INTERACTIVE LEAST SQUARES ESTIMATION OF MISSING VALUES IN ANY GLM

Paul D. Hamilton, Patrick D. McCray, and Michael C. Palmer, G. D. Searle & Co.

Abstract. A simple, user friendly method for estimating missing values with the correct error sum of squares is presented. Using the least squares method, the missing values are computed so that their residuals equal zero for the specified model. The necessary SAS code is provided, along with a TSO CLIST which prompts the user for all necessary information, then writes and submits the appropriate command file.

1. Introduction. A statistician commonly receives data from a designed experiment and finds missing values because of an accident or error during the course of the experiment. If repeated measures were collected on experimental units, then one is faced with the possibility of excluding from sophisticated analysis all of the data collected for an experimental unit if only a single observation is missing. Exclusion of valid data from analysis always places one in an uncomfortable position. An alternative to excluding data, if relatively few values are missing from an experiment, is to fill in missing value estimates, and then proceed with the analysis.

Several ways of estimating missing values exist. One of the oldest is Yate's method of least squares missing value estimation (1933), extended by Rubin (1972) to any analysis of variance. This method naturally fits in with the least squares method of PROC GLM, and provides a convenient method of determining missing values for SAS users.

2. Description of Algorithm. Rubin presented a method whereby the vector of least squares missing value estimates, u , is computed using only a routine to find residuals, a routine to invert real symmetric matrices, and a routine to find products of matrices with vectors. The Rubin method is not iterative, only calculating residuals m + 1 times, where m is the number of missing values to be estimated. A brief outline of the estimation algorithm follows.

The method begins by defining a vector r of m elements and a matrix R of size m by m . First, the data vector y of observations is modified by placing a zero at each place in the vector where an estimate is desired, leaving the rest of the y vector alone. Then the desired analysis of variance is run on this y vector. The m residuals, one for each position in y where a zero was placed, form the vector r .

The matrix R is formed by running the desired analysis of variance m times, where each run supplies a column of R . For each of these m runs, all of the elements of the y vector are now set to zero except for a single element in the position in y corresponding to a value to be estimated; this single element is set to one. During each of these runs, the residuals corresponding to the values to be estimated are computed, and then put into a column of R . At the end of m runs, the matrix R has been constructed from the m columns of residuals. Then u , the least square estimate of the missing values, is computed using u=(R**(-1))*(-r). Appendix 1 contains an alternate proof to that given by Rubin, using matrix algebra.

Note that R may be singular, and then a unique u does not exist. The implications of singularity to statistical inference are not explored in this presentation. Rubin has a brief discussion on this topic.

3. SAS Implementation. A combination of SAS DATA and PROC GLM and MATRIX macros computes the least squares estimate u . Appendix 2 contains a listing of the SAS macro code. This macro assumes that the data is held in a disk based SAS database, not in an OS file or a temporary dataset. Also, the missing values must be "present" in the data. That is, the independent or design, variables must be present in the data and the corresponding dependent variables must be set to the SAS missing value code ("."). When these conditions are met, and R is non-singular, the least squares estimate for any model specified in GLM can be obtained.

In order to execute the macro in Appendix 2, the following items must be provided:

(1) the name of the SAS dataset on disk,
(2) the name of the member in the dataset to be analyzed,
(3) the estimated amount of core

(in K units) that the job will require (default = 500K),

(4) a class statement for the GLM (optional),

(5) the SAS name of the data vector of observations,

(6) m, the number of missing values present for that variable,

(7) the model statement for the GLM,

(8) the priority under which the job will run (optional feature),

(9) the name of the new member which the job will be creating,

(10) the name of the new variable being created,

(11) the names of column work variables for PROC MATRIX (Appendix 2),

(12) the names of row work variables for PROC MATRIX (Appendix 2).

These twelve items are provided by placing the appropriate lines in a file with an editor, (see example in Appendix 2), or by executing the TSO CLIST available from the first author. The result is a new SAS dataset, containing the original data plus a new variable consisting of the original values with least squares estimates replacing the missing values.

Several parts of the included CLIST are site-dependent, and need to be changed before an implementation. To facilitate this process, the string "SITE DEPENDENT" appears in the macro in the appropriate locations. To find these locations, one can search for the given character string.

The program assumes that the file will be built in the account the program resides in. Steps may need to be taken which will allow the CLIST to reside in one account, and multiple users to access it from other accounts. The JCL that the program writes is appropriate for TSO, and must be modified for any other operating system's version of SAS. The program performs an error check for the amount of core specified, and sets an error condition if the amount is greater than 1500K. This limit may differ widely among sites or applications. The SAS code for the macros is most easily kept in an external file and read in at the beginning of the run. The macro code and the nested macros may also be merged into one file with a system editor or other system commands.

4. Concluding Remarks. The user of this macro should be aware that the inclusion of least squares missing value estimates in the data vector leaves the sum of squares for error unchanged, but the other elements of the PROC GLM output will not be valid. SAS counts the missing value estimates as observations, inflating the error degrees of fredom by m . Also, one must subtract one degree of freedom for each value estimated. Therefore the degrees of freedom for error in the output must be reduced by 2*m. As a consequence of missing value estimation, the model sums of squares will in general be inflated, and adjustments must be made before valid significance testing may be done. Finally, one should be aware that the estimates are not predicted values, but numbers computed to facilitate analysis of the data vector **y** .

Appendix 1
Matrix Algebra Proof of Method

Let **y** be a N by 1 vector of observations y(i) , i = 1, 2, ..., N . Suppose that there are m observations, 1 <= m <<N , whose values are to be replaced by their least squares estimates. Let **u** denote the m by 1 vector of least squares missing value estimates u(k) , k = 1, 2, ..., m .

Let r be the m by 1 vector of residuals where r(k) is the residual corresponding to the (k)th value to be estimated when all of the y(i) observations to be estimated are assigned the value zero, the other y(i) observations being left alone. Let R be the m by m matrix of residuals where the (k)th column of R contains all of the residuals corresponding to the y(i) values to be estimated. The y(i) values are assigned the value zero except for the one corresponding to the (k)th observation to be estimated, which is assigned the value one. If R is non-singular, then we shall show that u = (R**(-1))(-r).

Consider the following linear model:

(1) y = X b + e , E(e) = 0 ,

E(e'e) = σ^2 I ,

where X is an N by p design matrix, b is the p by 1 vector of parameters, e is an N by 1 vector of errors, and I is the N by N identity matrix. The normal equations **X'X b = X'y** lead to the least squares estimate of the parameters **$\hat{b}$ = GX'y** , where G is a generalized inverse of **X'X**. The least squares estimate of the observations is **$\hat{y}$ = X $\hat{b}$ = XGX'y.** Letting **T = XGX'** this may be written as $\hat{y}$ = T y. The following argument applies in the case of any other linear model for which the statement that $\hat{y}$ = Ty , for some matrix T , can be made.

Let π be a permutation of the first N integers such that y(π 1), y(π 2), ..., y(π m) are the entries of y we wish to replace. Define a N by N matrix Q = (q(i,j)) by q(i,j) = δ (π i,j), where δ is the Kronecker- δ . For z = Q y we may write z as [x' v']' where x is a m by 1 column vector and v is a (N - m) by 1 column vector. Then x(k) = z(k) = y(pik) for k = 1, 2, ..., m and v(j) = z(m+j) = y(π (m+j)) for j = 1, 2, ..., N-m. Therefore x consists of those entries of y we wish to replace and v consists of those entries of y we do not.

Let P = [I(m) O(m,N-m)] where I(m) is the m by m identity matrix and O(m,N-m) is the m by (N-m) matrix of zeroes. Then x = P z = **PQ** y , so **PQ** is a matrix which selects those entries of y we wish to replace. Writing $\hat{x}$ = **PQ** $\hat{y}$ = **PQT** y we can get an expression for the residuals x - $\hat{x}$ of the observations we wish to replace with their least squares estimates. Thus x - $\hat{x}$ = **PQ** y - **PQT** y = **PQ(I - T)Q**(-1)** z .

Writing **Q(I - T)Q**(-1) = [A B]** where **A** is an N by m matrix and **B** is an N by (N-m) matrix, we get

(2) x - $\hat{x}$ = P [**A B**] [x' v']' = **PA** x + **PB** v.

Letting x = 0 we see that **PB** v = r . If x(k) = 1 and x(i) = 0 for all i except i=k, then **PAx** is the (k)th column of **PA** . But for v = 0 the (k)th column of R is x-$\hat{x}$ = **PAx** + **PB** 0 = **PAx** , namely the (k)th column of **PA**. Therefore **PA** = R , and

(3) x - $\hat{x}$ = Rx + r .

Yates (1933) observed that the residuals corresponding to the least squares estimates must be zero, that is x-$\hat{x}$ = 0 . Thus for x to be the least squares estimate u of the missing values, x must satisfy the equation 0 = Rx + r . Therefore when R is non-singular we may write the solution to this equation as

(4) u = (R**(-1))*(-r).

Appendix 2
Rubin Macro

A listing of the SAS code in macro form follows. A copy of the code needed to supply the nested macros follows the main macro, but the execution of the CLIST mentioned before writes out the command file in a much easier manner.

```
* TURN OFF PRINTING TO SAS LOG;
PROC PRINTTO NEW UNIT=20;
DATA DBR01;
* CREATE DATA SET WITH ORIGINAL
   Y VALUES;
SET DSNM;
* READ IN THE ORIGINAL SET
   FROM DISK;
KEEP ORIGY;
* KEEP THE ORIGINAL Y VALUES
ONLY;
PROC MATRIX;
* SET UP OUTPUT MATRIX TO INPUT
TO
  GLM TO GET RESIDUALS;
FETCH Y DATA=DBR01;
* ABOVE DATA STEP;
IF (1 < NCOL(Y)) THEN STOP;
* INPUT MUST BE A COLUMN;
A = (Y=.);
* CHARACTERISTIC FUNCTION OF NE.
;
IF ((A(+,)) < 1) THEN STOP;
* INPUT MUST HAVE MISSING
VALUES;
B = I(NROW(Y));
* IDENTITY MATRIX;
Z = (Y (1.-A))||(B(,LOC(A)));
* CONCATENATE AND SELECT;
OUTPUT Z OUT=DBR02;
* GO CALCULATE RESIDUALS OF Z;
DATA A;   SET DSNM;
DATA B;   SET DBR02;
DATA DBR02;   MERGE A B;
* COMBINE ORIGINAL DATA AND
   MISSING COLS;
DROP ROW; * GET RID OF EXTRA
  MATRIX VARIABLE;
PROC GLM DATA=DBR02;
CLASS MCLASS;
* USER SUPPLIED MACRO;
MODEL MMODEL=
MVARS;
* USER SUPPLIED MACRO;
OUTPUT OUT=DBR03
RESIDUAL =
MROWS;
* USER SUPPLIED MACRO;
PROC MATRIX;
* ESTIMATE MISSING VALUES;
FETCH F DATA = DBR03;
FETCH Y DATA = DBR01;
A = (Y=.);
* CHARACTERSITIC FUNCTION
   OF '.';
M = A(+,);
* NUMBER OF MISSING VALUES;
N = NCOL(F);
* NUM OF COLUMNS IN DATA SET;
```

```
B = LOC(A);
* LOCATION OF MISSING VALUES
   (ROWS)*(1*M);
R   = F(B,((N-(M-1)):N));
* RESIDUALS USING ONES
   FOR MISSING;
IF (ABS(DET(R)) < 0.000001)
 THEN DO;
  NOTE THE MATRIX IS (NEAR)
   SINGULAR);

  NOTE THE MATRIX OF RESIDUALS
    TO BE INVERTED IS;
  PRINT R;
  NOTE THE DETERMINATE IS;
  BDET = DET(R);
  PRINT BDET;
END;
RHO =  F(B,((N-M):(N-M)));
* RESIDUALS USING ALL ZEROS FOR
   MISSING, DATA VALUES FOR
   EVERYTHING ELSE;
X = -(INV(R))*RHO;
* RUBIN'S ESTIMATES;
Y(B,) = X;
OUTPUT Y OUT=DBR04
(RENAME=(COL1= NEWCOL ));
* ORIGINAL DATA PLUS ESTIMATES;
STOP;
DATA E;   SET DSNM;
DATA F;   SET DBR04;  DROP ROW;
DATA DSNNEW ;   MERGE E F;
COMMENT THIS DATASET WAS CREATED
  BY THE RUBIN MACRO;
PROC PRINTTO;
* RESTORES PRINTING TO SAS LOG;
COMMENT END OF RUBIN MACRO CALL;
*******************************;
```

The following shows the format of the SAS command file. The 12th and 13th lines are statements needed by PROC MATRIX; they consist simply of the characters 'ROW' and 'COL' repeated from N = 1 to M+1 times. Thus, if one had 4 missing values, the lines would appear like this:

```
COL1 COL2 COL3 COL4 COL5
ROW1 ROW2 ROW3 ROW4 ROW5
```

Example of SAS Command File

In the following example, items in capital letters represent code that must be entered exactly as shown. The underlined lowercase text is replaced by the appropriate information for a given application.

```
//  EXEC SAS,REGION=unitsK
//RUBIN DD DSN=filename,DISP=OLD
//FT20F001 DD UNIT=SYSDA,
//     SPACE=(TRK,(20,5))
//SYSIN DD DSN=macro,DISP=OLD
//       DD  *
 MACRO DSNM RUBIN.dataset name %
 MACRO MCLASS  model statement %
 MACRO ORIGY original variable %
 MACRO MVARS model variables %
 MACRO DSNNEW RUBIN.new ds name %
 MACRO NEWCOL new var name %
 MACRO MMODEL COL1  to COLm+1 %
       (needed by MATRIX)
 MACRO MROWS  ROW1  to ROWm+1 %
       (needed by MATRIX)
 RUBIN
       (starts main macro)
```

The authors wish to express their appreciation to Paul von Doehren and Peter Nelson for their helpful suggestions.

In order to obtain a copy of the TSO CLIST, contact the first author at the following address:

G. D. Searle & Co.
4901 Searle Parkway Skokie, Il 60077
(312) 982-8196

Appendix 4
References

1. D. B. Rubin, "A Non-iterative Algorithm for Least Squares Estimation of Missing Values in any Analysis of Variance Design." Applied Statistics: Journal of the Royal Statistical Society (Series C) 21 (1972) 136-141.

2. F. Yates, "The Analysis of Replicated Experiments when the Field Results are Incomplete." Emp. J. Exp. Agric., 1 (1933) 129-142.