Journal of Statistical Software

December 2011, Volume 45, Issue 6. http://www.jstatsoft.org/

Multiple Imputation Using SAS Software

Yang Yuan
SAS Institute Inc.

Abstract

Multiple imputation provides a useful strategy for dealing with data sets that have missing values. Instead of filling in a single value for each missing value, a 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 of parameter estimates and their associated standard errors from different imputed data sets is essentially the same. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values.

This paper reviews methods for analyzing missing data and applications of multiple imputation techniques. This paper presents the SAS/STAT `MI` and `MIANALYZE` procedures, which perform inference by multiple imputation under numerous settings. `PROC MI` implements popular methods for creating imputations under monotone and nonmonotone (arbitrary) patterns of missing data, and `PROC MIANALYZE` analyzes results from multiply imputed data sets.

Keywords: multiple imputation, monotone missing pattern, Markov chain Monte Carlo.

1. Introduction

Most SAS statistical procedures exclude observations with any missing variable values from the analysis. Although using only complete cases is simple, information that is in the incomplete cases is lost. Excluding observations with missing values also ignores the possible systematic difference between the complete cases and incomplete cases, and the resulting inference might not be applicable to the population of all cases, especially with a smaller number of complete cases.

There are several approaches to handling missing data. The first approach uses all available data, which ignores any incomplete data in the cases. For example, the `CORR` procedure estimates a variable mean by using all cases with nonmissing values for this variable, ignoring the possible missing values in other variables. The CORR procedure also estimates a correlation by using all cases with nonmissing values for this pair of variables. This estimation might make better use of the available data, but the resulting correlation matrix might not be positive definite.

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

Instead of filling in a single value for each missing value, a multiple imputation procedure replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute (Rubin 1987). The multiply imputed data sets are then analyzed by using standard procedures for complete data and combining the results from these analyses. No matter which complete-data analysis is used, the process of combining results from different data sets is essentially the same.

Multiple imputation does not attempt to estimate each missing value through simulated values, but rather to represent a random sample of the missing values. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values; for example, valid confidence intervals for parameters.

Multiple imputation inference involves three distinct phases:

- 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 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. After the m complete data sets are analyzed by using standard procedures, the MIANALYZE procedure can then be used to generate valid statistical inferences about these parameters by combining results from the m complete data sets.

Documentation for SAS/STAT 9.2, SAS/STAT 9.22, and SAS/STAT 9.3 is available online (SAS Institute Inc. 2011a).

2. Multiple imputation methods in the MI procedure

This section describes methods that are available in PROC MI. PROC MI assumes that the missing data are missing at random (MAR)—that is, the probability that an observation is missing might depend on $\mathbf{Y}_{obs}$, but not on $\mathbf{Y}_{mis}$ (Rubin 1976, 1987). Furthermore, PROC MI also assumes 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.

Pattern of missingness	Type of imputed variable	Available methods
Monotone	Continuous	Monotone regression
		Monotone predicted mean matching
		Monotone propensity score
Monotone	Classification (ordinal)	Monotone logistic regression
Monotone	Classification (nominal)	Monotone discriminant function
Arbitrary	Continuous	MCMC full-data imputation
		MCMC monotone-data imputation

Table 1: Imputation methods in `PROC MI`.

The imputation method of choice depends on the pattern of missingness in the data and the type of the imputed variable. A data set with variables Y_1, Y_2, ..., Y_p (in that order) is said to have a *monotone missing pattern* when the event that a variable Y_j is missing for a particular individual implies that all subsequent variables Y_k, $k > j$, are missing for that individual. Table 1 summarizes the available methods.

For data sets with monotone missing patterns, the variables with missing values can be imputed sequentially with covariates constructed from their corresponding sets of preceding variables. To impute missing values for a continuous variable, one of the following methods can be used: a regression method (Rubin 1987), a predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996), or a propensity score method (Lavori, Dawson, and Shera 1995). To impute missing values for a classification variable, one of the following methods can be used: a logistic regression method when the classification variable has a binary or ordinal response, or a discriminant function method when the classification variable has a binary or nominal response.

For data sets with arbitrary missing patterns, a Markov chain Monte Carlo (MCMC) method that assumes multivariate normality can be used to impute missing values (Schafer 1997). The MCMC method can be used to impute either all the missing values or just enough missing values to make the imputed data sets have monotone missing patterns. A monotone missing data pattern offers greater flexibility in the choice of imputation models (such as the monotone regression method) that do not use Markov chains. A different set of covariates can also be specified for each imputed variable.

For data sets with arbitrary missing patterns, a fully conditional specification (FCS) method can also be used to impute missing values for both continuous and classification variables (Brand 1999; van Buuren 2007). The FCS method assumes the existence of a joint distribution for all variables. The method does not start with an explicitly specified multivariate distribution for all variables, but rather uses a separate conditional distribution for each imputed variable. This feature is not described further in this paper, but is described in the documentation of the MI procedure for SAS/STAT 9.3 (SAS Institute Inc. 2011b).

2.1. Methods for data sets with monotone missing data patterns

For a data set with a monotone missing data pattern, one of the following methods can be used: a regression method, a predictive mean matching method, or a propensity score method to impute missing values for a continuous variable; a logistic regression method for a classification variable with a binary or ordinal response; or a discriminant function method for a classification variable with a binary or nominal response.

For a variable with missing values, a model is fitted using observations with observed values for the variable. With this resulting model, a new model is drawn and is used to impute missing values for the variable. The missing values are imputed sequentially for variables in the order given by the `VAR` statement.

That is, for a variable Y_j with missing values, the missing values are imputed from the distribution

$$Y_j \sim P(\, Y_j \,|\, Y_1, Y_2, \ldots, Y_{j-1} \,)$$

An example is a regression model

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

where X_1, X_2, …, X_k are the covariates generated from preceding variables Y_1, Y_2, … , Y_{j-1}. The following steps are used to impute missing values for Y_j at each imputation:

1. The regression model is fitted using 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.

2. New parameters $\beta_* = (\beta_{*0}, \beta_{*1}, ..., \beta_{*(k)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters (Rubin 1987). That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$, $\hat{\sigma}_j^2$, and $\mathbf{V}_j$. The variance is drawn as

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

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

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

   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.

3. 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.

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 that is selected from the specified number of nearest observations to the predicted value from the simulated regression model (Rubin 1987). The predictive mean matching method ensures that imputed values are plausible, and it might be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001).

2.2. MCMC methods for data sets with arbitrary missing patterns

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, a Markov chain long enough for the distribution of the elements to stabilize to a common distribution is constructed. This stationary distribution is the distribution of interest. Repeatedly simulating steps of the chain simulates draws from the distribution of interest Schafer (1997).

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, the entire joint posterior distribution of the unknown quantities can be simulated and simulation-based estimates of posterior parameters can be obtained.

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 the variables with missing values for observation i are denoted by $Y_{i(mis)}$ and the variables with observed values are denoted 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. Other informative priors can also be used. For example, a prior information about the covariance matrix might help stabilize the inference about the mean vector for a near singular covariance matrix.

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 $p(\theta|Y_{obs}, Y_{mis}^{(t+1)})$. The two steps are iterated long enough for the results to reliably simulate an approximately independent draw of the missing values for a multiply imputed data set (Schafer 1997).

3. The `MI` procedure

`PROC MI` provides various methods to create multiply imputed data sets for incomplete multivariate data that can be analyzed using standard SAS procedures. Table 2 summarizes the available statements in `PROC MI`.

The imputation method of choice depends on the pattern of missingness in the data and the type of the imputed variable. For a data set with a monotone missing pattern, the `MONOTONE` statement can be used to specify applicable monotone imputation methods; otherwise, the `MCMC` statement can be used assuming multivariate normality.

Statement	Description
BY	Specifies groups in which separate sets of multiple imputations are performed
CLASS	Specifies the classification variables in the VAR statement
EM	Computes the maximum likelihood estimate (MLE) of data with missing values by expectation-maximization (EM) algorithm assuming a multivariate normal distribution
FREQ	Specifies the variable that represents the frequency of occurrence in the observation
MCMC	Specifies Markov chain Monte Carlo imputation methods
MONOTONE	Specifies imputation methods for a data set with a monotone missing pattern
TRANSFORM	Specifies the variables to be transformed in the imputation process
VAR	Specifies the variables to be analyzed

Table 2: Statements in PROC MI.

Option	Description
DATA=	Specifies the input data set
NIMPUTE=	Specifies the number of imputations
OUT=	Specifies the output SAS data set in which to put the imputation results
ROUND=	Specifies units to round imputed variable values
MINIMUM=	Specifies minimum values for imputed variable values
MAXIMUM=	Specifies maximum values for imputed variable values
SEED=	Specifies a positive integer that is used to start the pseudorandom number generator
MU0=	Specifies variable means under the null hypothesis in the t-test for location

Table 3: Key options in PROC MI.

The TRANSFORM statement specifies the variables to be transformed before the imputation process; the imputed values of these transformed variables are reverse-transformed to the original forms before the imputation. The Box-Cox, exponential, logarithmic, logit, and power transformations can be used for the variables.

Table 3 lists key options available in the PROC MI statement. Often, as few as three to five imputations are adequate in multiple imputation (Rubin 1996). If the NIMPUTE= option is not specified, NIMPUTE=5 is used. The OUT= option specifies the output SAS data set that includes an identification variable, _IMPUTATION_, to identify the imputation number.

3.1. MONOTONE statement

The MONOTONE statement specifies monotone methods to impute variables for a data set with a monotone missing pattern. A VAR statement must be specified, and the data set must have a monotone missing pattern with variables ordered in the VAR list. Table 4 lists available methods in the MONOTONE statement.

For each imputed variable, the imputation method and, optionally, a set of the effects as covariates to impute the variable can be specified. Each effect is a variable or a combination of variables preceding the imputed variable in the VAR statement. If no covariates are specified, then all preceding variables are used as the covariates.

Option	Description
`REG`	Specifies the regression method
`REGPMM`	Specifies the predictive mean matching method
`PROPENSITY`	Specifies the propensity scores method
`DISCRIM`	Specifies the discriminant function method
`LOGISTIC`	Specifies the logistic regression method

Table 4: Summary of imputation methods in `MONOTONE` statement.

With a `MONOTONE` statement, the variables are imputed sequentially in the order given by the `VAR` statement. For a continuous variable, the following methods can be used: a regression method, a regression predicted mean matching method, or a propensity score method to impute missing values. For a nominal classification variable, a discriminant function method can be used to impute missing values without using the ordering of the class levels. For an ordinal classification variable, a logistic regression method can be used to impute missing values by using the ordering of the class levels. For a binary classification variable, either a discriminant function method or a logistic regression method can be used.

3.2. Example 1: Regression method for monotone missing pattern data

This example uses the regression method to impute missing values for variables in the following `Fish` data set, which has a monotone missing pattern. The data set contains two species of the fish (Bream and Pike) and three measurements: `Length`, `Height`, `Width`. Some values have been set to missing, and the resulting data set has a monotone missing pattern in the variables `Length`, `Height`, `Width`, and `Species`.

```
data Fish;
   title 'Fish Measurement Data';
   input Species $ Length Height Width @@;
   datalines;

Bream   30.0  11.520  4.020        .   31.2  12.480  4.306
Bream   31.1  12.378  4.696    Bream   33.5  12.730  4.456
    .   34.0  12.444    .      Bream   34.7  13.602  4.927
Bream   34.5  14.180  5.279    Bream   35.0  12.670  4.690
Bream   35.1  14.005  4.844    Bream   36.2  14.227  4.959
    .   36.2  14.263    .      Bream   36.2  14.371  4.815
Bream   36.4  13.759  4.368    Bream   37.3  13.913  5.073
Bream   37.2  14.954  5.171    Bream   37.2  15.438  5.580
Bream   38.3  14.860  5.285    Bream   38.5  14.938  5.198
    .   38.6  15.633  5.134    Bream   38.7  14.474  5.728
Bream   39.5  15.129  5.570        .   39.2  15.994    .
Bream   39.7  15.523  5.280    Bream   40.6  15.469  6.131
    .   40.5     .      .      Bream   40.9  16.360  6.053
Bream   40.6  16.362  6.090    Bream   41.5  16.517  5.852
Bream   41.6  16.890  6.198    Bream   42.6  18.957  6.603
Bream   44.1  18.037  6.306    Bream   44.0  18.084  6.292
```

```
Bream   45.3  18.754  6.750     Bream   45.9  18.635  6.747
Bream   46.5  17.624  6.371
Pike    34.8   5.568  3.376     Pike    37.8   5.708  4.158
Pike    38.8   5.936  4.384        .    39.8     .      .
Pike    40.5   7.290  4.577     Pike    41.0   6.396  3.977
   .    45.5   7.280  4.323     Pike    45.5   6.825  4.459
Pike    45.8   7.786  5.130     Pike    48.0   6.960  4.896
Pike    48.7   7.792  4.870     Pike    51.2   7.680  5.376
Pike    55.1   8.926  6.171        .    59.7  10.686    .
Pike    64.0   9.600  6.144     Pike    64.0   9.600  6.144
Pike    68.0  10.812  7.480
;
```

The following statements invoke the `MI` procedure and request the regression method for variables `Height` and `Width` and the logistic regression method for the variable `Species`. The resulting data set is named `OutFish`.

```
proc mi data=Fish seed=1305417 out=OutFish;
   class Species;
   monotone reg(Height Width/ details)
            logistic( Species= Length Height Width Height*Width/ details);
   var Length Height Width Species;
run;
```

The `Model Information` table describes the method and options used in the multiple imputation process. By default, `NIMPUTE=5`: five imputations are created for the missing data. The `Monotone Model Specification` table displays specific monotone methods used in the imputation.

```
                          The MI Procedure

                         Model Information

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


                   Monotone Model Specification

                                         Imputed
                   Method                Variables

                   Regression            Height Width
                   Logistic Regression   Species
```

The `Missing Data Patterns` table lists distinct missing data patterns with their corresponding frequencies and percentages. An ‘X’ indicates that the variable is observed in the corresponding group, and a '.' indicates that the variable is missing. The variable means for continuous variables in each group are also displayed.

```
                            Missing Data Patterns

    Group     Length     Height     Width     Species          Freq      Percent

        1     X          X          X         X                  43        82.69
        2     X          X          X         .                   3         5.77
        3     X          X          .         .                   4         7.69
        4     X          .          .         .                   2         3.85


                   -----------------Group Means----------------
            Group         Length         Height          Width

                1      41.997674      12.819512       5.359860
                2      38.433333      11.797667       4.587667
                3      42.275000      13.346750              .
                4      40.150000              .              .
```

The `DETAILS` option in the `REG` option displays the regression coefficients in the regression model that are estimated from the observed data and the regression coefficients that are used in each imputation.

```
                    Regression Models for Monotone Method

Imputed                              ----------------Imputation----------------
Variable    Effect       Obs-Data             1              2              3

Height      Intercept     0.00173     -0.152270      -0.136544      -0.064801
Height      Length       -0.22453     -0.133455      -0.155687      -0.319043


             Imputed                 ---------Imputation---------
             Variable  Effect                 4              5

             Height    Intercept       0.036585       0.088415
             Height    Length         -0.108935      -0.215399


                    Regression Models for Monotone Method

Imputed                              ----------------Imputation----------------
Variable    Effect       Obs-Data             1              2              3

Width       Intercept     0.00682      0.054140       0.018049      -0.015137
```

Width	Length	0.75519	0.838485	0.768945	0.789577
Width	Height	0.73890	0.832117	0.831748	0.809482

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

Similarly, the `DETAILS` option in the `LOGISTIC` option displays the regression coefficients in the logistic regression model that are estimated from the observed data and the regression coefficients that are used in each imputation.

Logistic Models for Monotone Method

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

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

The following statements list the first 10 observations of `OutFish` with imputed values.

```
proc print data=OutFish(obs=10);
   var _Imputation_ Species Length Height Width;
   title 'First 10 Observations of the Imputed Data Set';
run;
```

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length	Height	Width

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

3.3. `MCMC` statement

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. Table 5 summarizes the key options available for the `MCMC` statement.

The key options for the imputation details are:

- `CHAIN=SINGLE | MULTIPLE`: The `CHAIN=` option specifies whether a single chain (`CHAIN=SINGLE`) is used for all imputations or a separate chain (`CHAIN=MULTIPLE`) is used for each imputation (Schafer 1997). The default is `CHAIN=SINGLE`.

- `IMPUTE=MONOTONE | FULL`: The `IMPUTE=` option specifies whether a full-data imputation (`IMPUTE=FULL`) is used for all missing values or a monotone-data imputation (`IMPUTE=MONOTONE`) is used for a subset of missing values to make the imputed data sets have a monotone missing pattern. The default is `IMPUTE=FULL`.

Option	Description
Data sets	
`INEST=`	Inputs parameter estimates for imputations
`OUTEST=`	Outputs parameter estimates used in imputations
`OUTITER=`	Outputs parameter estimates used in iterations
Imputation details	
`CHAIN=`	Specifies single or multiple chain
`IMPUTE=`	Specifies monotone or full imputation
`NBITER=`	Specifies the number of burn-in iterations for each chain
`NITER=`	Specifies the number of iterations between imputations in a chain
ODS output graphics	
`PLOTS=TRACE`	Displays trace plots of parameters from iterations
`PLOTS=ACF`	Displays autocorrelation plots of parameters from iterations

Table 5: Summary of key options in MCMC statement.

- `NBITER=`*numbers*: The `NBITER=` option specifies the number of burn-in iterations before the first imputation in each chain. The default is `NBITER=200`.
- `NITER=`*numbers* The `NITER=` option specifies the number of iterations between imputations in a single chain. The default is `NITER=100`.

3.4. Example 2: MCMC method for arbitrary missing pattern data

This example uses the MCMC method to impute missing values for variables in a data set with an arbitrary missing pattern. The following `Fitness` data set has been altered to contain an arbitrary missing pattern. These measurements were made on men involved in a physical fitness course at N.C. State University. Certain values have been set to missing and the resulting data set has an arbitrary missing pattern. Only selected variables of `Oxygen` (intake rate, ml per kg body weight per minute), `Runtime` (time to run 1.5 miles in minutes), `RunPulse` (heart rate while running) are used.

```
data Fitness;
   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 `OutFitness`. These statements also create an iteration plot for the successive estimates of the variable `Oxygen` and an autocorrelation function plot for `Oxygen`.

```
ods graphics on;
proc mi data=Fitness nimpute=4 seed=501213
        mu0=50 10 180 out=OutFitness;
   em;
   mcmc plots=(trace(mean(Oxygen)) acf(mean(Oxygen)));
```

```
   var Oxygen RunTime RunPulse;
run;
ods graphics off;
```

The `Model Information` table describes the method and options used.

```
                           The MI Procedure

                          Model Information

      Data Set                              WORK.FITNESS
      Method                                MCMC
      Multiple Imputation Chain             Single Chain
      Initial Estimates for MCMC            EM Posterior Mode
      Start                                 Starting Value
      Prior                                 Jeffreys
      Number of Imputations                 4
      Number of Burn-in Iterations          200
      Number of Iterations                  100
      Seed for random number generator      501213
```

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

```
                         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               .
```

The expectation-maximization (EM) algorithm is a technique that finds maximum likelihood estimates for parametric models for incomplete data (Little and Rubin 2002). By default, the procedure uses the statistics from the available cases in the data as the initial estimates for EM algorithm, and the correlations are set to zero. With the `EM` statement, the initial parameter estimates for the EM algorithm and the resulting maximum likelihood estimates are displayed.

```
                    Initial Parameter Estimates for EM

_TYPE_     _NAME_            Oxygen         RunTime        RunPulse

MEAN                      47.116179       10.688214      171.863636
COV        Oxygen         29.301078               0               0
COV        RunTime                0        1.904067               0
COV        RunPulse               0               0      102.885281


                       EM (MLE) Parameter Estimates

_TYPE_     _NAME_            Oxygen         RunTime        RunPulse

MEAN                      47.104077       10.554858      171.381669
COV        Oxygen         27.797931       -6.457975      -18.031298
COV        RunTime        -6.457975        2.015514        3.516287
COV        RunPulse      -18.031298        3.516287       97.766857
```

The EM algorithm can also be used to compute posterior modes, the parameter estimates with the highest observed-data posterior density. These posterior modes are used to begin the MCMC process.

```
                      EM (Posterior Mode) Estimates

_TYPE_     _NAME_            Oxygen         RunTime        RunPulse

MEAN                      47.103766       10.554320      171.382196
COV        Oxygen         24.549967       -5.726112      -15.926036
COV        RunTime        -5.726112        1.781407        3.124798
COV        RunPulse      -15.926036        3.124798       83.164045
```

After the completion of the specified four imputations, the `Variance Information` table displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences.

```
                          Variance Information

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

Oxygen	0.067395	0.962300	1.046544	24.54
RunTime	0.000211	0.064026	0.064290	28.062
RunPulse	0.801827	3.441013	4.443298	15.929

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.087544	0.084443	0.979326
RunTime	0.004129	0.004123	0.998970
RunPulse	0.291276	0.250570	0.941050

The `Parameter Estimates` table displays the estimated mean and standard error of the mean for each variable. The table also displays a 95% confidence interval for the variable mean and a t statistic with the associated p-value for the hypothesis that the population mean is equal to the value specified with the `MU0=` option.

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF
Oxygen	47.129771	1.023007	45.0208	49.2387	24.54
RunTime	10.583493	0.253555	10.0642	11.1028	28.062
RunPulse	172.041037	2.107913	167.5708	176.5112	15.929

Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Oxygen	46.783898	47.395550	50.000000	-2.81	0.0097
RunTime	10.570896	10.599616	10.000000	2.30	0.0290
RunPulse	170.934337	173.122002	180.000000	-3.78	0.0017

With the `TRACE(MEAN(OXYGEN))` option, the procedure displays a trace plot for the mean of `Oxygen`, as shown in Figure 1. The plot shows no apparent trends for the variable `Oxygen`.

With the `ACF(MEAN(OXYGEN))` option, an autocorrelation plot for the mean of `Oxygen` is displayed, as shown in Figure 2. It shows no significant positive or negative autocorrelation.

The following statements list the first 10 observations of the output data set `OutFitness`:

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

First 10 Observations of the Imputed Data Set

Run

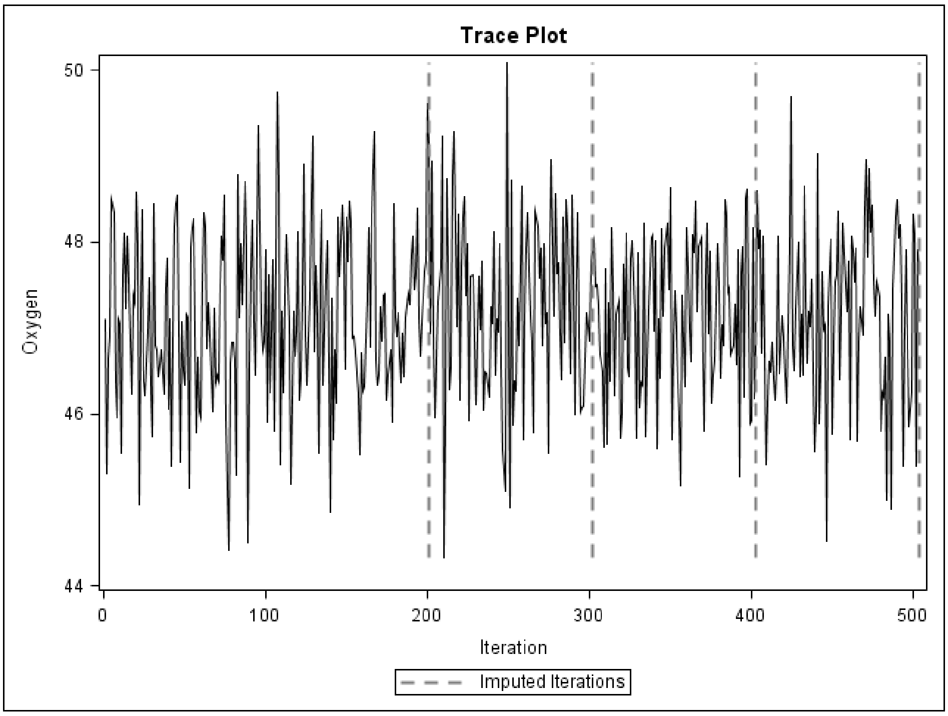

Figure 1: Trace plot for `Oxygen`.

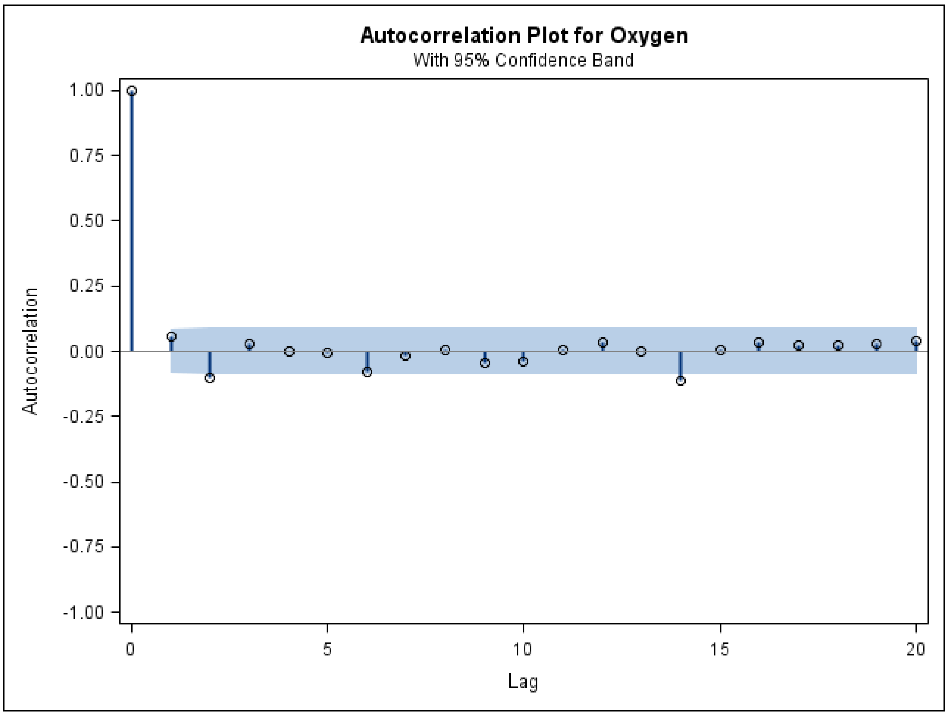

Figure 2: Autocorrelation function plot for `Oxygen`.

```
Obs    _Imputation_     Oxygen    RunTime      Pulse

  1          1         44.6090    11.3700    178.000
  2          1         45.3130    10.0700    185.000
  3          1         54.2970     8.6500    156.000
  4          1         59.5710     8.0747    155.925
  5          1         49.8740     9.2200    176.837
  6          1         44.8110    11.6300    176.000
  7          1         42.8857    11.9500    176.000
  8          1         46.9992    10.8500    173.099
  9          1         39.4420    13.0800    174.000
 10          1         60.0550     8.6300    170.000
```

4. Combining inferences from imputed data sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Let $\hat{Q}_i$ and $\hat{U}_i$ be the point and variance estimates from the 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

Statement	Description
BY	Specifies groups in which separate sets of multiple imputations are performed
CLASS	Specifies classification variables in the MODELEFFECTS statement
MODELEFFECTS	Lists the effects in the data set to be analyzed
STDERR	Lists standard errors associated with effects
TEST	Tests linear hypotheses about the parameters

Table 6: Statements in PROC MIANALYZE.

is inappropriate. Barnard and Rubin (1999) recommend the use of an adjusted degrees of freedom, v_m^*,

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

where

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

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

5. 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. Table 6 lists available statements in PROC MIANALYZE.

The 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.

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

Table 7 lists available options in the PROC MIANALYZE statement. Input data sets are specified based on the requested type of inference. The appropriate combination depends on the type of inference and the SAS procedure that was used to create the data sets. For example, if PROC REG was used to create an OUTEST= data set of type EST that contains the parameter estimates and covariance matrix, the DATA= option would be used to read the OUTEST= data set.

5.1. Example 3: Reading results from PARMS= and COVB= data sets

This example creates data sets that contain parameter estimates and corresponding covariance

Option	Description
Input data sets	
DATA=	Specifies the input COV, CORR, or EST type data set
DATA=	Specifies the input data set for parameter estimates and standard errors
PARMS=	Specifies the input data set for parameter estimates
PARMINFO=	Specifies the input data set for parameter information
COVB=	Specifies the input data set for covariance matrices
XPXI=	Specifies the input data set for $(\mathbf{X}'\mathbf{X})^{-1}$ matrices
Statistical analysis	
ALPHA=	Specifies the level for the confidence interval
EDF=	Specifies the complete-data degrees of freedom
THETA0=	Specifies parameters under the null hypothesis
Printed output	
WCOV	Displays the within-imputation covariance matrix
BCOV	Displays the between-imputation covariance matrix
TCOV	Displays the total covariance matrix
MULT	Displays multivariate inferences

Table 7: Options in PROC MIANALYZE.

matrices computed by a logistic regression model for imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC LOGISTIC to generate the parameter estimates and covariance matrix for each imputed data set stored in OutFish:

```
proc logistic data=OutFish;
   class Species;
   model Species= Length / covb;
   by _Imputation_;
   ods output ParameterEstimates=lgparms CovB=lgcovb;
run;
```

The following statements display the ODS output PARAMETERESTIMATES= data set from PROC LOGISTIC for the first two imputed data sets:

```
proc print data=lgparms(obs=4);
   title 'Logistic Model Coefficients (First Two Imputations)';
   var _Imputation_ Variable Estimate StdErr;
run;
```

The Logistic Model Coefficients (First Two Imputations) table displays the output parameter estimates and standard errors for the first two imputed data sets.

Logistic Model Coefficients (First Two Imputations)

```
Obs    _Imputation_    Variable     Estimate     StdErr

  1          1         Intercept     11.6446     3.5105
  2          1         Length        -0.2599     0.0836
  3          2         Intercept     10.9976     3.3477
  4          2         Length        -0.2477     0.0802
```

The following statements display the `ODS` output `COVB=` data set from `PROC LOGISTIC` for the first two imputed data sets:

```
proc print data=lgcovb(obs=4);
   title 'Logistic Covariance Matrices (First Two Imputations)';
run;
```

The `Logistic Covariance Matrices (First Two Imputations)` table displays the output covariance matrices for the first two imputed data sets.

```
           Logistic Covariance Matrices (First Two Imputations)

Obs    _Imputation_    Parameter    Intercept       Length

  1          1         Intercept      12.3239     -0.29171
  2          1         Length        -0.29171     0.006986
  3          2         Intercept     11.20695     -0.26691
  4          2         Length        -0.26691     0.006433
```

The following statements use the `MIANALYZE` procedure to read parameter estimates in the `PARMS=` data set and the associated covariance matrix in the `COVB=` data set:

```
proc mianalyze parms=lgparms
               covb=lgcovb;
   modeleffects Intercept Length;
run;
```

The `Model Information` table lists the input data sets and the number of imputations. The `Variance Information` table displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

```
                     The MIANALYZE Procedure

                        Model Information

              PARMS Data Set           WORK.LGPARMS
              COVB Data Set            WORK.LGCOVB
              Number of Imputations    5


                      Variance Information
```

	Variance			
Parameter	Between	Within	Total	DF
Intercept	0.372426	12.323246	12.770157	3266
Length	0.000126	0.006976	0.007127	8927.7

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.036266	0.035587	0.992933
Length	0.021625	0.021386	0.995741

The `Parameter Estimates` table displays the parameter estimate and standard error of the regression coefficient for each variable. With an estimate -0.25906 and its associated p-value 0.0022 for the parameter `Length`, the length of `Bream` is significantly shorter than the length of `Pike`.

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	11.614996	3.573536	4.60840	18.62159	3266
Length	-0.259060	0.084419	-0.42454	-0.09358	8927.7

Parameter	Minimum	Maximum
Intercept	10.997552	12.217637
Length	-0.270055	-0.247650

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	3.25	0.0012
Length	0	-3.07	0.0022

5.2. Example 4: Reading results from a `DATA=` 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 in each imputed data set stored in `OutFitness`:

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

The following statements display the output `OUTEST=` data set from `PROC REG` for the first two imputed data sets:

```
proc print data=regest(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'REG Model Coefficients (First Two Imputations)';
run;
```

The `REG Model Coefficients (First Two Imputations)` table displays regression coefficients and their covariance matrices for the first two imputed data sets.

REG Model Coefficients (First Two Imputations)

Obs	_Imputation_	_TYPE_	_NAME_	Intercept	RunTime	RunPulse
1	1	PARMS		95.0397	-3.39792	-0.06817
2	1	COV	Intercept	66.8696	-0.81692	-0.33708
3	1	COV	RunTime	-0.8169	0.14815	-0.00436
4	1	COV	RunPulse	-0.3371	-0.00436	0.00223
5	2	PARMS		92.0495	-3.29472	-0.06029
6	2	COV	Intercept	81.2318	-0.86457	-0.41496
7	2	COV	RunTime	-0.8646	0.13230	-0.00308
8	2	COV	RunPulse	-0.4150	-0.00308	0.00259

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

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

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.

The `Model Information` table lists the input data set and the number of imputations. The `Variance Information` table displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

The MIANALYZE Procedure

Model Information

Data Set	WORK.REGEST

Number of Imputations 4

Variance Information

Parameter	Between (Variance)	Within (Variance)	Total (Variance)	DF
Intercept	8.872382	80.351747	91.442225	20.683
RunTime	0.022390	0.137756	0.165744	18.038
RunPulse	0.000119	0.002602	0.002750	24.191

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.138024	0.129776	0.968575
RunTime	0.203169	0.184223	0.955972
RunPulse	0.057212	0.055958	0.986204

The `Parameter Estimates` table displays the parameter estimate and standard error of the regression coefficient for each variable. The table also displays a 95% mean confidence interval and a t test with the associated p-value for the hypothesis that the regression coefficient is equal to zero. Since the p-value for `RunPulse` is 0.2987, this variable can be removed from the regression model.

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	91.220141	9.562543	71.31514	111.1251	20.683
RunTime	-3.260213	0.407116	-4.11540	-2.4050	18.038
RunPulse	-0.055700	0.052445	-0.16390	0.0525	24.191

Parameter	Minimum	Maximum
Intercept	88.378636	95.039651
RunTime	-3.397916	-3.047243
RunPulse	-0.068166	-0.042970

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

```
Intercept          0          9.54    <.0001
RunTime            0         -8.01    <.0001
RunPulse           0         -1.06    0.2987
```

Acknowledgments

The author is grateful to Bob Rodriguez and Anne Baxter of the SAS Advanced Analytics Division for their valuable assistance in the preparation of this manuscript.

References

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

Brand JPL (1999). *Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets*. Ph.D. thesis, Erasmus University Rotterdam.

Heitjan F, Little RJA (1991). "Multiple Imputation for the Fatal Accident Reporting System." *Applied Statistics*, **40**, 13–29.

Horton NJ, Lipsitz SR (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 PW, Dawson R, Shera D (1995). "A Multiple Imputation Strategy for Clinical Trials with Truncation of Patient Data." *Statistics in Medicine*, **14**, 1913–1925.

Little RJA, Rubin DB (2002). *Statistical Analysis with Missing Data*. 2nd edition. John Wiley & Sons, Hoboken.

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

Rubin DB (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons.

Rubin DB (1996). "Multiple Imputation After 18+ Years." *Journal of the American Statistical Association*, **91**, 473–489.

SAS Institute Inc (2011a). *SAS/STAT Product Documentation*. SAS Institute Inc., Cary, NC. URL http://support.sas.com/documentation/onlinedoc/stat/.

SAS Institute Inc (2011b). *SAS/STAT User's Guide – Procedures*. SAS Institute Inc., Cary, NC. URL http://support.sas.com/documentation/onlinedoc/stat/indexproc.html#stat93.

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

Schenker N, Taylor JMG (1996). "Partially Parametric Techniques for Multiple Imputation." *Computational Statistics & Data Analysis*, **22**, 425–446.

van Buuren S (2007). "Multiple Imputation of Discrete and Continuous Data by Fully Conditional Specification." *Statistics Methods in Medical Research*, **16**, 219–242.

Affiliation:

Yang Yuan
SAS Institute Inc.
Rockville, MD, United States of America
E-mail: Yang.Yuan@sas.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 45, Issue 6 *Submitted:* 2009-10-27
December 2011 *Accepted:* 2011-03-30