Paper 283-2013

A Flexible Method to Apply Multiple Imputation Using SAS/IML® Studio

Xue Yao, University of Manitoba, Winnipeg, MB, Canada

Lisa M. Lix, University of Manitoba, Winnipeg, MB, Canada

ABSTRACT

Multiple imputation methods are widely used for missing data problems in various scientific fields. Imputation methods can also be applied to measurement error problems, which arise frequently in many data-analytic problems. SAS/STAT® software offers the MI and MIANALYZE procedures for creating and analyzing multiple imputation data. PROC MI can be used to impute continuous or categorical variables with a monotone missingness pattern and continuous variables with an arbitrary missingness pattern. This paper provides an imputation method developed using SAS/IML® Studio for categorical variables with an arbitrary missingness pattern. This method expands the SAS analyst's ability to apply multiple imputation methods to a wide variety of variables. The method is illustrated using an example of measurement (i.e., misclassification) error in disease diagnoses.

INTRODUCTION

Since Rubin (1987) proposed the multiple imputation method, it has become an important and influential approach in the statistical analysis of incomplete data and has been applied to a number of different types of missing data problems. For example, researchers have demonstrated that the multiple imputation method can be used to correct for measurement error by treating the unobserved true measurements as missing observations (Cole, Chu, & Greenland, 2006). Multiple imputation will result in more accurate statistical inference than complete case analysis methods.

The multiple imputation method has three primary steps as depicted in Figure 1 (Mayer, Muche, & Hohl, 2012). First, *m*>1 complete datasets are obtained by replacing the missing values with *m* imputed plausible values. Then the *m* complete datasets are analyzed using standard statistical analysis techniques. Third, the estimates of the parameters of interest from the *m* complete datasets are combined, typically by averaging them.

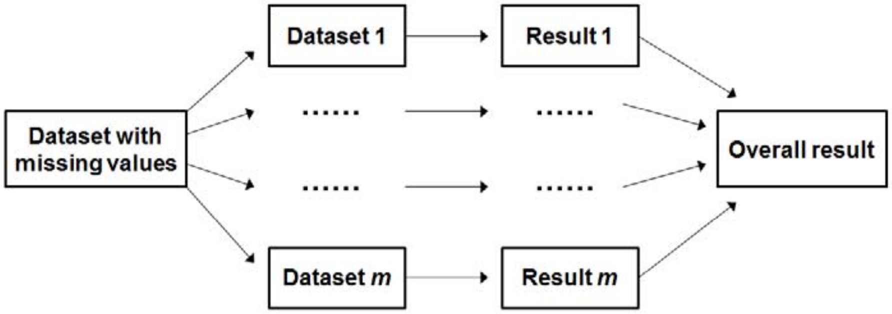


Figure 1. A Schematic Diagram of the Multiple Imputation Method

In SAS, the MI and MIANALYZE procedures can be used to implement the first and third steps in the multiple imputation method (Figure 2). The MI procedure creates multiple imputed datasets for the first step. Then the complete datasets are analyzed by standard SAS statistical procedures, such as PROC GLM. The MIANALYZE procedure is used to generate valid statistical inferences about the parameters of interest by combining results from the standard procedures based on complete datasets. The patterns of missingness (i.e., monotone or arbitrary) and type of variable (i.e., continuous or categorical) determine the MI method to use (Figure 3). A monotone missing data pattern, which may arise in longitudinal data, is one in which the absence of an observation implies that all subsequent observations on the same variable will be missing. An arbitrary missing data pattern is one in which the missing data appear regardless of the variable or observation. The recommended MI methods for different patterns of missing data and types of imputed variables are described in Table 1 (Refaat, 2007).

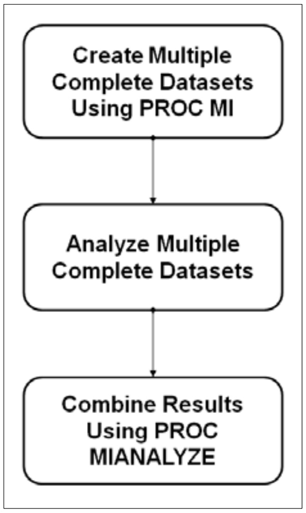
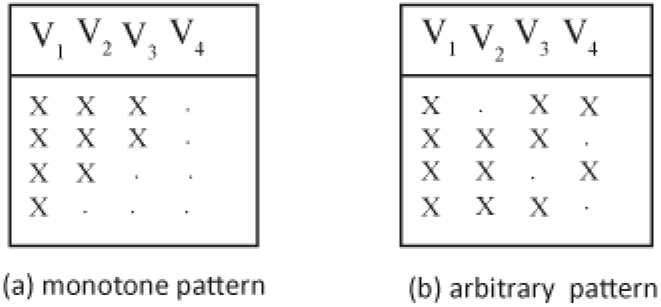

Create Multiple Complete Datasets Using PROC MI
Analyze Multiple Complete Datasets
Combine Results Using PROC MIANALYZE

Figure 2. The Multiple Imputation Process using SAS/STAT® Procedures



V_1	V_2	V_3	V_4
X	X	X	.
X	X	X	.
X	X	.	.
X	.	.	.

(a) monotone pattern

V_1	V_2	V_3	V_4
X	.	X	X
X	X	X	.
X	X	.	X
X	X	X	.

(b) arbitrary pattern

Figure 3. Pattern of Missingness

Pattern of Missingness	Type of Imputed Variables	Recommended Imputation Method
Monotone	Continuous	Regression
		Predicted mean matching
		Propensity score
Monotone	Classification (Ordinal)	Logistic regression
Monotone	Classification (Nominal)	Discriminant function method
Arbitrary	Continuous	MCMC full-data imputation
		MCMC monotone-data imputation

Table 1. Imputation Methods Available in PROC MI

Table 1 shows that when the missing pattern is monotone, PROC MI offers flexibility in the choice of algorithm to use. When the missing pattern is arbitrary, the MCMC method can impute missing values for continuous variables. However, to date, there are no specific recommendations about multiple imputation methods for categorical/classification variables (i.e., binary, nominal or ordinal) with an arbitrary missingness pattern in SAS. Therefore, the objective of this paper is to demonstrate a SAS/IML® Studio program that can be implemented when the pattern of missingness for a categorical variable is arbitrary.

SAS/IML® STUDIO MULTIPLE IMPUTATION PROCESS

SAS/IML® Studio provides a flexible programming environment in which SAS/STAT or SAS/IML analyses can be run interactively. SAS/IML is a matrix language which can implement operations on rows, columns and matrices. SAS/IML® Studio can solve complicated problems and is versatile for data exploration and model building.

To execute the proposed multiple imputation method in SAS/IML® Studio, an imputation model mapping the observed values to missing values is built based on the part of the dataset that is not missing. The process is shown in Figure 4.

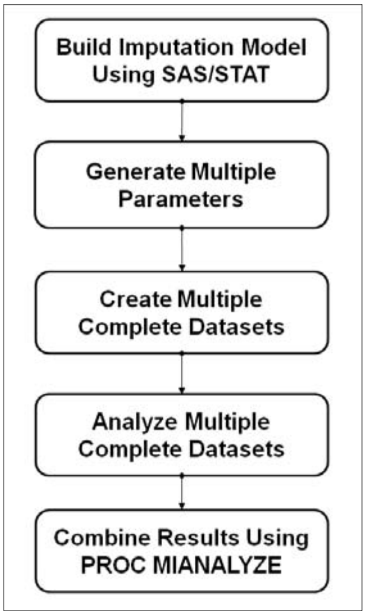

Build Imputation Model Using SAS/STAT
Generate Multiple Parameters
Create Multiple Complete Datasets
Analyze Multiple Complete Datasets
Combine Results Using PROC MIANALYZE

Figure 4. Schematic Diagram of a Flexible Multiple Imputation Process

DATA PREPARATION

SAS is a procedural programming language and SAS procedures run on the SAS server and read data from a SAS data set into a library. The SAS/IML® Studio supports the IMLPlus programming language, which runs on the client and keeps its data in memory on the client. In SAS/IML® Studio, DataObject manages an in-memory version of the data by providing methods to query, retrieve, and manipulate the data. DataObject has a lot of flexibility to be created from different environments such as a SAS data set, Microsoft Excel file or SAS/IML matrix. Programs that call SAS procedures require the transfer of data between an in-memory version of the data and SAS data sets on the server. To create a server data set by copying from a DataObject, analyst can use the WriteToServerDataSet or WriteVarsToServerDataset functions of the DataObject. To create a server data set from IML matrices, analysts can use CREATE and APPEND statements.

BUILD AN IMPUTATION MODEL

The choice of an imputation model is based on data exploration and an understanding of the relationship between the observed and missing values. For simplicity, we focus on the case where we impute a single variable's missing data using multiple other observed variables in the dataset. In the model, the variable of interest (i.e., the variable for which the data are incomplete) is specified as the dependent variable, and characteristics thought to be related to the values of the dependent variable are used as explanatory (i.e., independent) variables. If the variable of interest is categorical, the dependent variable of the imputation model is also categorical. Using this flexible multiple imputation method, a procedure appropriate for the analysis of categorical outcomes in SAS can be used to construct the imputation model. For example, the logistic regression model for binary, ordinal, or nominal data can be conducted using PROC LOGISTIC, PROC SURVEYLOGISTIC, PROC GENMOD or PROC MCMC; the latter is appropriate for a Bayesian analysis. Other procedures in SAS for categorical data include PROC CATMOD, PROC PROBIT, PROC GLIMMIX, PROC CORRESP, PROC PRINQUAL and PROC TRANSREG. After fitting the imputation model, an output SAS data set is created that contains the model parameter estimates and estimated covariances. For most

SAS procedures, this data set can be created using the OUTEST statement. For PROC MCMC, the data set is created using the OUTPOST statement. If there are several variables, either continuous or categorical, that are missing this method can be applied for each variable iteratively.

GENERATE MULTIPLE PARAMETERS

In this step, the estimated parameters of the fitted imputation model and their covariance matrix are specified as the mean vector and covariance matrix, respectively, for a multivariate Gaussian distribution (Wang & Robins, 1998); it is assumed that the parameters will asymptotically follow this multivariate distribution. Estimates of the coefficients are randomly drawn from this distribution for *m* imputations.

We developed the function module 'betagen' using SAS/IML. It generates multiple values of coefficients estimates from the multivariate normal distribution. The function is called with the estimated parameters (beta) and their covariance matrix (cov) as inputs. The first component (tx) is the output. The parameters nimpute represents specified number of imputations and the seed is the specified seed for random generation function.

```
start betagen(tx,beta,cov,nimpute,seed);
       v=nrow(cov);   /*v is the number of variables*/
       do i=1 to v;   /* Ensure the covariance matrix is symmetric*/
              do j=1 to v;
                     if cov[j,i]^=cov[i,j] then cov[j,i]=cov[i,j];
              end;
       end;
       l=t(root(cov)); /*Decompose the covariance matrix*/
       z=normal(j(v,nimpute,seed)); /*Generate multivariate standard normal variable*/
       x=l*z;   /*Multiply the standard normal variable with the root of covariance*/
       x=repeat(beta,1,nimpute)+x;  /*Obtain the multiple random samples of
                                     parameter estimates*/
       tx=t(x);
finish;
```

CREATE MULTIPLE COMPLETE DATASETS

The multiple complete datasets are generated by replacing missing values with plausible imputed values. The plausible values are obtained by putting the multiple parameters into the imputation model in SAS/IML.

ANALYZE MULTIPLE COMPLETE DATASETS

The multiple complete datasets are then analyzed by using existing procedures for complete data and the results are combined. The user may analyze the data by virtually any technique that would be appropriate if the data were complete (Schafer, 1999). It is assumed that with complete data, tests and confidence intervals based on the normal approximation are appropriate. The complete data sets are used to obtain the parameter estimates of interest. Existing SAS procedures can be applied for most parameter of interest including means, regression coefficients, and correlation coefficient. For example, UNIVARIATE and CORR procedures can compute the mean and covariance matrices from imputed datasets using OUT statement. Some regression procedures, such as REG and LOGISTIC, create an EST type dataset that contains both the parameter estimates for the regression coefficients and their associated covariance matrix. For MIXED and GENMOD procedures, ODS OUTPUT statement saves parameter estimates in a dataset and the associated covariance matrix in a separate dataset. The CORR procedure can be used to compute the correlation coefficients as well. However, for a given parameter of interest, it is not always possible to compute the estimate and associated covariance matrix directly from a SAS procedure, for example the ratios of variable means. In such a situation, the analyst can use SAS/IML to develop the algorithm for estimates and covariance matrices.

COMBINE RESULTS USING PROC MIANALYZE

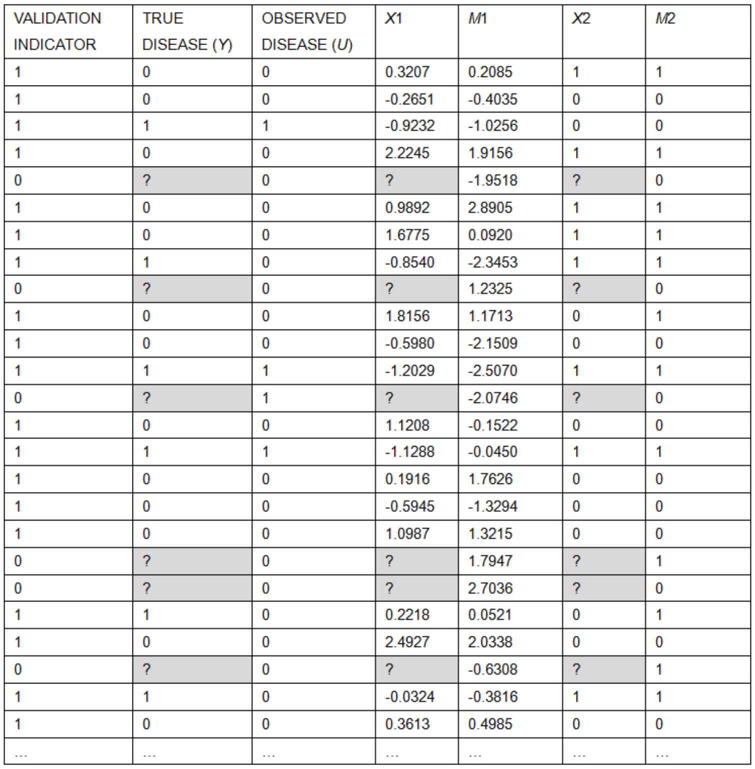
The results of analyses are combined using PROC MIANALYZE to derive valid inferences. The MIANALYZE procedure reads parameter estimates and associated standard errors or covariance matrices that are computed by the standard statistical procedure for each complete data set. No matter which analysis method is used, the process of combining results from different imputed datasets is essentially the same and results in valid statistical inferences that reflect the uncertainty in the data due to missing values.

AN EXAMPLE: MULTIPLE IMPUTATION OF MISCLASSIFIED DISEASE DIAGNOSES IN ADMINISTRATIVE HEALTH DATA

Population-based administrative health databases (AHDs), including hospital records and physician claims, are widely used for chronic disease research and surveillance because they contain diagnosis codes that are recorded in a standardized method using the International Classification of Diseases (ICD) system developed by the World Health Organization (O'Malley, Cook, & Price, 2005). The presence or absence of a diagnosis code in AHDs is used to ascertain disease status (i.e., disease presence/absence). Ascertained disease status can be used in a number of ways, including the estimation of disease prevalence and incidence. However, the accuracy of diagnoses in AHDs is questionable (Tu, Campbell, & Chen, 2007). Validation studies, in which AHDs are linked, via a unique personal identifier, to a 'gold standard' data source in which the true disease status is known, have been used to assess the accuracy of diagnoses recorded in AHDs. In this example, our purpose is to accurately estimate disease prevalence using multiple imputation methods for the problem of misclassification error in disease diagnosis codes.

We used a simulated dataset to illustrate the multiple imputation methods for the case in which the model for disease status contains two explanatory variables. True disease status is observed in a validation dataset that is a subset of the entire dataset. Variables such as the presence of comorbid (i.e., related) health conditions, demographics, and disease treatment, are typical predictors of true disease status that could be used to build the imputation model to predict the probability of have the diagnosed disease. It is important to note that disease predictors in the entire dataset could be error prone in the entire dataset, but they are assumed to be measured without error in the validation dataset.

In our example, a binary indicator variable is used to denote whether an individual is or is not present in the validation dataset (i.e., presence in the validation data set is coded as 1; absence is coded as 0). Figure 5 provides an illustration of the dataset.

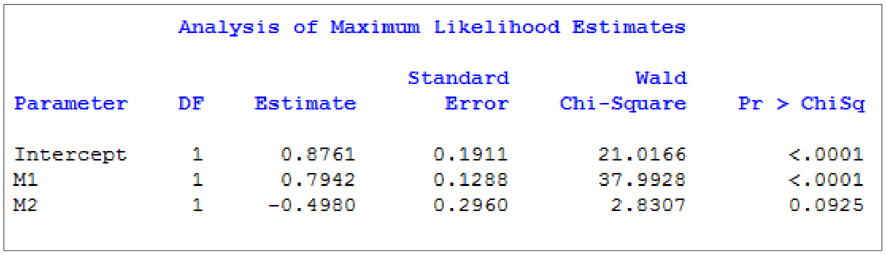


VALIDATION INDICATOR	TRUE DISEASE (*Y*)	OBSERVED DISEASE (*U*)	*X*1	*M*1	*X*2	*M*2
1	0	0	0.3207	0.2085	1	1
1	0	0	-0.2651	-0.4035	0	0
1	1	1	-0.9232	-1.0256	0	0
1	0	0	2.2245	1.9156	1	1
0	?	0	?	-1.9518	?	0
1	0	0	0.9892	2.8905	1	1
1	0	0	1.6775	0.0920	1	1
1	1	0	-0.8540	-2.3453	1	1
0	?	0	?	1.2325	?	0
1	0	0	1.8156	1.1713	0	1
1	0	0	-0.5980	-2.1509	0	0
1	1	1	-1.2029	-2.5070	1	1
0	?	1	?	-2.0746	?	0
1	0	0	1.1208	-0.1522	0	0
1	1	1	-1.1288	-0.0450	1	1
1	0	0	0.1916	1.7626	0	0
1	0	0	-0.5945	-1.3294	0	0
1	0	0	1.0987	1.3215	0	0
0	?	0	?	1.7947	?	1
0	?	0	?	2.7036	?	0
1	1	0	0.2218	0.0521	0	1
1	0	0	2.4927	2.0338	0	0
0	?	0	?	-0.6308	?	1
1	1	0	-0.0324	-0.3816	1	1
1	0	0	0.3613	0.4985	0	0
…	…	…	…	…	…	…

Figure 5. Illustration of a dataset with measurement error in selected variables

To illustrate the proposed multiple imputation method, the imputation process for the true disease status (*Y*) is depicted. The entire dataset is saved in a temporary library (e.g., Work.org_data). To construct a logistic regression model to generate regression parameter estimates in SAS/IML® Studio, the validation dataset (e.g., Work.Validation) is read from the temporary library and analyzed using PROC LOGISTIC. The observed values of disease predictors (*M*1 and *M*2) are used to predict true disease status (*Y*). Note that *M*1 and *M*2 themselves may be prone to measurement error. PROC LOGISTIC produces the estimated parameters of the observed disease predictors *M*1 and *M*2 for the predictive model of true disease status (Figure 6).

```
OriginalData="Work.org_data";

Val= "Work.Validation";

submit Val;

proc logistic data=&Val outest=EST covout;

       model y=m1 m2;

run;

endsubmit;
```



```
                  Analysis of Maximum Likelihood Estimates

                                   Standard          Wald
Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

Intercept     1      0.8761      0.1911       21.0166        <.0001
M1            1      0.7942      0.1288       37.9928        <.0001
M2            1     -0.4980      0.2960        2.8307        0.0925
```

Figure 6. Output from PROC LOGISTIC

The estimated parameter vector and covariance matrix are saved using the OUTEST statement. They are used to generate plausible values of the parameter estimates. In order to apply the 'betegen' function, the estimated parameter vector and covariance matrix are read into a SAS/IML matrix.

```
declare DataObject dobj;

dobj=DataObject.CreateFromServerDataSet("Work.Est");

postn=dobj.GetNumObs();

dobj.GetVarData("Intercept",Intercept);

dobj.GetVarData("M1",M1);

dobj.GetVarData("M2",M2);

EstMatrix=Intercept||M1||M2;

param=EstMatrix[1,];

param_vec=param`;

param_cov=EstMatrix[2:4,];

call betagen(Param_M,param_vec,param_cov,10,6385);
```

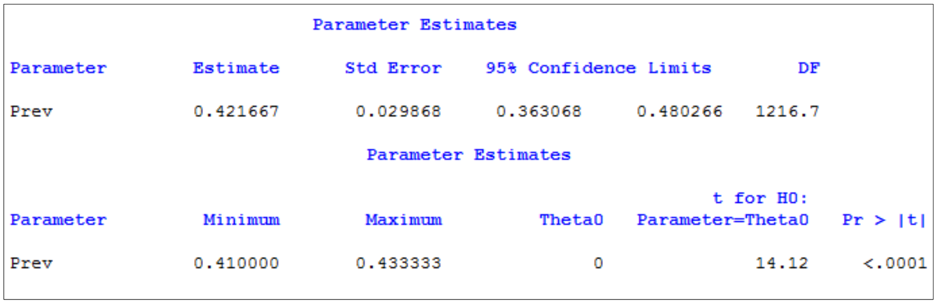
The first statement declares `dobj` to be an IMLPlus variable that refers to a DataObject. The CreateFromServerDataSet statement creates and populates a DataObject from a dataset stored in a SAS library. Each variable is copied into SAS/IML matrices and manipulated into the format for 'betagen' module (e.g., param_vec and param_cov). The betagen function is called with the param_vec and param_cov inputs; multiple estimates of parameters are generated and put into the Param_M matrix. After obtaining the matrix of multiple parameter estimates, we impute the missing true disease status for each imputation by replacing the parameters of the imputation model with each vector of coefficients. The imputed values for missing true disease status are saved into ImputeData matrix by the number of imputations. In this example, we use 10 imputations. The code is:

```
do k=1 to 10;

       prob=j(ntot,1,.);
       fill_vec=j(ntot,1,.);
       do i=1 to ntot;
              if Param_M[k,1]+Param_M[k,2]*alldata[i,5]+Param_M[k,2]*alldata[i,7]>0
then p=1/(1+exp(-(Param_M[k,1]+Param_M[k,2]*alldata[i,5]+Param_M[k,2]*alldata[i,7])));
              else
p=exp(Param_M[k,1]+Param_M[k,2]*alldata[i,5]+Param_M[k,2]*alldata[i,7])/(1+exp(Param_M
[k,1]+Param_M[k,2]*alldata[i,5]+Param_M[k,2]*alldata[i,7]));
              prob[i]=p;
              if alldata[i,1]=0 then do;
                     call randseed(9047);
                     call randgen(fill,'bernoulli',prob[i]);
                     fill_vec[i]=fill;
              end;
              else if alldata[i,1]=1 then fill_vec[i]=alldata[i,2];
       end;
       yimpute[,k]=fill_vec;
end;
IN=10*ntot;
ImputeData=j(IN,2,.);
do k=1 to 10;
       do i=1 to ntot;
              Iy=yimpute[i,k];
              IDi=(k-1)*ntot+i;
              ImputeData[IDi,1]=k;
              ImputeData[IDi,2]=Iy;
       end;
end;
```

After the ImputeData matrix is transferred to a SAS dataset, disease prevalence is estimated by the UNIVARIATE procedure. The estimate and covariance of the UNIVARIATE procedure is saved as an output data set named outuni. Then PROC MIANALYZE is applied to combine the results from the output dataset.

```
Create Work.ImputeData from ImputeData [colname={_Imputation_ ImputeY}];
append from ImputeData;
close Work.ImputeData;
submit;
proc univariate data=ImputeData noprint;
       var ImputeY;
       output out=outuni mean=Prev stderr=SEPrev;
       by _Imputation_;
run;
proc mianalyze data=outuni;
       modeleffects Prev;
       stderr SEPrev;
```

```
run;

endsubmit;
```



```
                                    Parameter Estimates

Parameter          Estimate       Std Error     95% Confidence Limits          DF

Prev               0.421667        0.029868      0.363068      0.480266    1216.7

                                        Parameter Estimates

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

Prev               0.410000        0.433333               0               14.12      <.0001
```

Figure 7. Output from PROC MIANALYZE

In the simulated data, the prevalence estimate based on true disease status is 0.38, while the prevalence estimate based on observed disease status is 0.22, which indicates that the observed value severely underestimates the true prevalence. The prevalence estimate based on the multiple imputation method, which is 0.42, has smaller bias and the confidence interval includes the true prevalence (Figure 7). Therefore, the imputed prevalence of multiple imputation has better accuracy and precision than prevalence based on the observed and misclassified measure of disease status.

If the SAS analyst wants to impute the missing true values of the disease predictor variables (i.e., *X*1 and *X*2), imputation models reflecting the relationships between true and observed values (i.e., *M*1 and *M*2) can be constructed separately. After iterations of flexible multiple imputation method steps several times for each of the variables with missing data, the incomplete dataset including categorical variables with arbitrary missing data pattern can be fully filled by plausible values based on the imputation model.

CONCLUSION

SAS/IML® Studio provides an integrated development environment that enables users to combine the flexibility of the SAS/IML matrix language, analytical power of SAS/STAT procedures and data manipulation capabilities of DataObject class to apply multiple imputation methods to problems of incomplete data for categorical variables when the missingness mechanism is arbitrary. To implement the multiple imputation method using SAS/IML® Studio, the betagen function module was developed to generate multiple values based on an imputation model generated from observed data. The numeric example demonstrated how to apply this multiple imputation method to the problem of misclassified disease diagnoses. Breaking the multiple imputation process into a series of linked steps improves the flexibility of the process for a variety of models. Moreover, there are other situations for which PROC MI and MIANALYZE may not be appropriate choices. For example, a set of explanatory variables may be collinear or the analyst needs to impute the mean plus noise for each missing observation in an intercept-only regression model (Paulin, Tsai, & Grance, 2004). SAS/IML® Studio is a platform that the analyst can use to manage all steps of the multiple imputation method.

REFERENCES

Cole, S. R., Chu, H., & Greenland, S. (2006). Multiple imputation for measurement-error correction. *International Journal of Epidemiology, 35*, 1074-1081.

Mayer, B., Muche, R., Hohl, K. (2012). Software for the handling and imputation of missing data: An overview. *Journal of Clinical Trials, 2*, 103-111.

O’Malley, K.J., Cook, K.F., & Price, M.D. et al. (2005). Measuring diagnoses: ICD code accuracy. *Health Services Research*, 40, 1620-1639.

Paulin, G., Tsai, S., & Grance, M. et al. (2004). Model-based multiple impuation. *Proceedings of the Twenty-Ninth Annual SAS® Users Group International Conference*. Cary, NC: SAS Institute Inc.

Refaat, M. (2007). *Data Preparation for Data Mining Using SAS*. San Francisco: Morgan Kaufmann.

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

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

Schafer, J.L. (1999). Multiple imputation: a primer. *Statistical Methods in Medical Research*, 8, 3-15.

Tu, K., Campbell, N.R., & Chen Z.L. et al. (2007). Accuracy of administrative databases in identifying patients with hypertension. *Open Medicine,* 1, e18-26.

Wang, N. & Robins, J.M. (1998). Large sample theory for parametric multiple imputation procedures. *Biometrika*, 85, 935-948.

RECOMMENDED READING

- *SAS/STAT® 9.2 User's Guide*
- *SAS/IML® Studio 3.4 for SAS/STAT® Users*

CONTACT INFORMATION

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

Xue Yao
Department of Community Health Sciences, University of Manitoba
S113-750 Bannatyne Avenue
Winnipeg, MB CANADA R3E 0W3
Work Phone: 204-480-1371
E-mail: Xue.Yao@med.umanitoba.ca

Lisa Lix
Department of Community Health Sciences, University of Manitoba
S113-750 Bannatyne Avenue
Winnipeg, MB CANADA R3E 0W3
Work Phone: 204-789-3573
E-mail: lisa.lix@med.umanitoba.ca