Paper 210-29

Model-Based Multiple Imputation

Geoffrey Paulin, Shirley Tsai, Melissa Grance, Bureau of Labor Statistics, Washington DC

ABSTRACT

Nonresponse to income questions is a long-standing problem in the household surveys. In order to improve the data quality, the SAS Institute developed a test version of an imputation procedure in V.8, and has since made significant improvement. However, in some cases, analysts may need more information or statistics from the imputation process to help them with their analyses. The purpose of this paper is to demonstrate how to use SAS/STAT and SAS/IML to build model-based multiple imputation macros such that analysts can streamline the analytical process without performing these tasks step by step. The underlying theoretical framework of this model-based multiple imputation is derived from Rubin's (1987) repeated-imputation method.

I. INTRODUCTION

Nonresponse to income questions is a long-standing problem in household surveys. The more detailed the information requested, the more difficult it can be to collect. Some respondents may be reluctant to provide information on income because they consider it to be a private matter. Some respondents simply may not know precise values for each source of income the family may receive. Nevertheless, income data are of critical importance to researchers and policy makers in any number of fields. Therefore, the lack of statistically reliable income data represents a loss of valuable information.

The Bureau of Labor Statistics sponsors the Consumer Expenditure Survey, which is the most detailed source of household expenditures collected by the Federal government. In addition to obtaining information on expenditures, the survey collects detailed information on demographics and income. Producing statistically reliable income data is of crucial importance to users of this survey, because of the important role income plays in a household's expenditure decisions. Currently, the Bureau of Labor Statistics is working to provide statistically reliable income data using a method known as multiple imputation to fill in the blanks where they exist. In brief, results from regression models are produced, and used to obtain several estimates for each missing income value. Detailed information on the underlying theoretical framework of this model-based multiple imputation is derived from Rubin's (1987) repeated-imputation method. The purpose of this paper is to demonstrate how to use SAS/STAT and SAS/IML to build model-based multiple imputation macros such that analysts can streamline the analytical process without performing these tasks step by step. This paper introduces the analytical components of the model-based multiple imputation macros. The topics are organized in six sections. Section I is a brief introduction to our income imputation project. Section II is a general introduction to the source data. Section III describes the reasons for creating an alternative application instead of using the experimental SAS PROC MI in version 8.2. Section IV introduces the Model-Based Multiple Imputation. Section V provides the sample SAS macro procedures to perform these tasks. Section VI is the summary.

II. DATA DESCRIPTION

To perform multiple imputations, a standard regression is run. That is, reported values of the variable in question (income in this case) are used as the dependent variable, and characteristics thought to be related to the dependent variable (such as age or occupation in this case) are used as independent variables. Random noise is added to the coefficients of these independent variables to produce a set of equations used to predict values for missing cases. When the predicted value is obtained, an additional random noise is added to the predicted value, to ensure that two families with identical characteristics will not receive the same imputed value. To further preserve the variance (including the uncertainty associated with the imputation process—that is, by the very fact that data are imputed, the actual value is not known with certainty) the process is repeated several times, with new random noises added to each coefficient. As noted, a detailed description of the process is available in Rubin (1987).

In the current system different regression models are run on families with different characteristics. For example, single parent families and married couples without children are included in separate regression models. Currently, the project is working on imputing incomes collected in the survey for the family as a whole. Sources include pension, interest earnings, unemployment compensation, and welfare income. Based on the results of the regressions, missing income values for families with similar characteristics (e.g., single parents and married couples without children) are filled in. Despite the differences among the incomes and family groups, generalized macros have been created to improve the computing efficiency and future maintenance.

Our source data consist of twenty consecutive quarters of Consumer Expenditure Survey data, and the appropriate values of the Consumer Price Index (CPI). To reduce the effect of inflation on the imputations, the CPI data are used to adjust the nominal incomes of the previous nineteen quarters based upon the price level in the current quarter.

III. WHY NOT USE PROC MI – OUR USER'S CONCERN

Starting with the release of SAS V.8 the multiple imputation procedure has been "built in" as a test version to SAS under the name "PROC MI." Given the availability of these procedures, the reader may well wonder, "Why not just use them?" Indeed, the team initially considered doing this. However, experimentation and data necessities demonstrated that these procedures were not suited for the task at hand. However, using other SAS procedures, the team continues to build a system to impute income upon its completion. Nevertheless, some of the reasons for not using PROC MI include:

LIMITED DIAGNOSTIC CAPABILITIES

The user selects the variable to be imputed and the variables to use as predictors. The PROC MI procedure then produces m imputations of the missing variable, where m is specified by the user. The problem is that beyond this, the user cannot decompose the process into its component parts to find out exactly what happened at each step of the way. For example, it could happen that too many extreme values are imputed. However, there is no way to ascertain whether this was due to something inherent in the regression (such as extraordinary parameter estimates), a problem with the noise (for example, a particularly large number drawn from the normal distribution), or some other problem.

THE PROCEDURE GIVES NO WARNING WHEN THE MODEL IS "NOT FULL RANK."

Multicollinearity is sometimes a problem in producing regression results, especially when the number of observations with non-missing values of the dependent variable is small. With binary independent variables, there are times when everyone in the sample has the same characteristic or no one in the sample has the characteristic. For example, in a small sample, it is entirely possible that everyone interviewed resided in the same region of the country—e.g., the Midwest—or that no one in the sample resided there. The code for a binary variable, MIDWEST, would be "1" for all observations in the first case and "0" for all observations in the second case. Additionally, although some variables are mutually exclusive, some relationships are not evident before running the model. For example, when imputing interest income for single men, it is obvious that no binary variables for sex of respondent or type of consumer unit (e.g., husband and wife only, or single parent) are necessary to include in the model. However, it is possible that all the single men in the model who lived in Midwestern, urban areas were retired, and no one outside the urban Midwest was retired. In this case, the variables MIDWEST, URBAN and RETIRED are perfectly collinear. (Knowing that the respondent is retired provides no additional information once it is known that he is living in an urban area of the Midwest.) The fact that in the given sample all Midwestern urbanites are retired is not something that can be known before analyzing the data set. The larger the number of independent variables, the more chance there is of observing unexpected perfect multicollinearity.

When PROC REG is used, the system provides an error message saying, "Model is not full rank," and then describing what linear combinations of variables were found. However, PROC MI continues and produces results. The problem is the black box issue again. Consider a case where two perfectly collinear variables (e.g., MALE and FEMALE) were included in the model for some reason. Is SAS ignoring one of the variables, or is it somehow obtaining imputed values using both of these? Since there is not even a warning to the programmer, this remains

LACK OF AUTOMATED REMEDY WHEN SINGULAR MATRICES OCCUR (K > N CASES).

In the multiple imputation literature (including the SAS manual) it is common to urge the user to include as many independent variables as possible in the model. However, there are cases (again when the sample size is small) where the number of potential predictors (or independent variables), k, clearly exceeds the number of observations, n, for use in the imputation (or dependent variables). In these cases, SAS produces the following message: "ERROR: Not enough observations to fit regression models with METHOD=REGRESSION." In production, the system would stop and the user would have to constantly reevaluate and select which variables to include in the model. In the system under development, this problem has been solved using the PROC CORR procedure, sorting the results in descending order of correlation, and selecting the top n-1 independent variables from this list. The system under development automatically runs this procedure and makes the selection when k > n, and no intervention on the programmer's part is necessary.

THE "INTERCEPT ONLY" CASE.

There are times when the number of observations with non-missing values for the dependent variable is too small to invoke the Central Limit Theorem (i.e., 30 or fewer observations). In this case, it is not likely that any independent variables will be of great use in imputing the dependent variable. The team's solution is essentially to impute the mean plus noise for each missing observation. PROC MI has no mechanism to do this. However, in the system under development, an "intercept only" model is run. That is, the model statement is:

```
Proc reg;
Model = ;
```

This produces a result in which the intercept equals the data mean, and noise is added to it as described by Rubin (1987).

PROC MI IS EXPERIMENTAL IN V.8

Although PROC MI is no longer experimental in version 9, the latest version in our office is 8.2. During testing we found that the transform statement for the Box-Cox method did not work properly. SAS technical support gave us a work-around for this problem. It is very important that our imputations come from production code. The project could not afford to wait for our office to obtain the latest software.

IV. MODEL-BASED MULTIPLE IMPUTATION

Only the income variables yield missing values in the source data set. Since the reported income values yield a wide range, a normal score transformation is applied to these income variables. This transformation identifies the cumulative probabilities corresponding to the income variable and a standard normal random variable Z through a quantile mapping function. The transformed Z-value is utilized as a dependent variable in our regression analysis later.

Each family group may have a different number of reported consumer units. When the number of observations is relatively small, an intercept-only regression model is used; otherwise the normal regression model is used in the regression process. A random adjustment is applied to both regression parameter estimates and the unique consumer unit. These adjusted parameters will be incorporated into the imputation method later. By multiplying the analytical variables for both response and non-response family data, we obtain the imputed income values. The imputed income values are derived from this imputation method for repeated measures to minimize the bias in an effective and efficient manner. Rubin's F-test is applied as an imputation performance test.

The general mathematical formula of the model-based imputation at the consumer unit level is demonstrated as follows:

$$\tilde{Z}_{pqr} = \left\{ \sum_{n=0}^{N} \left[\hat{b}_0 + \left(V_{nj}^{'} \sqrt{\frac{(n_j - j)_p}{g_{pq}}} * RANNOR_{pq}(B_q) \right) \right] * X_{npqr} \right\} + \left(RMSE * \sqrt{\frac{(n_j - j)_p}{g_{pq}}} * RANNOR_{pqr}(\varepsilon) \right).$$

where $\tilde{Z}_{pqr}$ is the imputed value for consumer unit r in the q^{th} imputation, n_j is the number of non-missing observations, j is the number of regressors including the intercept, g is the random variable drawn from a chi-square distribution where degrees of freedom equals $n_j - 1$, p is the family group number, q is the imputation number, r is the consumer unit number, V' is the Cholesky square root of the variance-covariance matrix, B is the matrix of parameter based random numbers which are generated for each imputation, ε is the set of random numbers which are generated for each imputed consumer unit.

The intercept-only model is a reduced form of the general case. Because all income values for the group undergoing imputation are regressed against the intercept only, the coefficient on the intercept will equal the mean of those reporting the source. Noise will then be added to this parameter estimate as we did for the normal regression case. The mathematical presentation for the intercept-only model is presented as follows:

$$\tilde{Z}_{p1r} = \left(\hat{b}_0 + \left[V_{00}^{'} \sqrt{\frac{(m-1)}{g_{p1}}} * RANNOR(B_{0p1}) \right] * \frac{1}{n_j} \right) + (V_{00}^{\prime}) \sqrt{\frac{(m-1)}{g_{pq}}} * RANNOR(\varepsilon_{p1r}).$$

where V' is the Cholesky square root of the variance-covariance matrix (see below), n_j is the number of non-missing observations, g is a random variable drawn from a chi-square distribution with degree of freedom and is equal to $n_j - 1$, p is the model number, q is the imputation number, and m is number of imputation which is equal to 5 in our case. The symbol V' can further be described as:

$$V' \equiv \overset{Cholesky}{\sqrt{\sigma^2 (X'X)^{-1}}} = \sigma \overset{Cholesky}{\sqrt{(X'X)^{-1}}}; \because \overset{Cholesky}{\sqrt{(X'X)^{-1}}} = \sqrt{\frac{1}{n_j}} \therefore V' = \sigma * \sqrt{\frac{1}{n_j}} = \frac{\sigma}{\sqrt{n_j}}.$$

Note that σ is a scalar, equal to root mean square error (RMSE), shown in the SAS regression output.

For each family group, we impute the income five times and obtain the average mean and variance from the imputations for both intercept-only and normal regression models. The imputation variances for the multiple imputation estimates of the mean and variance are the basis for obtaining the inferences such as degrees of freedom, total variances, within imputation variances, between-imputation variances, fraction of information, and so on. The p-values from Rubin's F-test help researchers to analyze whether the imputations properly reflect the uncertainty and preserve important aspects of the data distributions. The results from the relative efficiency of point estimates help the researchers to determine the degree of efficiency of the five repeated-imputation estimator relative to the fully

efficiency infinite-m repeated-imputation in terms of the units of standard errors. These inferences are developed in Rubin (1987). A similar description can be found at Yuan (2000), p.5. The source codes corresponding to the F-test and relative efficiency are provided in the next section.

V. SAMPLE CODES:

A. Regression:

```
/* ************************************************************************/
/* JOB NAME : REGRESS.SAS                                                 */
/* FUNCTION : Performs regression. Get required statistics for imputation. */
/* PARAMETERS :                                                           */
/* INPUT = SAS data set that contains dependent and independent variables. */
/* OUTPUT = Name of the SAS data set that will contain the predicted value.*/
/* IVARS = List of independent variables (if no variables are listed,     */
/*         then use intercept only).                                      */
/* DEPVAR = Dependent variable.                                           */
/* ALPHA = Specifies the parameter of significance level.                 */
/* GROUPNUM = Group number.                                               */
/* OUTCOV = Output data set that contains the Covariance Matrix.          */
/* OUTPARM = Output data set that contains the parameter estimates as a   */
/*           single column vector.                                        */
/* SELECTION = Selection type; Only FORWARD, BACKWARD, or STEPWISE.      */
/************************************************************************/

%MACRO REGRESS (INPUT =, OUTPUT =, DEPVAR =, IVARS =, ALPHA =, GROUPNUM =,
                OUTCOV =, OUTPARM =, SELECTION =);

/************************************************************************/
/* PERFORM REGRESSION. GET COVARIANCE MATRIX, RMSE, PARAMETER ESTIMATES   */
/* FOR IMPUTATIONS.                                                      */
/************************************************************************/

  %IF (&IVARS NE ) %THEN %DO;
    TITLE "&SELECTION REGRESSION";
    %END;
  %ELSE %DO;
    TITLE 'INTERCEPT ONLY REGRESSION';
    %END;

  PROC REG DATA = &INPUT OUTEST=EST COVOUT;
    GROUP&GROUPNUM : MODEL &DEPVAR = &IVARS
    %IF (&IVARS NE ) %THEN %DO;
        / SELECTION = &SELECTION
      %IF (&SELECTION NE FORWARD) %THEN %DO;
        SLS = &ALPHA
      %END;
        %ELSE %DO;
        SLE = &ALPHA
      %END;
    %END;;
    OUTPUT OUT = &OUTPUT;
  QUIT;
  TITLE;

/************************************************************************/
/* PUT PARAMETER ESTIMATES AND COVARIANCE MATRIX INTO DIFFERENT DATA SETS. */
/* PUT RMSE INTO A GLOBAL MACRO VAR.                                      */
/************************************************************************/

 DATA PARMS &OUTCOV;
 SET EST;
   IF (_TYPE_ = 'PARMS') THEN OUTPUT PARMS;
     ELSE IF (_TYPE_ = 'COV') THEN OUTPUT &OUTCOV;
   IF (_N_ = 1) THEN DO;
```

```
     CALL SYMPUT('GLRMSE', PUT(_RMSE_, BEST.));
   END;
   KEEP INTERCEPT &IVARS;
 RUN;

 %PUT GLRMSE = &GLRMSE;

/*************************************************************************/
/* TRANSPOSE PARAMETER ESTIMATES INTO A SINGLE ROW VECTOR.               */
/*************************************************************************/

  PROC TRANSPOSE DATA = PARMS OUT = &OUTPARM (WHERE = (COL1 ^= .));
  VAR INTERCEPT &IVARS;
  RUN;

  DATA &OUTPARM (RENAME = (COL1 = PAR1));
  SET &OUTPARM;
  RUN;

/*************************************************************************/
/* PUT REDUCED PARAMETER LIST (WITH INTERCEPT) INTO A MACRO VARIABLE.    */
/*************************************************************************/

  PROC SQL NOPRINT;
  SELECT _NAME_ INTO : GLPARMLIST SEPARATED BY ' '
  FROM &OUTPARM;
  QUIT;

  %PUT GLPARMLIST = &GLPARMLIST;

/*************************************************************************/
/*  MAKE REDUCED MACRO LIST WITH ONLY                                    */
/*  INDEPENDENT VARS - NO INTERCEPT                                      */
/*************************************************************************/

  %LET GLINDVARS = %REMOVSTR(LIST =
   &GLPARMLIST, STR = INTERCEPT);
  %PUT GLINDVARS = &GLINDVARS;

/*************************************************************************/
/* DELETE MISSING INDEP. VARS FROM OUTCOV                                */
/*************************************************************************/

  DATA &OUTCOV (KEEP = &GLPARMLIST);
    SET &OUTCOV;
  RUN;

%MEND REGRESS;
```

B. Imputation:

```
PROC IML;
   START IMPUTE(IVARMTRX, PARM1, CHOLESKY, RMSE, CHIMX, RANNOR1, RANNOR2);
     NROW = NROW(IVARMTRX);

    /* &DEGF IS THE DEGREE OF FREEDOM */
     DF_CHI = SQRT(&DEGF / CHIMX)
       %IF (&N <= 30) %THEN %DO;
         * (1 / &N)
       %END;;

     PLABEL = {&GLPARMLIST};
     OBSLABEL = "OBS1" : "OBS&NMOBSIMP";
     NOISE = (CHOLESKY * RANNOR1) * DF_CHI;
```

```
/*************************************************************************/
/* ADD NOISE TO BETA MATRIX. PARM_NOISE IS A M x N MATRIX WHERE M IS ONE   */
/* PLUS NUMBER OF PARAMETERS, N IS NUMBER OF MODEL REGRESSION.             */
/*************************************************************************/

    BNOISE = NOISE + PARM1;

/*************************************************************************/
/* GET IMPUTED ADJUSTED XB MATRIX. IMP_XB IS A P X 1 MATRIX.               */
/*************************************************************************/

    DO I = 1 TO &NMOBSIMP;
      BNORMSE = BNORMSE // (IVARMTRX[I, ] * BNOISE[ , 1]);
    END;

    RNOISE = RANNOR2 * RMSE * DF_CHI; /* CREATE RMSE NOISE */
    IMP_VAL = BNORMSE + RNOISE; /* ADD RMSE NOISE */
    RETURN(IMP_VAL);
    FINISH IMPUTE;

/*************************************************************************/
/* CONCATENATE INTERCEPT COLUMN TO INDEPENDENT VARIABLE MATRIX.            */
/*************************************************************************/

    INTERCEPT =  J(&NMOBSIMP, 1, 1);
    USE IMP_DATA;
    %IF (&GLINDVARS NE  ) %THEN %DO;
      READ ALL VAR {&GLINDVARS} INTO BASEDATA;
      BASEDATA = INTERCEPT || BASEDATA;
      READ ALL VAR {&GLINDVARS} INTO PURIND;
    %END;
    %ELSE %DO;
      BASEDATA = INTERCEPT;
    %END;

    READ ALL VAR {FAMID
      %IF (&GLVTYPE = MEMB) %THEN %DO; /*  &GLVTYPE IS THE DATABASE TYPE  */
          MEMBNO
      %END;
    } INTO FAMNUM;

    CLOSE IMP_DATA;

/*************************************************************************/
/*   CHOLESKY MATRIX                                                     */
/*************************************************************************/

    USE COV;
    READ ALL VAR {&GLPARMLIST} INTO COVMTRX;
    ROOTCOV = ROOT(COVMTRX);
    CLOSE COV;
    PLABEL = {&GLPARMLIST};
    FREE COVMTRX;

/*************************************************************************/
/* PARAMETER ESTIMATION MATRIX                                             */
/*************************************************************************/

    USE PARMTRAN;
    READ ALL VAR {PAR1} INTO BETA;
    CLOSE PARMTRAN;

/*************************************************************************/
```

```
/* RANDOM PARAMETER NOISE MATRIX                                          */
/************************************************************************/

    USE RANDPARM;
    %DO R = 1 %TO &NUMIMP;
      READ ALL VAR {RAND&R} INTO RANDPAR&R;
    %END;
    CLOSE RANDPARM;

/************************************************************************/
/* RANDOM RMSE NOISE MATRIX                                             */
/************************************************************************/

    USE RANDRMSE;
    %DO T = 1 %TO &NUMIMP;
      READ ALL VAR {RAND&T} INTO RANDRMS&T;
    %END;
    CLOSE RANDRMSE;

/************************************************************************/
/* CHI-SQUARE NOISE MATRIX                                              */
/************************************************************************/

    USE RANDCHI;
    %DO V = 1 %TO &NUMIMP;
      READ ALL VAR {G&V} INTO CHIMX&V;
    %END;
    CLOSE RANDCHI;

/************************************************************************/
/* RMSE SCALAR VECTOR                                                   */
/************************************************************************/

    RMSEM = {&GLRMSE};

/************************************************************************/
/* CALL IMPUTE FUNCTION.  &INCVAR IS THE INCOME VARIABLE NAME.          */
/************************************************************************/

    %DO J = 1 %TO &NUMIMP;
      Z&INCVAR&J = IMPUTE(BASEDATA, BETA, ROOTCOV, RMSEM, CHIMX&J,
                          RANDPAR&J, RANDRMS&J);
    %END;

/************************************************************************/
/* CONCATENATE HORIZONTALLY THE COLUMN VECTORS WITH IMPUTED VALUES.      */
/************************************************************************/

    %IF (&NUMIMP = 1) %THEN %DO;
      %IF (&GLVTYPE = FMLY) %THEN %DO;
            ALLZ = FAMID || Z&INCVAR&NUMIMP;
      %END;
      %ELSE %IF (&GLVTYPE = MEMB) %THEN %DO;
        ALLZ = FAMID || MEMBNO || Z&INCVAR&NUMIMP;
      %END;
    %END;
    %ELSE %DO;
      ALLZ = %DO K = 1 %TO %EVAL(&NUMIMP - 1);
      Z&INCVAR&K || %END;
      Z&INCVAR&NUMIMP;
    %END;

    %IF (&GLVTYPE = FMLY) %THEN %DO;
      FAMNAME = {FAMID};
```

```
    %END;
    %ELSE %IF (&GLVTYPE = MEMB) %THEN %DO;
      FAMNAME = {FAMID MEMBNO};
    %END;

    %IF (&N > 30) %THEN %DO;
      INDVAR = {&GLINDVARS};
      IMPUTENM = "Z&INCVAR.1" : "Z&INCVAR&NUMIMP";
      OIMPVARS = FAMNAME || INDVAR || IMPUTENM;
      OUTIMP = FAMNUM || PURIND || ALLZ ;
    %END;
    %ELSE %DO;
      IMPUTENM = "Z&INCVAR.1" : "Z&INCVAR&NUMIMP";
      OIMPVARS = FAMNAME || IMPUTENM;
      OUTIMP = FAMNUM || ALLZ ;
    %END;

    CREATE IMPUTEDS FROM OUTIMP[COLNAME = OIMPVARS];
    APPEND FROM OUTIMP;

  QUIT;
```

C. Inferences:

```
/*************************************************************************/
/* JOB NAME : STATISTICS.SAS                                             */
/* FUNCTION : CALCULATES THE INFERENCES SUCH AS DEGREES OF FREEDOM, TOTAL  */
/*            VARIANCES, WITHIN IMPUTATION VARIANCES, BETWEEN IMPUTATION   */
/*            VARIANCE FRACTION OF INFORMATION MISSING DUE TO NON-RESPONSE,*/
/*            ETC.                                                       */
/* PARAMETERS : INCVAR = THE INCOME VARIABLE (DEPENDENT VARIABLE) THAT WAS */
/*                       IMPUTED.                                        */
/*              NUMIMP = NUMBER OF IMPUTATIONS.                          */
/*              INPUT = INPUT SAS DATA SET.                              */
/*************************************************************************/

%MACRO STATISTICS(INCVAR =, NUMIMP =);


/*************************************************************************/
/* CALCULATE THE RELATIVE EFFICIENCY OF THE MEAN AND VARIANCE            */
/*************************************************************************/

  PROC MEANS DATA = &INPUT MEAN VAR STDERR NOPRINT;
  VAR Z&INCVAR.1 - Z&INCVAR&NUMIMP;
  OUTPUT OUT = MEANVAR MEAN = MEAN1 - MEAN&NUMIMP
         VAR = VARIANCE1 - VARIANCE&NUMIMP
         STDERR = STDERR1 - STDERR&NUMIMP;
  RUN;

  DATA RE;
  SET MEANVAR;
  H_MEAN = 0; /* NULL HYPOTHESIS VALUE */
  H_VARIANCE = 1;

  LABEL H_MEAN = 'NULL HYPOTHESIS OF THE MEAN';
  LABEL H_VARIANCE = 'NULL HYPOTHESIS OF THE VARIANCE';

/*************************************************************************/
/* CALCULATE THE AVERAGE OF THE &NUMIMP COMPLETE-DATA ESTIMATES.         */
/*************************************************************************/

    Q_BAR_MEAN = MEAN(OF MEAN1 - MEAN&NUMIMP);
    Q_BAR_VARIANCE = MEAN(OF VARIANCE1 - VARIANCE&NUMIMP);
    LABEL Q_BAR_MEAN = 'MI ESTIMATE OF THE MEAN';
```

```
    LABEL Q_BAR_VARIANCE = 'MI ESTIMATE OF THE VARIANCE';

    %DO NIMP = 1 %TO &NUMIMP;
      UM&NIMP = STDERR&NIMP ** 2;
      UV&NIMP = (2 / _FREQ_) * (VARIANCE&NIMP ** 2);
      BM&NIMP = (MEAN&NIMP - Q_BAR_MEAN) ** 2;
      BV&NIMP = (VARIANCE&NIMP - Q_BAR_VARIANCE) ** 2;
    %END;

/*************************************************************************/
/* CALCULATE THE AVERAGES OF THE &NUMIMP COMPLETE-DATA VARIANCES FOR THE  */
/* ESTIMATES OF INTEREST.                                                 */
/*************************************************************************/

    U_BAR_MEAN = MEAN(OF UM1 - UM&NUMIMP);
    U_BAR_VARIANCE = MEAN(OF UV1 - UV&NUMIMP);

 LABEL U_BAR_MEAN = 'WITHIN IMPUTATION VARIANCE FOR THE MI ESTIMATE OF THE MEAN';
  LABEL U_BAR_VARIANCE = 'WITHIN IMPUTATION VARIANCE FOR THE MI ESTIMATE OF THE
 VARIANCE';

/*************************************************************************/
/* CALCULATE THE VARIANCE BETWEEN THE &NUMIMP COMPLETE-DATA ESTIMATES FOR */
/* BOTH THE MEAN AND THE VARIANCE                                         */
/*************************************************************************/

    B_MEAN = SUM(OF BM1 - BM&NUMIMP) / (&NUMIMP - 1);
    B_VARIANCE = SUM(OF BV1 - BV&NUMIMP) / (&NUMIMP - 1);

LABEL B_MEAN = 'BETWEEN IMPUTATION VARIANCE FOR THE MI ESTIMATE OF THE MEAN';
 LABEL B_VARIANCE = 'BETWEEN IMPUTATION VARIANCE FOR THE MI ESTIMATE OF THE
 VARIANCE';

/*************************************************************************/
/* CALCULATE THE TOTAL VARIANCES                                          */
/*************************************************************************/

    T_MEAN = U_BAR_MEAN + 1.2 * B_MEAN;
    LABEL T_MEAN = 'TOTAL VARIANCE OF THE MI MEAN ESTIMATE';
    T_VARIANCE = U_BAR_VARIANCE + 1.2 * B_VARIANCE;
    LABEL T_VARIANCE = 'TOTAL VARIANCE OF THE MI VARIANCE ESTIMATE';


/*************************************************************************/
/* CALCULATE THE RELATIVE INCREASE IN VARIANCE DUE TO NON-RESPONSE        */
/*************************************************************************/

    R_MEAN = 1.2 * (B_MEAN / U_BAR_MEAN);
    R_VARIANCE = 1.2 * (B_VARIANCE / U_BAR_VARIANCE);
    LABEL R_MEAN = 'RELATIVE INCREASE IN VARIANCE DUE TO NONRESPONSE FOR THE
    MI ESTIMATE OF THE MEAN';

    LABEL R_VARIANCE = 'RELATIVE INCREASE IN VARIANCE DUE TO NONRESPONSE FOR
    THE MI ESTIMATE OF THE VARIANCE';

/*************************************************************************/
/* CALCULATE THE DEGREES OF FREEDOM WITH EACH ESTIMATE                    */
/*************************************************************************/

    NU_MEAN = (&NUMIMP - 1) * (1 + (1 / R_MEAN)) ** 2;
    NU_VARIANCE = (&NUMIMP - 1) * (1 + (1 / R_VARIANCE)) ** 2;
    LABEL NU_MEAN = 'DF ASSOCIATED WITH THE MI ESTIMATE OF THE MEAN';
    LABEL NU_VARIANCE = 'DF ASSOCIATED WITH MI ESTIMATE OF THE VARIANCE';
```

```
/*************************************************************************/
/* CALCULATE THE FRACTION OF INFORMATION ABOUT THE ESTIMATES WHICH IS    */
/* MISSING DUE TO NON-RESPONSE                                           */
/*************************************************************************/

    GAMMA_MEAN = (R_MEAN + (2 / (NU_MEAN + 3))) / (R_MEAN + 1);
    GAMMA_VARIANCE = (R_VARIANCE + (2 / (NU_VARIANCE + 3)))/(R_VARIANCE + 1);
     LABEL GAMMA_MEAN = 'FRACTION OF INFORMATION ABOUT THE MEAN MISSING
    DUE TO NONRESPONSE';
    LABEL GAMMA_VARIANCE = 'FRACTION OF INFORMATION ABOUT THE VARIANCE
    MISSING DUE TO NONRESPONSE';

/*************************************************************************/
/* CALCULATE DM STATISTIC                                                */
/*************************************************************************/

    DM_MEAN = (Q_BAR_MEAN ** 2) / T_MEAN;
    DM_VARIANCE = ((1 - Q_BAR_VARIANCE) ** 2)/ T_VARIANCE;

/*************************************************************************/
/* CALCULATE RUBIN'S SIGNIFICANCE LEVEL                                  */
/*************************************************************************/

    IF (NU_MEAN <= (2 ** 31 - 1)) THEN DO;
      P_MEAN = 1 - CDF('F', DM_MEAN, 1, NU_MEAN);
    END;
    ELSE DO;
      P_MEAN = 1 - CDF('F', DM_MEAN, 1, 2 ** 31 - 1);
    END;

    LABEL P_MEAN = 'RUBIN''S SIGNIFICANCE LEVEL ASSOCIATED WITH THE NULL VALUE
    OF THE MEAN EQUAL TO 0';

    IF (NU_VARIANCE <= (2 ** 31 - 1)) THEN DO;
      P_VARIANCE = 1 - CDF('F', DM_VARIANCE, 1, NU_VARIANCE);
    END;
    ELSE DO;
      P_VARIANCE = 1 - CDF('F', DM_VARIANCE, 1, 2 ** 31 - 1);
    END;

    LABEL P_VARIANCE = 'RUBIN''S SIGNIFICANCE LEVEL ASSOCIATED WITH
    THE NULL VALUE OF THE VARIANCE EQUAL TO 1';

  /* IDENTIFY WHETHER OR NOT THE DIFFERENCE FROM THE NULL HYPOTHESIS IS    */
  /* SIGNIFICANT AT THE 95 OR 99 PERCENT CONFIDENCE LEVEL                  */

    LENGTH SIG95_MEAN $3 SIG95_VARIANCE $3 SIG99_MEAN $3 SIG99_VARIANCE $3;
    IF (P_MEAN <= 0.05) THEN SIG95_MEAN = 'YES';
      ELSE SIG95_MEAN = 'NO';

    IF (P_VARIANCE <= 0.05) THEN SIG95_VARIANCE = 'YES';
      ELSE SIG95_VARIANCE = 'NO';

    IF (P_MEAN <= 0.01) THEN SIG99_MEAN = 'YES';
      ELSE SIG99_MEAN = 'NO';

    IF (P_VARIANCE <= 0.01) THEN SIG99_VARIANCE = 'YES';
      ELSE SIG99_VARIANCE = 'NO';

    LABEL SIG95_MEAN = 'SIGNIFICANT AT 95 PERCENT CONFIDENCE LEVEL (Y/N)?';
    LABEL SIG95_VARIANCE = 'SIGNIFICANT AT 95 PERCENT CONFIDENCE LEVEL (Y/N)?';
    LABEL SIG99_MEAN = 'SIGNIFICANT AT 99 PERCENT CONFIDENCE LEVEL (Y/N)?';
    LABEL SIG99_VARIANCE = 'SIGNIFICANT AT 99 PERCENT CONFIDENCE LEVEL (Y/N)?';
```

```
/*************************************************************************/
/* CALCULATE THE RELATIVE EFFICIENCY OF THE MI ESTIMATES                 */
/*************************************************************************/

  RE_MEAN = (1 + (GAMMA_MEAN / 5)) ** -1;
  RE_VARIANCE = (1 + (GAMMA_VARIANCE / 5))** -1;
   LABEL RE_MEAN = 'RELATIVE EFFICIENCY OF THE MI ESTIMATE OF THE MEAN';
   LABEL RE_VARIANCE = 'RELATIVE EFFICIENCY OF THE MI ESTIMATE OF THE VARIANCE';

RUN;

/*************************************************************************/
/* DISPLAY RUBIN'S TEST STATISTICS                                       */
/*************************************************************************/

TITLE6 'RUBIN''S TEST STATISTICS FOR THE MEAN';
ODS PROCLABEL 'RUBIN''S TEST STATISTICS FOR THE MEAN';

PROC PRINT DATA = RE NOOBS LABEL;
  VAR RE_MEAN GAMMA_MEAN Q_BAR_MEAN H_MEAN P_MEAN SIG95_MEAN SIG99_MEAN;
RUN;

TITLE6 'RUBIN''S TEST STATISTICS FOR THE VARIANCE';
ODS PROCLABEL'RUBIN''S TEST STATISTICS FOR THE VARIANCE';

PROC PRINT DATA = RE NOOBS LABEL;
  VAR RE_VARIANCE GAMMA_VARIANCE Q_BAR_VARIANCE H_VARIANCE P_VARIANCE
      SIG95_VARIANCE SIG99_VARIANCE;
RUN;

%MEND STATISTICS;
```

VI. SUMMARY

Multiple imputation is a statistical procedure that helps solve the problem of missing data. In the present case, data are missing due to nonresponse to income questions in the Consumer Expenditure Survey. This paper describes a multiple imputation system built as an alternative to PROC MI and PROC MIANALYZE, the SAS procedures available starting in version 8. In the custom-made system, built to better suit data production purposes, the model-based imputation modules incorporate major analytical components of the imputation procedure. With fewer input parameters required, the module processes the regression, imputation, and evaluation steps all at once. The modules, built in SAS, have advantages over the built-in PROCs, including visibility of process, and flexibility in handling special statistical problems encountered during the implementation of the multiple imputation system. The resultant “complete data sets” produced by the multiple imputation process will be used to publish Consumer Expenditure Survey data, and are expected to result in improved data quality.

REFERENCES

[1] Rubin, Donald B. (1987). Multiple Imputation for Nonresponse in Surveys. John Wiley & Sons, Inc, New York.
[2] C. Deutsch, A. Journel. (1998) GSLIB Geostatistical Software Library and User’s Guide 2nd Edition. Oxford University Press, New York.
[3] Yang C. Yuan. (2000). Multiple Imputation for Missing Data: Concepts and New Development, SAS Users Group International Proceedings.
[4] SAS Institute Inc. SAS/STAT Software: Changes and Enhancements, Release 8.2.

DISCLAMATION

This paper describes work in progress. It may not be cited or quoted without express permission of the authors. The views expressed are those of the authors and do not reflect the policies of the Bureau of Labor Statistics (BLS) or the views of other BLS staff members.

CONTACT INFORMATION

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

Author Name: Geoffrey Paulin, Senior Economist
Agency : Bureau of Labor Statistics

Address: 2 Massachusetts Avenue, NE
City state ZIP: Washington, DC 20212
Work Phone: 202 – 691-5132
Email: : paulin.geoffrey@bls.gov
Web: www.bls.gov

Author Name: Shirley Tsai, Software Engineer
Company : User Technology Associates, Inc.
Address: 950 N. Glebe Road Suite 100
City state ZIP: Arlington, VA 22203
Work Phone: 202 – 691-6748
Email: tsai.shirley@bls.gov
Web: www.utanet.com

Author Name: Melissa Grance, Information Technology Specialist
Agency : Bureau of Labor Statistics
Address: 2 Massachusetts Avenue, NE
City state ZIP: Washington, DC 20212
Work Phone: 202 – 691-5684
Email: grance.melissa@bls.gov
Web: www.bls.gov