      -0.0793
              8          2         Scale          2.4866
```

Figure 13. GENMOD Model Coefficients

```
                Covariance Matrices from Imputed Data Sets

                               Row
   Obs    _Imputation_    Name          Prm1          Prm2          Prm3

     1          1         Prm1     67.202199     -0.768281     -0.345611
     2          1         Prm2     -0.768281     0.2004598     -0.007953
     3          1         Prm3     -0.345611     -0.007953     0.0025275
     4          1         Scale    3.868E-15     8.402E-16     -6.87E-17
     5          2         Prm1     51.398305     -0.213201     -0.289244
     6          2         Prm2     -0.213201     0.1119883     -0.005707
     7          2         Prm3     -0.289244     -0.005707      0.002064
     8          2         Scale    -3.82E-15     1.827E-16     1.431E-17
```

Figure 14. GENMOD Covariance Matrices

The following MIANALYZE procedure also uses PARMS= and COVB= options:

```
proc mianalyze parms=gmparms covb=gmcovb;
   var Intercept RunTime RunPulse;
run;
```

```
                          The MIANALYZE Procedure

                Multiple-Imputation Variance Information

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

  Intercept       20.626871       48.899884       73.652129          35
  RunTime          0.010739        0.135275        0.148162         529
  RunPulse         0.000378        0.001856        0.002310         104

                Multiple-Imputation Variance Information

                                 Relative       Fraction
                                 Increase        Missing
                Variable      in Variance    Information

                Intercept        0.506182       0.370635
                RunTime          0.095268       0.090415
                RunPulse         0.244585       0.211597
```

Figure 15. Variance Information

```
                             The MIANALYZE Procedure

                     Multiple-Imputation Parameter Estimates

                                   Std Error
Variable              Mean           Mean     95% Confidence Limits          DF

Intercept        95.565478       8.582082     78.14293     112.9880          35
RunTime          -3.060741       0.384918     -3.81690      -2.3046         529
RunPulse         -0.092776       0.048061     -0.18809       0.0025         104

                     Multiple-Imputation Parameter Estimates

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

             Intercept                0      11.135466      <.0001
             RunTime                  0      -7.951670      <.0001
             RunPulse                 0      -1.930359      0.0563
```

Figure 16. Parameter Estimates

The parameter estimates are identical to the estimates from the previous regression example, but the standard errors are slightly different because by default, PROC GENMOD computes maximum likelihood estimates for the standard errors.

Example 4. GLM: PARMS= and XPXI= data sets
PROC GLM generates the parameter estimates and XPX inverse matrix for each model fit:

```
proc glm data=miout;
  model Oxygen= RunTime RunPulse/inverse;
  by _Imputation_;
  ods output ParameterEstimates=glmparms
             InvXPX=glmxpxi;
run;

proc print data=glmparms (obs=6);
  var _Imputation_ Parameter Estimate;
  title 'GLM Model Coefficients';
run;

proc print data=glmxpxi (obs=6);
  var _Imputation_ Parameter Intercept RunTime
      RunPulse;
  title 'GLM X'X Inverse Matrices';
run;
```

```
                GLM Model Coefficients from Imputed Data Sets

                Obs    _Imputation_    Parameter        Estimate

                  1          1         Intercept     102.6503419
                  2          1         RunTime        -3.1773574
                  3          1         RunPulse       -0.1249526
                  4          2         Intercept      91.8276618
                  5          2         RunTime        -2.9571715
                  6          2         RunPulse       -0.0793226
```

Figure 17. GLM Model Coefficients

```
                           GLM X'X Inverse Matrices

Obs    _Imputation_    Parameter       Intercept          RunTime         RunPulse

  1          1         Intercept    6.9487124244     -0.079440359     -0.035736245
  2          1         RunTime      -0.079440359     0.0207275543     -0.000822365
  3          1         RunPulse     -0.035736245     -0.000822365      0.000261345
  4          1         Oxygen        102.6503419     -3.177357439     -0.124952649
  5          2         Intercept    8.3129028319     -0.034482111     -0.046780865
  6          2         RunTime      -0.034482111     0.0181124283     -0.000922977
```

Figure 18. GLM X'X Inverse Matrices

The following MIANALYZE procedure step uses the PARMS= and XPXI= options and produces the same results as in the previous regression example:

```
proc mianalyze parms=glmparms xpxi=glmxpxi edf=28;
   var Intercept RunTime RunPulse;
run;
```

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.

References

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

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.

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.

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 1.0