Paper ST-08-2011

Acknowledging the Unknown: A SAS® Macro for Investigating Omitted Variable Bias in Two-Level Linear Models

Jason A. Schoeneberger
Bethany A. Bell
University of South Carolina
Jeffrey D. Kromrey
University of South Florida

ABSTRACT

Albeit model specification is an essential aspect of any statistical model, there is little evidence to suggest that applied researchers adequately consider the impact of model misspecification. To help address this important issue, in this paper we introduce users to MIXED_OVA, a SAS macro for conducting sensitivity analysis of hypothetical omitted variable bias in two-level linear models. By utilizing data from PROC MIXED ODS tables in conjunction with PROC IML data simulation, the macro provides a comparison of parameter estimates, standard errors, and *p*-values from a user's analytic model with those generated from a model that contains the hypothetical omitted variable. This paper provides the macro programming language, as well as results from an executed example.

Keywords: MODEL SPECIFICATION, SENSITIVITIY ANALYSIS, PROC MIXED

INTRODUCTION

Consideration of specification errors is important in any statistical model. A variety of specification errors may lead to biased inferences, including errors arising from omitted variables, regressor measurement errors, sample self-selection, and functional forms of relationships (Ebbes, Böckenholt, & Wedel, 2004).

Omitted variable bias is a particularly pernicious problem with non-experimental studies (e.g., observational studies). In such research, important explanatory variables may not be measured because the researcher is either not aware of such variables or is unable to collect the requisite data. When important variables are omitted from a statistical model, the predictor variables and residuals become correlated. This, in turn, leads to biased and inconsistent estimates of the model parameters. Caveats about the interpretation of observational data and concerns about omitted variables have been raised by Box (1966), Marcus (1997), Mosteller and Tukey (1977), Rubin (1978), Rosenbaum and Rubin (1983), and Rosenbaum (1986, 1991). Such caveats highlight the uncertainty in any sample about the degree of bias resulting from omitted variables.

Kim and Frees (2006) illustrate the problem by distinguishing between a "true" model (including all variables that affect the dependent variable) and an estimated model (that includes only some of these variables). The true model may be written as

$$y = X\beta + U\gamma + \varepsilon$$

in which ***y*** is the criterion variable, ***X*** is the matrix of observed explanatory variables, ***U*** is the matrix of unobserved explanatory variables, and ***ε*** is the residual or disturbance term.

The estimated model is based only on the variables in ***X*** (because the variables in ***U*** have not been observed) and may be written as

$$y = X\beta + \tilde{\varepsilon}$$

in which $\tilde{\varepsilon} = U\gamma + \varepsilon$.

The least squares estimates of the parameters in the estimated model are not ***β***, but are $\beta + \left(X'X\right)^{-1} X'U\gamma$. Unless the omitted variables are uncorrelated with the observed predictors (i.e., $X'U = 0$) or are unrelated to the criterion variable (i.e., $\gamma = 0$), the estimated regression coefficients and their standard errors will be both biased and inconsistent.

In multilevel models, omitted variable bias may be particularly problematic because of its impact on estimates of both fixed and random effects. In addition, omitted variables at lower levels of the model may produce greater bias than omitted variables at higher levels (Kim & Frees, 2005, 2006). An important issue for research involving multilevel models is the assessment of the potential impact of omitted variables.

Although model specification issues have long plagued applied inquiry, there is little evidence to suggest that researchers either consider the impact of misspecification or evaluate the integrity of their models. Such complacency around this issue is both outdated and unwarranted. In an effort to help researchers reduce the uncertainty of omitted variable bias in their multilevel models, this paper provides a SAS macro to conduct omitted variable analysis within two-level linear models. For both fixed and random effects, the macro provides an examination of changes in parameter estimates and standard errors as a function of the association between a hypothetical omitted variable and both the observed predictors and the criterion variable.

MACRO MIXED_OVA DETAILS

The MIXED_OVA macro was developed to examine omitted continuous variable bias in two-level models estimated using PROC MIXED. Written using both SAS IML and SAS/STAT, the macro utilizes both the correlation matrix from the user's analytic model and a simulated correlation matrix that consists of the analytic correlation matrix plus the inclusion of a hypothetical omitted variable (simply referred to as omitted variable henceforth). This allows researchers to compare model results based on their observed data with those from models estimated from the simulated data. A complete copy of the MIXED_OVA macro can be downloaded from the second author's website (http://www.ed.sc.edu/bell/).

Specifically, the MIXED_OVA macro output contains two summary tables displaying PROC MIXED model estimates based on simulated data using the original set of predictors and corresponding average bias for each parameter estimate based on models using the simulated data augmented with the omitted variable (i.e. the bias is estimated as the difference between the average results from the augmented models and the original-predictor models). One table is generated for the fixed effects and another for the random effects. The omitted variable can occur at either level-1 or level-2; however, the macro can only examine the impact of one omitted variable at a time and the omitted variable can only be a continuous variable.

Macro inputs, in addition to the inclusion of PROC MIXED code for model specification, include:

path: folder location where the data file for analysis resides
data: this argument is the name of the data file containing the data to be analyzed and simulated.
dv: the criterion variable
lvl1iv: the list of level-1 predictors separated by a space
lvl2iv: the list of level-2 predictors separated by a space
interact: the interaction effects, separated by a space in alphabetic order (e.g. female*ses, not ses*female)
random: the list of random effects to be modeled separated by a space (random intercepts as "intercept"
lvl2id: the variable denoting cluster id
ddfm: degrees of freedom calculation method, default=contain
cov: covariance structure (only variance components [vc] or unstructured [un] are acceptable entries)
r_x: correlation of omitted variable with level-1 predictors, if blank average correlation is calculated
r_z: correlation of omitted variable with level-2 predictors, if blank average correlation is calculated
gamma: relationship of omitted variable to criterion variable
alpha: desired alpha level for determining rate of rejection in analyses of simulated data
omitlvl: level of model where omitted (continuous) variable occurs (1 or 2)
sim_n: number of simulated datasets desired

After the user enters the macro input information and specifies her/his PROC MIXED model, MIXED_OVA operates by first calculating the correlation matrix from the analytic model (i.e., the hypothetical misspecified model). Second, through the use of SAS IML procedures, the observed correlation matrix is then augmented to include the omitted variable, as specified by the user in the macro input commands (e.g., the correlation of the omitted variable with level-1 predictors, the level of the model that the omitted variable occurs). Third, using a specified number of replications (e.g. sim_n=1,000), the macro uses the simulated correlation matrix to estimate the user's previously specified PROC MIXED code plus the omitted variable as an additional predictor (i.e., the hypothetical correctly specified model). Fourth, using information from PROC MIXED ODS tables, the macro calculates the average parameter estimates and standard errors for both fixed and random effects from the simulated, correctly specified models. Fifth, using these average values, estimates of bias in parameters, degrees of freedom and *p*-values are calculated as the difference between the average simulated values using augmented data and the values based on the predictors from the original analytic model. Lastly, using PROC PRINTTO, the macro generates summary output tables for fixed and random effects that contain the original misspecified parameter estimates, standard errors, and *p*-values along with estimated levels of bias for each outcome.

To help users determine possible correlation and gamma values to include in the macro input, MIXED_OVA provides options for users to conceptualize these relationships (i.e., use the average gamma value to specify the relationship between the hypothetical omitted variable and the dependent variable and use the average correlation among observed predictor variables to specify the relationship between the hypothetical omitted variable and other predictors). With this in mind, the macro is intended to be used in an interactive fashion, allowing users to select different correlation levels as they see fit (i.e., selecting various values to represent realistic values given the particular nature of the research context).

Users should also note several limitations with the version of the MIXED_OVA macro. First, the macro can only handle a Variance Component (VC) and Unstructured (UN) random effect covariance structures. Second, MIXED_OVA has not been developed for use with repeated measurers analyses conducted within the multilevel framework. The authors will continue to explore these and other options, so readers should check second author's website (http://www.ed.sc.edu/bell/) for the latest MIXED_OVA release.

The macro begins by creating a copy of the user-supplied data file in the SAS work directory and adding a running count variable denoting level-2 unit membership. In addition, dynamic macro calls are created based on the vector of level-1 and level-2 predictors supplied by the user for later processing. Here, PROC MIXED is used to fit the user-specified original multilevel model. Based on the criterion, level-1, level-2, interact, random, and ddfm input values, the PROC MIXED code below generates the fixed and random effect estimates associated with the misspecified model. These values are used in subsequent processing when calculating bias estimates.

```
**mixed code to obtain parameter estimates using original data;
ods listing close;
proc mixed data=indata noclprint covtest;
      class &lvl2id;
      model &dv=&lvl1iv &lvl2iv &interact/solution ddfm=&ddfm;
      %if %length(&random) ne 0 %then %do;
            random &random/subject=&lvl2id type=&cov;
            ods output solutionf=realfixeff covparms=realcov;
      %end;
      %if %length(&random) eq 0 %then %do;
            ods output solutionf=realfixeff;
      %end;
      run;
ods listing;
```

Next, the mean values of the criterion, level-1 and level-2 variables are calculated and the total number of level-2 units is identified. SAS syntax to call the GENDAT macro is written within the data file and exported into the GCODE macro argument to be called within PROC IML processing. Finally, the mean values of the criterion, level-1 and level-2 variables are calculated across level-2 units and outputted to a SAS dataset that will be read into matrix form within PROC IML.

```
**calculate mean dv, level-1 and level-2 predictor values at level-2;
proc sort data=indata;
      by &lvl2id;
proc means noprint data=indata;
      by &lvl2id;
      var &dv &lvl1iv &lvl2iv;
      output out=indata_l2(rename=(_freq_=lvl1_count)drop=_type_) mean=;
      run;

**calculate the means and variances of level-2 predictors, across level-2 units;
**results are read into matrices for gendat iml routine for producing level-2 data;
proc means data=indata_l2 noprint mean;
      var &lvl1iv &lvl2iv;
      output out=l2_mean (drop=_type_ _freq_)mean=;
      run;
proc means data=indata_l2 noprint var;
      var &lvl1iv &lvl2iv;
      output out=l2_var (drop=_type_ _freq_)var=;
      run;

**write macro code for level-1 gendat IML subroutine processing by level-2 unit;
```

```
data indata_l2;
       set indata_l2;
       **create count of level-2 units;
       count+1;
       gencode=cat('%gendat(',lvl1_count,',',count,');');
       keep &lvl2id lvl1_count &dv &lvl1iv &lvl2iv count gencode;
       run;

**count the number of level-2 records for input into gendata call;
proc sql noprint;
select count(&lvl2id), gencode
              into : n2ss,
                   : gcode separated by ' '
       from indata_l2;
       quit;
```

In the instance when a user does not supply a specific correlation between the omitted variable and the other included predictors, a pooled correlation among level-1 predictors is calculated. The following section of code creates a data file used to define the macro arguments necessary for the calculation of the pooled correlation matrix.

```
**create l2_pool file for generating code to calculate pooled correlation matrix;
data l2_pool;
       retain &lvl2id l2id nplus nsub;
       set indata_l2;
       by &lvl2id;
       **this creates running count of level-2 records;
       if first.&lvl2id then l2id+1;
       **this creates code for generating pooled covariance matrix (per J. Kromrey);
       if (l2id ne &n2ss) then do;
              nplus=cat('n',l2id,'+');
              nsub=cat('(n',l2id,'-1)*s',l2id,'+');
       end;
       else do;
              nplus=cat('n',l2id,'-');
              nsub=cat('(n',l2id,'-1)*s',l2id);
       end;
       **keep only primary variables of interest;
       keep &lvl2id l2id nplus nsub;
       run;
```

The following code compiles the L1STAT macro, processing a number of data points for each level-1 predictor provided by the user. The mean and variance of each level-1 predictor within each level-2 unit is calculated and outputted to a SAS dataset in the work directory. In addition, a SAS dataset is compiled holding all fixed effect estimates for further processing.

```
**this macro will create level-1 stats by level-2 group id;
**read parameters from estimated model and create macro arguments for each parameter;
%macro l1stat (arg,arg2);

**calculate level-1 means by level-2 groups;
proc sql noprint;
       create table l1_mean_&arg2 as
       select &lvl2id, mean(&arg) as l1_mn_&arg2
       from indata
       group by &lvl2id;
       quit;

**calculate level-1 variances by level-2 groups;
proc sql noprint;
       create table l1_var_&arg2 as
       select &lvl2id, var(&arg) as l1_var_&arg2
       from indata
       group by &lvl2id;
       quit;
```

```
**create overall files holding all level-1 means and vars by level-2 group;
%if &arg2=1 %then %do;
      data l1_mean; set l1_mean_&arg2;  run;
      data l1_var; set l1_var_&arg2;run;
%end;
%else %do;
      data l1_mean;merge l1_mean l1_mean_&arg2;by &lvl2id;run;
      data l1_var;merge l1_var l1_var_&arg2;by &lvl2id;run;
%end;

**compile a data file of only the level-1 fixed effects;
proc sql noprint;
create table l1_fix_&arg2 as
select effect, estimate
      from realfixeff
      where (effect="&arg");
      quit;

**create overall file holding all level-1 effect estimates;
%if &arg2=1 %then %do;
      data l1_fix;set l1_fix_&arg2;run;
%end;
%else %do;
      data l1_fix;set l1_fix l1_fix_&arg2;run;
%end;
**end macro;
%mend l1stat;

**this code runs macro call to cycle through for each level-1 var listed;
%do i=1 %to &p_x;
      %unquote(&&l1&i);
%end;
```

Here we use the SAS dataset l1_fix containing the fixed effect estimates from the misspecified model to write dynamic SAS syntax to generate simulated predicted values via a multilevel regression formula using estimated parameter values. The written code is then output to macro arguments that can subsequently be called when necessary. Figure 1 shows a snapshot of the dataset after the MIXED_OVA macro has been executed. Similar processing is done for level-2 fixed, random, and interaction effects, generating similar statements for use in generating predicted values.

```
**use level-1 fixed estimate file and write call statements for each level-1 effect,
in numeric order of entry by user;
data l1_fix;
      set l1_fix;
      count+1;
      countx=count*10;
      fixcode=cat('%let gam',compress(countx),'=',estimate,';');
      ccode=cat('"',compress(effect),'"');
      run;
```

	Effect	Estimate	count	countx	fixcode	ccode
1	female	-1.1701	1	10	%let gam10=-1.17006706032821;	"female"
2	ses	2.0002	2	20	%let gam20=2.000189987;	"ses"

Figure 1. Snapshot of l1_fix dataset with variables containing dynamic SAS code.

The following code compiles the L1ICC macro for calculation of the average level-1 intra-class correlation (correlation within level-2 units) across the level-1 predictors. PROC MIXED is used to model the variance of each level-1 predictor attributable to level-1 (σ) and level-2 (τ) for use in calculating the ICC value as:

$$ICC = \tau/(\tau + \sigma)$$

The calculated ICC values for each predictor are outputted to a SAS dataset, l1_icc , for use in calculating a level-2 variance output to a macro argument L2_VARIMP, calculated as:

$$Level - 2\ variance = \bar{\rho}/(1 - \bar{\rho})$$

where $\bar{\rho}$ is the mean ICC value across all level-1 predictors.

```
**this macro calculates the average level-1 icc value;
%macro l1icc (arg,arg2);
**use proc mixed to get tau and sigma values for each level-1 variable;
ods listing close;
proc mixed data=indata noclprint covtest;
       class &lvl2id;
       model &arg=/solution ddfm=&ddfm;
       random intercept/subject=&lvl2id;
       ods output covparms=&arg.cov;
       run;
ods listing;

**calculate icc for each level-1 variable;
data &arg.cov;
       retain &arg._tau &arg._sigma;
       set &arg.cov;
       if (covparm="Intercept") then &arg._tau=estimate;
       if (covparm="Residual") then &arg._sigma=estimate;
       icc=&arg._tau/(&arg._tau+&arg._sigma);
       keep &arg._tau &arg._sigma icc;
       run;

**create overall file holding all level-1 icc;
%if &arg2=1 %then %do;
       data l1_icc;  set &arg.cov; run;
%end;
%else %do;
       data l1_icc;set l1_icc &arg.cov;
       run;
%end;

**end macro;
%mend l1icc;

**cycle through the level-1 variables;
%do i=1 %to &p_x;
       %unquote(&&l1icc&i);
%end;

**output l2_varimp macro arg as mean icc of level-1 variables (rho-bar/(1-rho-bar);
proc sql noprint;
select mean(icc)/(1-mean(icc)) into : l2_varimp
       from l1_icc;
       quit;
```

In the instance when a user does not supply a specific correlation between the omitted variable and the other included predictors, a pooled correlation among level-1 predictors is calculated. The following section of code creates a data file used to define the macro arguments necessary for the calculation of the pooled correlation matrix.

```
**create l2_pool file for generating code to calculate pooled correlation matrix;
data l2_pool;
       retain &lvl2id l2id nplus nsub;
       set indata_l2;
       by &lvl2id;
       **create running count of level-2 records;
       if first.&lvl2id then l2id+1;
       **create code for pooled covariance matrix;
       if (l2id ne &n2ss) then do;
              nplus=cat('n',l2id,'+');
              nsub=cat('(n',l2id,'-1)*s',l2id,'+');
       end;
       else do;
              nplus=cat('n',l2id,'-');
```

l2id	nplus	nsub
1	n1+	(n1-1)*s1+
2	n2+	(n2-1)*s2+
3	n3+	(n3-1)*s3+
4	n4+	(n4-1)*s4+

Figure 2. Pooled correlation setup code.

```
            nsub=cat('(n',l2id,'-1)*s',l2id);
      end;
      **keep only primary variables;
      keep &lvl2id l2id nplus nsub;
      run;
```

The next portion of code is contained within a macro DO LOOP to repeat the number of times supplied by the user in the SIM_N macro argument (e.g. sim_n=1000). PROC IML is executed, and processing within the IML environment starts with the calculation of the pooled level-1 correlation matrix via the calculation of a covariance matrix for each level-2 unit that is subsequently pooled across level-2 units. In addition, the level-2 data generated above are used to create a level-2 correlation matrix denoted z (syntax not shown).

```
**this command executes all of the following sim_n-times (1000);
%do j=1 %to &sim_n;

**call iml;
proc iml;
**read indata, retaining only dv and level-1 iv, create pooled correlation matrix;
use indata;
      ** the do loop cycles for every level-2 unit;
      %do i=1 %to &n2ss;
            read all var {&dv &lvl1iv} where (l2id=&i) into y&i;
            **set number of groups=k;
            k=&n2ss;
            p=ncol(y&i);
            **compute covariance matrix for each group;
            n&i=nrow(y&i);
            y&i.bar=1/n&i*y&i`*j(n&i,1);
            s&i=1/(n&i-1)*y&i`*(i(n&i)-1/n&i*j(n&i))*y&i;
      %end;
      **compute pooled covariance matrix - the macro args here were created above;
      spl=1/(&nplus - &n2ss)*(&nsub);
      **convert covariance matrix to correlation matrix;
      d=sqrt(vecdiag(spl));
      r=spl/d`/d;/** divide columns, then divide rows **/
```

Here level-1 and level-2 mean and variance vectors are created based on the corresponding SAS datasets created previously. In the instance where the user has specified the omission of a variable at level-1, the level-1 mean vector has an additional cell added to reflect a mean of zero and the variance vector is augmented with an additional cell containing a 1 suggesting that the omitted variable has a variance equal to 1. For the level-2 variance, the vector is augmented with an additional cell containing the mean level-2 variance of level-1 predictors based on the calculation of mean ICC values executed previously. Vectors representing level-1 and level-2 means and variances, when the omission occurs at one of these levels, are also augmented to reflect the assumption that the omitted variable has a mean equal to zero and a variance of 1.

```
**read-in files containing mns and vars of level-1 variables within each level-2 unit;
use l1_mean; read all into l1_mean;
use l1_var;  read all into l1_var;
**add a variance of 1 for omitted variable at level-1;
%if (&omitlvl=1) %then %do;
     l1_mean=l1_mean||j(&n2ss,1,0);
     l1_var=l1_var||j(&n2ss,1,1);
%end;

**read-in files containing mns/vars of level-2 variables across level-2 units;
use l2_mean;  read all into l2_mean;
use l2_var;   read all into l2_var;
**add a zero for the omitted variable mean at the end of the l2_mean vector;
**add the l2_varimp (average level-1 icc) as omitted variable level-2 variance;
%if (&omitlvl=1) %then %do;
     l2_mean=l2_mean||j(1,1,0);
     l2_var=l2_var||j(1,1,&l2_varimp);
%end;
%if (&omitlvl=2) %then %do;
     l2_mean=l2_mean||j(1,1,0);
```

```
     l2_var=l2_var||j(1,1,1);
%end;
```

The next section of code manipulates the level-1 and level-2 correlation matrices based on input provided by the user. Correlation matrices are manipulated depending on the level of the omitted variable and the supplied correlation of the omitted variable with other predictors. For example, below is the code that handles the level-1 correlation matrix when the omitted variable is at level-1 and the user did not supply a hypothetical correlation value. The values of the cells in the correlation matrix are summed and divided by the number of elements to calculate an average correlation coefficient. This value then serves as the hypothetical correlation value between the omitted variable and other predictors, given the lack of a user-specified correlation. The original x matrix is then augmented to include an additional level-1 variable in the matrix, where each cell contains the average correlation coefficient and is outputted to the corrxiv matrix which will be used to generate level-1 data. The corrziv matrix represents the level-1 and level-2 correlation matrix for generating data at level-2. This matrix is also augmented with the average level-1 correlation for the omitted variable. Similar manipulations are carried out when the user supplies a hypothetical correlation value, though the matrices are augmented with the supplied value. Finally, the level-2 correlation matrix is manipulated in a similar manner when the omitted variable occurs at level-2.

```
**calculate average level-1 correlation if user does not supply correlation;
%if (&omitlvl=1) & (%length(&r_x)=0) %then %do;
       **retain only level-1 predictors from x matrix, which included dv from above;
       xs=px[2:&p_x+1,2:&p_x+1];
       **calculate average correlation of level-1 variables in the original data file;
       sum_corr_x=0;
       n_elements=0;
       **run for each row and column;
       do row=1 to nrow(xs);
              do col=1 to ncol(xs);
                     **grab elements below diagonal;
                     if row < col then do;
                            **mn_corr_x=sum lower diagonal;
                            sum_corr_x=sum_corr_x+xs[row,col];
                            **this counts the number of elements below the diagonal;
                            n_elements=n_elements+1;
                     end;
              end;
       end;
       **calculate the mean correlations;
       mn_corr_x=sum_corr_x/n_elements;
       **set r_x=average correlation, making the scalar value character;
       call symput('r_x',left(char(mn_corr_x)));
```

Figure 3 below depicts the progression from the original level-1 correlation matrix to an augmented matrix incorporating another, omitted predictor assuming the average correlation among the original predictors. The level-2 correlation matrix, which contains both level-1 and level-2 predictors, will also be augmented accordingly.

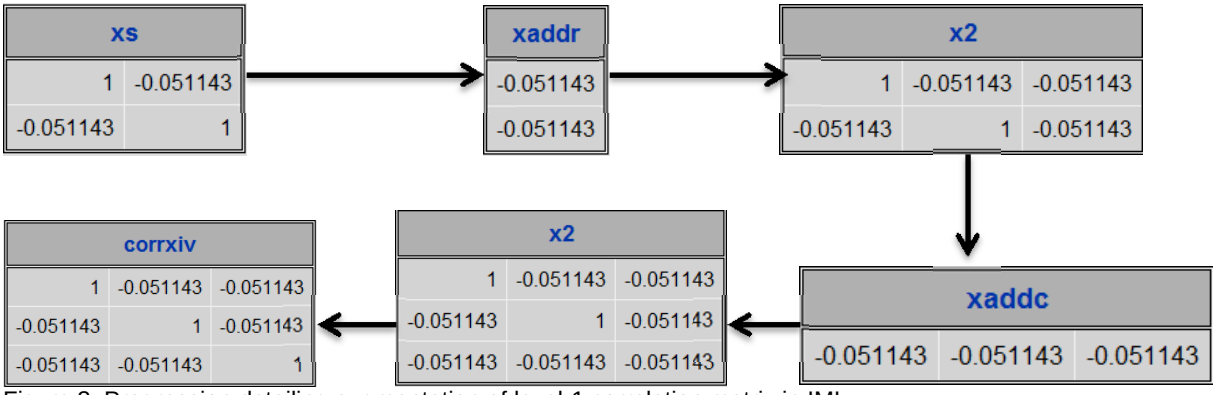

Figure 3. Progression detailing augmentation of level-1 correlation matrix in IML.

The following PROC IML subroutine called GENDATA is used to generate the simulated data based upon the vectors of means and variances, along with the inputted correlation matrix representing the interrelationships between variables. The code presented here is then executed using the 'run gendata' commands for generation of the level-2 simulated data into a matrix called grp_data.

```
start gendata(nn,seed1,variance,bb,cc,dd,mu,r_matrix,rawdata);
      **this create L matrix set equal to eigenvalues from corr matrix defined above;
      L=eigval(r_matrix);
      **this sets neg_eigval=0 in row 1, col 2 of L;
      neg_eigval=0;
      **do r 1 to the number of rows (nrow) in L;
      do r=1 to nrow(L);
            **if a value in the r-by-1 element is less than zero, set neg_eigval=1;
            if L[r,1] < 0 then neg_eigval=1;
      end;

**if matrix is positive definite, use Cholesky root approach;
if neg_eigval=0 then do;
      **set cols=number of columns in the correlation matrix (corr) from above;
      cols=ncol(r_matrix);
      **set g=cholesky decomposition of the correlation matrix (corr) from above;
      g=root(r_matrix);
      **rawdata=random values from normal distribution with mean=0, SD=1 (rannor);
      **repeat creates raw data matrix repeated across nn-by-cols;
      **e.g. if nn=10 and cols=5, then rawdata is 10-by-5 matrix of random numbers;
      rawdata=rannor(repeat(seed1,nn,cols));
      **then rawdata is multiplied by g matrix as matrix product;
      rawdata=rawdata*g;
      **the rawdata matrix is altered using the Fleishman constants for each NN rows
      and COLS columns, for each element;
      do r=1 to nn;
      do c=1 to cols;
      **alter element r,c by adding the following calculations;
      **sum of linear combination of standard normal variables [equation 3];
      **this produces zero mean, unit variance and desired (3rd and 4th moments);
            rawdata[r,c]=(-1*cc) + (bb*rawdata[r,c]) + (cc*rawdata[r,c]##2) +
            (dd*rawdata[r,c]##3);
            rawdata[r,c]=(rawdata[r,c] * sqrt(variance[1,c])) + mu[1,c];
      end;
      end;
end;

**if matrix is not positive definite, use PCA approach;
if neg_eigval=1 then do;
      **set cols=number of columns in the correlation matrix (corr) from above;
      cols=ncol(r_matrix);
      **set v=right eigenvectors of the correlation matrix (corr);
      v=eigvec(r_matrix);
      **the L eigenvalue matrix is altered for each row and column;
      do i=1 to nrow(L);
      do j=1 to ncol(v);
      **if row i, column 1 of L is gt zero, then j,i of v=v(j,i)*sqrt of L(i,1);
            if L[i,1] > 0 then v[j,i] = v[j,i] # sqrt(L[i,1]);
            **if row i, column 1 is le zero, then element v=v(j,i)*sqrt small number;
            if L[i,1] <= 0 then v[j,i] = v[j,i] # sqrt(.000000001);
            end;
      end;
      **rawdata=random values from normal distribution with mean=0, SD=1 (rannor);
      rawdata=rannor(repeat(seed1,nn,cols));
      **then multiply rawdata inverse by V matrix results;
      rawdata=V*rawdata`;
      **set rawdata= rawdata inverse;
      rawdata=rawdata`;
      **the rawdata matrix is altered using the Fleishman constants for each NN rows
      and COLS columns, for each element;
      do r=1 to nn;
      do c=1 to cols;
            **alter element r,c by adding the following calculations;
            **this produces zero mean, unit variance and desired (3rd & 4th moments);
```

```
            rawdata[r,c]=(-1*cc) + (bb*rawdata[r,c]) + (cc*rawdata[r,c]##2) +
            (dd*rawdata[r,c]##3);
            rawdata[r,c]=(rawdata[r,c] * sqrt(variance[1,c])) + mu[1,c];
      end;
      end;
end;
finish;

**set random seed value;
seed1=round(1000000*ranuni(0));

**the following statement executes the gendata module defined above;
run gendata(&n2ss,seed1,l2_var,1,0,0,l2_mean,corrziv,grp_data);
```

The GENDAT macro that follows utilizes the gendat IML subroutine in an iterative fashion, cycling through each level-2 unit generating data at level-1. In combination with the level-2 gendata subroutine call above, a simulated data file containing the same number of level-2 and level-1 units within each level-2 units is generated.

```
%macro gendat (n1ss,idlevel2);
      **set n1=value obtained from original data set;
      n1=&n1ss;
      **create a level-2 id;
      idlevel2=j(n1,1,&idlevel2);
      **set population mean=l1_mean obtained from the gendata at level-2;
      %if (&omitlvl=1) %then %do;
            first=grp_data[&idlevel2,1:&p_x];
            second=grp_data[&idlevel2,&p_x+&p_z+1];
            pop_mn1=first||second;
            **sets population variance=original l1_var for each level-2 unit;
            pop_var1=l1_var[&idlevel2,2:&p_x+2];
      %end;
      **set population mean=l1_mean obtained from the gendata at level-2;
      %if (&omitlvl=2) %then %do;
            pop_mn1=grp_data[&idlevel2,1:&p_x];
            **set population variance=original l1_mean for each level-2 unit;
            pop_var1=l1_var[&idlevel2,2:&p_x+1];
      %end;
      **utilized the gendata module to conduct similar routine for level-1 data;
      run gendata(n1,seed1,pop_var1,1,0,0,pop_mn1,corrxiv,x);

      **create zz matrix across level-1 when missing data is at level-1;
      %if (&omitlvl=1) %then %do;
            zz=repeat(grp_data[&idlevel2,&p_x+1:&p_x+&p_z],n1);
      %end;
      **create zz matrix across level-1 when missing data is at level-2;
      %if (&omitlvl=2) %then %do;
            zz=repeat(grp_data[&idlevel2,&p_x+1:&p_x+&p_z+1],n1);
      %end;

      **create obs_data as concatenation of idlevel2, x and zz;
      obs_data=idlevel2||x||zz;
      **create r matrix of zeroes with rows=number of n1 and 1 column;
      r=j(n1,1,0);
      **for each element in the vector of r, set each value to random number;
      do i=1 to n1;
            r[i,1]=rannor(0)*sqrt(&resid_var);
      end;
```

Here the simulated dependent variable Y is created in a dynamic fashion, dependent upon the level of the omitted variable supplied by the user. The dependent variable is generated using the appropriate multilevel regression formula, applying the intercept and coefficients from the original model to the simulated data, along with the user-supplied coefficient (gamma macro argument) applied to the simulated, omitted variable. Once the dependent variable has been calculated, the column vector is appended to the simulated data matrix and output to a traditional SAS data file before exiting IML.

```
      **create y-values based on parameters from original model above;
```

```
      %if (&omitlvl=1) %then %do;
            y=&gamxf (&gamma.#obs_data[,&p_x + 2])+ &fixzf &icod r;
      %end;
      %if (&omitlvl=2) %then %do;
            y=&gamxf +(&gamma.#obs_data[,&p_x + &p_z + 2]) + &fixzf &icod r;
      %end;
```

This translates into the following formula to generate the dependent variable:

y=&gam00+tau_data[&idlevel2,1])+&gam10#obs_data[,2]… +
&gamma(simulated omitted variable)+
&gam01#obs_data[,4]+&gam02#obs_data[,5]+
&gam50#obs_data[,1]#obs_data[,2])+
random number *sqrt(residual variance)

Subsequently, the simulated dependent variable is appended to the observed data matrix and exported for analysis outside IML.

```
      **horizontally concatenate obs_data with the y variable;
       obs_data=obs_data||y;

       ************************************************
       send simulated samples to regular SAS for analysis
       ************************************************;
       **this command declares the columns names of the outputted dataset;
       %if (&omitlvl=1) %then %do;
       cname={"IDLevel2" &fixxc "omit" &fixzc "&dv" };
       %end;
       %if (&omitlvl=2) %then %do;
       cname={"IDLevel2" &fixxc &fixzc "omit" "&dv" };
       %end;

       **create data file called simdat from obs_data, where column names=cname;
       if &idlevel2=1 then do;
             create simdat from obs_data[colname=cname];  append from obs_data;
       end;
       if &idlevel2 > 1 then do;
             setout simdat;  append from obs_data;
       end;

       **free obs_data of values;
       free obs_data idlevel2 n1 y x zz u0;

%mend gendat;
&gcode;
**this quits iml;
quit;
```

The following PROC MIXED calls analyze the simulated data file, first including the omitted variable and then analyzing the simulated data without the omitted variable (i.e., to generate results that mirror those from the original analytic model). The model results for the fixed effects and variance components are output to SAS datasets using the ODS feature. The fixeff macro processes these resulting data files for the augmented (with the omitted variable) and non-augmented (without the omitted variable) results. A data file containing the PROC MIXED output for each iteration of the overall macro (number of simulations) is compiled before the macro is ended. Similar steps are taken for the covariance parameters.

'omit' removed for non-augmented analysis

```
ods listing close;
proc mixed data=simdat covtest noclprint namelen=32;
      class idlevel2;
      model &dv=&lvl1iv &lvl2iv &interact omit/ solution ddfm=&ddfm;
      %if %length(&random) ne 0 %then %do;
            random &random/ sub=idlevel2 solution type=&cov;
            ods output solutionf=simfixaug covparms=simcovaug;
      %end;
      %if %length(&random) eq 0 %then %do;
            ods output solutionf=simfixaug;
```

```
%end;
run;

**this creates variable specific to augmented or not augmented for fixed
effects;
%macro fixeff (type);
data simfix&type;
      set simfix&type;
      &type._est=estimate;
      &type._stderr=stderr;
      &type._df=df;
      &type._tval=tvalue;
      &type._probt=probt;
      **create rejection indicator;
      &type._rej=0;
      if (probt lt &alpha) then &type._rej=1;
      counter=&j;
      keep effect &type._est &type._stderr &type._df &type._tval &type._probt
      &type._rej counter;
      run;
%mend fixeff;
%fixeff(aug);
%fixeff(noaug);
```

Here an overall file representing the augmented and non-augmented fixed effects across all iterations is generated. Bias variables, representing the difference between the estimates obtained from the augmented and non-augmented analyses are calculated. We estimate the bias in the parameter estimates, standard errors, degrees of freedom and *t*-values. In addition, a flag variable denoting whether each effect was 'rejected' based on the *t*-test was averaged across all simulations for each effect. Subsequently, the bias in the rate of rejection was calculated in a similar fashion.

```
**sort results for merge between files with augmented variable and those without;
proc sort data=simfixaugstack;
      by counter effect;
proc sort data=simfixnoaugstack;
      by counter effect;
data fixbias;
      merge simfixaugstack simfixnoaugstack;
      by counter effect;
      **calculate bias variables;
      est_bias=aug_est-noaug_est;
      se_bias=aug_stderr-noaug_stderr;
      df_bias=aug_df-noaug_df;
      tval_bias=aug_tval-noaug_tval;
      run;

**sort fixed effects from simulated data in prep for proc means;
proc sort data=fixbias;
      by effect;
proc means noprint data=fixbias;
      by effect;
      var est_bias se_bias df_bias tval_bias aug_rej noaug_rej;
      output out=sim_fixed Mean=;

**calculate rejection rate;
data sim_fixed;
      set sim_fixed;
      **calculate p-value bias;
      p_bias=aug_rej-noaug_rej;
      **exclude omit variable from output;
      where (effect ne "omit");
      drop aug_rej noaug_rej;
      run;
```

Next the fixed effect estimates from the original analytic model are merged with the fixed effect results from the simulations conducted. The FILE PRINT command is used to generate a report showing the original fixed effect estimates and the corresponding amount of bias estimated in the simulations based on the omission level, correlation and coefficient input parameters supplied by the user. A similar process is conducted to generate the report presenting the original and estimated biases associated with the variance components.

```
**sort fixed effects from original run (non-simulated data);
proc sort data=realfixeff;
      by effect;
data both_fixed;
      merge realfixeff sim_fixed;
      drop _TYPE_ _FREQ_;
      by effect;
      **recode probability and mean-p values;
      if Probt >= .000001 then ProbtA = put(Probt,8.6);
      if Probt < .000001 then ProbtA = '<.000001';
      if Probt = . then ProbtA = ' ';
      if Mn_p >= .000001 then Mn_pA = PUT(Mn_p,8.6);
      if Mn_p < .000001 then Mn_pA = '<.000001';
      file print header = H notitles;
      put @1 Effect @20 estimate best8. @30 Mn_est best8. @42 StdErr best8. @52 Mn_SE
      best8. @64 DF best8. @74 Mn_DF best8.
            @86 tValue best8. @96 Mn_t best8. @108 ProbtA @118 Mn_pA;
      return;
      H: put @1 'Omitted Variable Analysis' /
      @1 'Fixed Effects' //
      @1 "Gamma Value = &gamma" /
      %if (&omitlvl=1) %then %do;
            @1 "Level-1 Correlation = &r_x" /
      %end;
      %if (&omitlvl=2) %then %do;
            @1 "Level-2 Correlation = &r_z" /
      %end; @1 "Omission Level = &omitlvl" /
      @1 "Number of Simulations = &sim_n" //
      @20 'Parameter Estimate'  @42 '  Standard Error'     @64 'Degrees of Freedom'
      @86 '     t-value'         @108 ' Probability < |t|' /
      @20 '------------------' @42 '------------------'  @64 '------------------'
      @86 '------------------'  @108 '------------------'/
      @1 'Effect'  @20 'Observed  W/Omitted' @42 'Observed  W/Omitted'  @64 'Observed
      W/Omitted'  @86 'Observed  W/Omitted'
      @108 'Observed  W/Omitted' /
      @1 '-----------------' @20 '--------  ---------'  @42 '--------  ---------'
      @64 '--------  ---------'  @86 '--------  ---------'
      @108 '--------  ---------';
      run;
```

MACRO EXECUTION

The following is a sample call to the MIXED_OVA macro, using a subset of the High School and Beyond data as a sample data file with 7,185 level-1 units nested within 160 level-2 (school) units. The original random effects model includes two predictors at level-1 (female and ses) and two predictors at level-2 (meanses and size), an interaction between female and ses, and random effects for intercepts and ses slopes. The level-2 identifier is schoolid, and the Kenward-Roger method for calculating degrees of freedom has been requested. The blank values associated with r_x and r_z inform the macro to use the average correlations at both levels, and the gamma parameter set to .5 represents our perceived association between the omitted variable and the dependent variable. The continuous omitted variable has occurred at level-1 and, finally, the MIXED_OVA macro will conduct 500 simulation runs to generate the outcomes of interest. We recommend that at least 500 simulations runs be used to generate results with ample precision.

```
%mixed_ova(path=C:\Stuff\USC\SESUG\2011\OVA\,
      data=hsb,
      dv=mathach,
      lvl1iv=female ses,
      lvl2iv=meanses size,
      interact=female*meanses,
```

```
random=intercept ses,
lvl2id=schoolid,
ddfm=kenwardroger,
cov=vc,
r_x=,
r_z=,
gamma=.5,
alpha=.05,
omitlvl=1,
sim_n=500);
```

Figures 4 and 5 below show the output resulting from the execution of the MIXED_OVA macro. The first output summarizes the observed outcome and estimated bias values for fixed effects included in the original model, including parameter estimates, standard errors, degrees of freedom, *t*-values and associated *p*-values. Figure 5 depicts the output associated with the random effects. In both figures, we can see non-trivial bias values for parameter estimates and other outcomes, though little change in the calculated *p*-values associated with the *t* and *z*-tests. Our use of a subset of the High School and Beyond data may have prevented us from seeing much change in the calculated p-values due to the large size of our sample.

Omitted Variable Analysis
Fixed Effects

Gamma Value = .5
Level-1 Correlation = -0.051142594
Omission Level = 1
Number of Simulations = 500

	Parameter Estimate		Standard Error		Degrees of Freedom		t-value		Probability < |t|	
Effect	Observed	Bias	Observed	Bias	Observed	Bias	Observed	Bias	Observed	Bias
Intercept	13.35131	-0.51658	0.312888	-0.06321	178.9486	-0.39406	42.67126	0.079884	<.000001	0
female	-1.17007	0.459561	0.163117	-0.02749	6123.924	-12.2059	-7.17318	12.1701	<.000001	0
ses	2.00019	0.334392	0.159169	-0.02201	417.2422	-340.388	12.56647	2.283262	<.000001	0
meanses	3.687554	0.105135	0.378443	-0.0758	181.6335	-0.3539	9.744006	0.063171	<.000001	-0.006
size	-0.00007	0.000265	0.00024	-0.00005	158.0311	0.01192	-0.3034	0.247076	0.761987	-0.008
female*ses	0.295632	-0.00504	0.19836	-0.03061	2766.019	-131.005	1.490379	-0.03358	0.136239	0.058

Figure 2. Snapshot of MIXED_OVA fixed effect bias output.

Omitted Variable Analysis
Random Effects

Gamma Value = .5
Level-1 Correlation = -0.051142594
Omission Level = 1
Number of Simulations = 500

	Parameter Estimate		Standard Error		Z-value		Probability < |Z|	
Effect	Observed	Bias	Observed	Bias	Observed	Bias	Observed	Bias
Intercept	2.546616	-6.19964	0.395263	-0.74904	6.442846	0.030802	<.000001	0
ses	0.444311	-0.00179	0.227583	-0.09518	1.952303	0.272818	0.025451	0
Residual	36.56814	-17.5177	0.623664	-0.29754	58.6344	-0.04544	<.000001	0

Figure 3. Snapshot of MIXED_OVA random effect bias output.

CONCLUSION

The macro presented in this paper provides the user with the ability to estimate the level of bias present in fixed effects and variance components generated from a two-level, linear multilevel model omitting a known continuous variable of interest. By including estimates of the correlation between the omitted and measured variables, as well as the purported strength of the relationship between the omitted and dependent variable represented by a regression coefficient, the macro augments the original data structure and runs the same multilevel model with the omitted variable included, yielding updated fixed effect and variance component results for bias estimation. In addition, the MIXED_OVA macro allows the user to specify the level at which the omitted variable occurs, based on the knowledge that omitted variables at lower levels of the model may produce greater bias than omitted variables at higher levels (Kim & Frees, 2005, 2006). The authors acknowledge the limitations to the macro previously mentioned, but will continue to develop and make available newer version of the MIXED_OVA with expanding capabilities. In summary, the MIXED_OVA macro provides the user with a succinct summary of the potential bias they could expect given the parameters of model misspecification they supply in the macro arguments.

REFERENCES

Box, G. E. P. (1966). The use and abuse of regression. *Technometrics, 8,* 625-629.

Ebbes, P., Böckenholt, U., & Wedel, M. (2004). Regressor and random-effects dependencies in multilevel models. *Statistica Neerlandica*, *58*, 161-178.

Kim, J.-S. & Frees, E. W. (2005). Fixed effects estimation in multilevel models. University of Wisconsin working paper.

Kim, J.-S. & Frees, E. W. (2006). Omitted variables in multilevel models. *Psychometrika*, *71*, 659-690.

Marcus, S. M. (1997). Using omitted variable bias to assess uncertainty in the estimation of an AIDS education treatment effect. *Journal of Educational and Behavioral Statistics*, *22*, 193-201.

Mosteller, F., & Tukey, J. (1977). *Data analysis and regression.* Reading, MA: Addison-Wesley.

Rosenbaum, P. R. (1986). Dropping out of high school in the United States: An observational study. *Journal of Educational Statistics, 11,* 207-224.

Rosenbaum, P. R. (1991). Discussing hidden bias in observational studies. *Annals of Internal Medicine, 115,* 901-905.

Rosenbaum, P. R., & Rubin, D. B. (1983). Assessing sensitivity to an unobserved binary covariate in an observational study with binary outcome. *Journal of the Royal Statistical Society, B, 45,* 212-218.

Rubin, D. B. (1978). Bayesian inference for causal effects: The role of randomization. *Annals of Statistics, 6,* 34—58.

CONTACT INFORMATION

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

Jason Schoeneberger
University of South Carolina
14726 Provence Lane
Charlotte, NC 28277
Phone: 704-307-9395
E-mail: schoeneb@email.sc.edu