Paper 139-2007

GLMM_SIM: A SAS® Macro for Evaluating the Statistical Integrity of General Linear Mixed Models

James T. Coraggio, University of South Florida, Tampa, FL
Jeffrey D. Kromrey, University of South Florida, Tampa, FL
Ha T. Phan, University of South Florida, Tampa, FL
Jeanine L. Romano, University of Tampa, Tampa, FL
Melinda R. Hess, University of South Florida, Tampa, FL
Reginald S. Lee, University of South Florida, Tampa, FL
Constance V. Hines, University of South Florida, Tampa, FL
Stephen L. Luther, University of South Florida, Tampa, FL

ABSTRACT

Accommodating hierarchical or nested data has historically been problematic for researchers. General linear mixed models have the ability to account for nested data overcoming some of the limitations of the more traditional models. These advantages have made mixed modeling an increasingly popular statistical method, but questions still remain regarding the integrity of the model (e.g., power and sample size, the impact of measurement error, and the effect of multicollinearity). Programming was created to address these issues through simulation. This paper presents a SAS simulation program that creates samples of multilevel data and allows specification of sample size, number of regressors, reliability of regressors, and correlation between regressors. The program constructs multilevel models using simulated data as inputs, and outputs the parameter estimates and probabilities obtained from PROC MIXED. The resulting parameter estimates are then evaluated for accuracy. The paper provides a demonstration of SAS/IML, macro programming language, and recent examples of applications in simulation studies.

INTRODUCTION

Often data structures are of sufficient complexity to challenge the robustness and adequacy of many of the more common and traditional statistical techniques employed in research (e.g., ANOVA, Regression). The limitations of these techniques require the advancement of a more sophisticated method to analyze data and provide the most accurate and precise estimates possible. With the continuing expansion of the availability of technology and software designed to handle more complex analyses, research into these methods to identify conditions under which they outperform other methods, as well as conditions that may be problematic for a specific analyses is essential. Multilevel modeling, also called hierarchical linear modeling or mixed modeling, is one such technique. This analytic method has become more prevalent in applied research analyses as it takes into account the nested structure present in many research designs and helps overcome issues detrimental to the function of traditional methods (e.g., lack of independence, unbalanced designs). Theoretically, the number of levels in multilevel models may be quite large, although most current research, both methodological and applied, consists of 2-level or 3-level models. For example, in medical applications, an example of a 2-level design might be patients nested in clinics. In education, a 3-level design might consist of students nested in classes and classes nested in schools. As the popularity of this analytical approach grows with methodological and applied researchers, the amount of reference and instructional material is becoming increasingly available (see, for example, Brown & Prescott, 2006; Bingenheimer & Raudenbush, 2004; Raudenbush & Bryk 2002; McCulloch & Searle, 2001). However, as these methods become better known and used in research, issues that must be addressed when employing these techniques are also being identified. As such, further research into the strengths and limitations of multilevel modeling is needed.

Current research into multilevel modeling methodological issues includes sample size (Giesbrecht & Burns, 1985, Hess, Ferron, Bell-Ellison, Dedrick, & Lewis, 2006; Kenward & Roger, 1997; Maas & Hox, 2002), model misspecification (Donoghue & Jenkins, 1992; Ferron, Dailey, & Yi, 2002, Schwartz, 1978), measurement error (Kromrey, Coraggio, Phan, Romano, Hess, Lee, Hines, & Luther, 2007), and missing data (Basilevsky, Sabourin, Hum, & Anderson, 1985; Raymond & Roberts, 1987; Roy & Lin, 2002). The complexity and increased use of multilevel methods calls for further investigation into the behavior and robustness of the methods to data structures with challenging or complex characteristics. Some of the areas that need further investigation include issues of power and sample size, the impact of measurement error, and effects of multicollinearity.

To facilitate this focus of multilevel modeling research, a SAS macro simulating samples from a 2-level structure was developed that analyzes each sample with PROC MIXED and aggregates results across samples. The macro provides a simple way to use the power of Monte Carlo simulation techniques to investigate issues such as statistical bias, confidence interval coverage and width, and statistical power in hypothesis tests used in multilevel modeling. The macro may also be used for planning research studies by providing information about sample size requirements to obtain a desired level of precision in statistical estimates.

MACRO GLMM_SIM

In order to simulate data for multilevel modeling investigation, 15 variables were established that control the nature of the population from which samples are drawn. The variables are supplied as arguments when the macro is called:

1. The variance in intercepts (_tau0)
2. The variance in slopes (_tau1)
3. The fixed effects for intercept (_gamma0)
4. The fixed effects for slopes (_gamma1)
5. The number of level 2 units (_n2)
6. The minimum number of level 1 units (_n1min)
7. The maximum number of level 1 units (_n1max)
8. The number of level 1 predictors (_k1)
9. The number of level 2 predictors (_k2)
10. The correlation between level 1 predictors (_r1)
11. The correlation between level 2 predictors (_r2)
12. The reliability of predictors (_r_xx)
13. The number of replications to simulate (_total_reps)
14. dummy variable generates a column of zeros (error e1), one predictor measured without error (_fal)
15. A dummy variable that sets one of the fixed effects to zero, providing an estimate of Type I error rate (_zero_end)

Default values of each argument are provided in the macro definition, so that each call to the macro only needs to override selected defaults. This allows specific sequences of conditions to be simulated. For example, a researcher interested in determining the impact of measurement error on parameter estimates may invoke the macro three times, with each invocation providing a different value for regressor reliability (_r_xx).

To provide an overview of the macro's operations, consider a multilevel model in educational research that consists of students nested within classrooms. Within each of N_2 classrooms, student outcomes (Y) are modeled with k_1 student-level predictors (in this example, $k_1 = 3$), so that within the j^{th} classroom we can think of:

$$y_{ij} = b_{0j} + b_{1j}X_1 + b_{2j}X_2 + b_{3j}X_3 + \varepsilon_{ij}$$

In the macro, this equation is represented by the matrix equation:

Y = X*B_i` + r;

where **X** is the matrix of true scores for the N_1 students in classroom j (with a leading column of ones for the intercept), **B_i** is the vector of level 1 equation parameters, and **r** is the vector of residuals.

The parameters of this equation are modeled as a function of k_2 classroom-level predictors (here, $k_2 = 2$):

$$\begin{aligned} b_{0j} &= \gamma_{00} + \gamma_{01}W_1 + \gamma_{02}W_2 + U_{0j} \\ b_{1j} &= \gamma_{10} + \gamma_{11}W_1 + \gamma_{12}W_2 + U_{1j} \\ b_{2j} &= \gamma_{20} + \gamma_{21}W_1 + \gamma_{22}W_2 + U_{2j} \\ b_{3j} &= \gamma_{30} + \gamma_{31}W_1 + \gamma_{32}W_2 + U_{3j} \end{aligned}$$

In the macro, this equation is represented in the matrix equation:

B_i = W[ID,]*gamma_mtx + U[ID,];

where **W**[ID,] is the vector of true scores on W_1 and W_2 for the j^{th} classroom (with a leading element of 1 for the intercept), **U**[ID,] is the vector of level 2 residuals for the j^{th} classroom, and the gamma coefficients are all assembled in gamma_mtx:

$$\text{gamma\_mtx} = \begin{bmatrix} \gamma_{00} & \gamma_{10} & \gamma_{20} & \gamma_{30} \\ \gamma_{01} & \gamma_{11} & \gamma_{21} & \gamma_{31} \\ \gamma_{02} & \gamma_{12} & \gamma_{22} & \gamma_{32} \end{bmatrix}$$

Assembling these equations in matrices allows the program to automatically adjust the size of these equations as we change the number of predictors at level 1 or level 2 (k_1 and k_2).

Substituting the level 2 equations into the level 1 equation provides the full mixed model:

$$Y_{ij} = \gamma_{00} + \gamma_{01}W_1 + \gamma_{02}W_2 + \gamma_{10}X_1 + \gamma_{11}W_1X_1 + \gamma_{12}W_2X_1 +$$
$$\gamma_{20}X_2 + \gamma_{21}W_1X_2 + \gamma_{22}W_2X_2 + \gamma_{30}X_3 + \gamma_{31}W_1X_3 + \gamma_{32}W_2X_3 +$$
$$U_{0j} + U_{1j} + U_{2j} + U_{3j} + \varepsilon_{ij}$$

The parameter estimates from this equation are in the PROC MIXED output (the 12 gamma coefficients are in the 'solutions for fixed effects' and the estimated variances of the four U's are in the main diagonal of the 'covariance parameter estimates'). These parameter estimates represent the outcomes of interest for statistical investigations of mixed models.

In order to analyze the effects of fallible regressors on model design, measurement error may be introduced into the level 1 and level 2 predictors via the _r_xx argument to the macro. The value of this argument is used to calculate the error variance:

```
error_var = (1-r_xx)/r_xx;
```

The measurement error is then transformed into a vector of error variances for the level 1 and level 2 predictors using the following syntax.

```
pop_var1_e1 = J(1,k1,error_var);
pop_var2_e2 = J(1,k2,error_var);
```

Individual observations' error scores are then created using the data generation procedure (the gendata subroutine) along with the vector of error variances. Two error score matrices are created (E1 and E2).

```
run gendata(N1,RN_seed,pop_var1_e1,1,0,0,pop_mn1_e1,rmatrix1_e1,E1);
E1 = J(N1,1,1)||E1;
run gendata(N2,RN_seed,pop_var2_e2,1,0,0,pop_mn2_e2,rmatrix2_e2,E2);
E2 = J(N2,1,1)||E2;
```

These "error" score matrices are combined with the "true" score matrices for the level 1 and level 2 predictor values (to create the "observed score" matrices) as shown in the following syntax.

```
X[,2:k1+1]+E1[,2:K1+1]
W_out = repeat(W[ID,]+E2[ID,],N1);
```

The final assembled "observed" score matrices are then sent to regular SAS and the PROC MIXED procedure, through which each sample is analyzed.

The resulting values for the 'solutions for fixed effects' and the 'covariance parameter estimates' produced from the PROC MIXED procedure are then output to new SAS data sets (using the ODS command in MIXED). Because some data conditions may lead to convergence problems in the estimation of mixed models, the program checks for non-converging samples and makes a second pass through PROC MIXED, using "looser" convergence criteria options (maxfunc = 500, scoring = 5, and convh = .002).

The parameter estimates are compared to the original parameters used for generating the "true" score data to estimate statistical bias, root mean square error (RMSE), confidence interval coverage and width, and hypothesis test rejection rates (either Type I error rate or statistical power). The resulting "Fixed Effects Results" table and "Random Effects Results" table for each run are then written to the SAS output window with FILE PRINT statements. The tables also include the number of samples that did not converge after the first and second pass through the PROC MIXED procedure.

The SAS program includes other programming elements such as a procedure to temporarily reroute the log and output windows (PROC PRINTTO) in order to prevent overflow, the creation of text files to temporarily house macro generated RENAME and MODEL statements, and a subroutine to generate non-normal random data using Fleishman constants.

A second version of the macro is available that uses the mean level 1 predictor values as the level 2 predictor values. This version of the program was used to simulate instances of this data structure used in some educational research. This version of the macro is available from the authors.

EXAMPLE OF MACRO GLMM_SIM

Use of the macro is illustrated in planning a study to investigate the effect of teacher-reported practice and student perceptions of teachers' classroom practices on student science achievement measured by a performance task assessment. A pilot study was conducted to obtain estimates of the correlations among predictors and the reliability of predictors. With anticipated classroom sizes (level 1) ranging from 15 to 30 students, the researchers were interested in the number of classrooms needed to achieve power of at least .80 in the tests of the model coefficients. The macro was then invoked specifying 15 classrooms:

```
%GLMM_SIM1 (_tau0=1.75, _tau1=.5,_N2=15,_gamma0=1, _gamma1=0.5, _n1min=15, _n1max=30, _k1 = 2, _k2 =
4,  _r1 = 0.4, _r2 = 0.354, _r_xx = 0.60, _total_reps = 1000, _fal = 0, _zero_end= 0);
```

The macro output from this invocation is presented in Table 1.

As is evident in the output, none of the tests for the fixed effects using 15 classrooms (i.e., N_2=15) provide sufficient statistical power. The macro was subsequently run with 30, 50, 60, 80 100, and 120 classrooms specified. Figure 1 summarizes these results in a line graph. The effect rejection rates at each level were averaged (i.e., w_1, w_2, w_3, w_4) for each invocation of the macro and plotted as a function of the number of level-2 units. Results suggest that approximately 43 classrooms are needed to achieve power of .80 for tests of the level 2 coefficients in the design specified. A substantially larger sample of classrooms (approximately 71) will be needed to achieve this level of power for the level 1 tests.

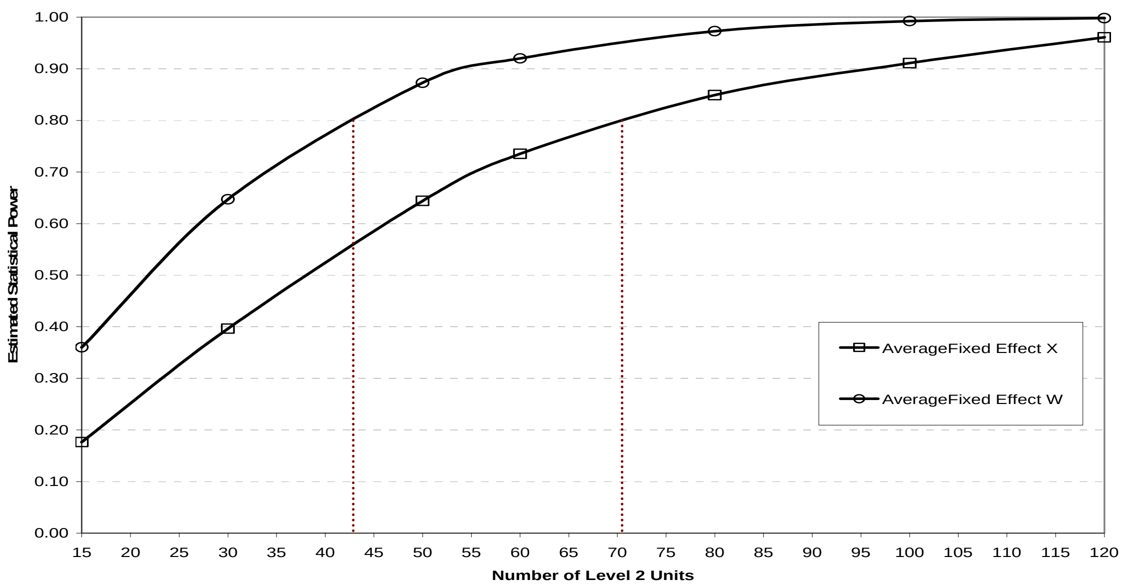

Figure 1. Estimated Statistical Power as a Function of the Number of Level 2 Units

CONCLUSION

The macro GLMM_SIM is provided to facilitate statistical investigation of mixed models. Specifically, this macro uses Monte Carlo simulation techniques to investigate issues such as statistical bias, confidence interval coverage and width, and statistical power in hypothesis tests used in multilevel modeling. The macro creates samples from a 2-level structure, analyzes each sample with PROC MIXED, and aggregates results across samples. Simply changing the arguments sent to the macro (i.e., sample size, number of level-1 and level-2 predictors, and reliability of predictors) will allow for the impact of measurement errors on mixed model to be estimated. When convergence problems in estimation of parameters are encountered, the macro will relax the convergence criteria and re-estimate parameters of the mixed models. The macro also includes a procedure to temporarily reroute the log and output windows in order to prevent overflow, and a subroutine to generate random data using Fleishman constants. This macro may be used for planning research studies by providing information about sample size requirements to obtain a desired level of precision in statistical estimates.

Table 1
Sample Output from Macro.

Multilevel Data Simulation: Fixed Effects

n1 (min):	15
n1 (max):	30
n2:	15
k1:	2
k2:	4
r1:	0.40
r2:	0.35
rxx:	0.60
True Value of Gamma0:	1.00
True Value of Gamma1:	0.50
Number of Samples Simulated:	1000
Non-convergence (default):	19
Non-convergence (loose criteria):	0

Multilevel Data Simulation: Random Effects

n1 (min):	15
n1 (max):	30
n2:	15
k1:	2
k2:	4
r1:	0.40
r2:	0.35
rxx:	0.60
True Value of Tau0:	1.75
True Value of Tau1:	0.50
Number of Samples Simulated:	1000
Non-convergence (default):	19
Non-convergence (loose criteria):	0

				95% Confidence Interval		
Effect	Mean	Bias	RMSE	Coverage	Width	Rejection Rate
Intercept	0.9889084423	-0.011091558	0.6489726705	0.9439439439	2.6420555087	0.3263263263
w1	0.7319353581	-0.268064642	0.6173674442	0.8748748749	2.0127376556	0.3493493493
w2	0.771543317	-0.228456683	0.5926445917	0.8898898899	2.0136331464	0.3613613614
w3	0.7801955554	-0.219804445	0.5725400806	0.8958958959	1.9927440979	0.3703703704
w4	0.7400906019	-0.259909398	0.6219637644	0.8758758759	2.0068033284	0.3743743744
x1	0.3390687436	-0.160931256	0.3433600535	0.9189189189	1.3149194412	0.1801801802
x1*w1	0.254128359	-0.245871641	0.3687212827	0.7847847848	1.0010364372	0.1991991992
x1*w2	0.2525945408	-0.247405459	0.362935455	0.7857857858	1.0000283482	0.2122122122
x1*w3	0.2637130782	-0.236286922	0.3554942048	0.7917917918	0.9882864804	0.2312312312
x1*w4	0.2624962407	-0.237503759	0.3575834812	0.7967967968	0.9971138894	0.2182182182
x2	0.3344500463	-0.165549954	0.3674779218	0.8958958959	1.3019061594	0.2012012012
x2*w1	0.2638527322	-0.236147268	0.3622693585	0.7887887888	0.9923249294	0.2352352352
x2*w2	0.2516766714	-0.248323329	0.3694218206	0.7757757758	0.9964890857	0.2442442442
x2*w3	0.2605097038	-0.239490296	0.3697328524	0.7637637638	0.9789097694	0.2502502503
x2*w4	0.2489074449	-0.251092555	0.3693337801	0.7717717718	0.9888628601	0.2232232232

				95% Confidence Interval		
CovParm	Mean	Bias	RMSE	Coverage	Width	Rejection Rate
UN(1,1)	3.6776876121	1.9276876121	2.6277451786	0.5720000000	10.708676272	0.9940000000
UN(2,2)	0.8144789698	0.3144789698	0.5536610375	0.7780000000	4.1757408905	0.8180000000
UN(3,3)	0.7999602756	0.2999602756	0.5357094099	0.7910000000	3.3095187693	0.8160000000

REFERENCES

Basilevsky, A., Sabourin, D., Hum, D., & Anderson, A. (1985). Missing data estimators in the general linear model: An evaluation of simulated data as an experimental design. C*ommunications in Statistics*, *14*, 371-394.

Bingenheimer, J. B. & Raudenbush, S. W. (2004). Statistical and substantive inferences in public health: Issues in the application of multilevel models. *Annual Review of Public Health*, *25*, 53–77.

Brown, H. & Prescott, R. (2006). *Applied Mixed Models in Medicine* (2nd ed.). John Wiley & Sons, LTD: New York.

Donoghue, J.R., & Jenkins, F. (1992). *A Monte Carlo study of the effects of model misspecification on HLM estimates*. Technical report, Educational Testing Service, Princeton, NJ. November 1992.

Ferron, J., Dailey, R., & Yi, Q. (2002). Effects of misspecifying the first-level error structure in two-level models of change. *Multivariate Behavioral Research*, *37*, 379-403.

Giesbrecht, F., & Burns, J. (1985). Two-stage analysis based on a mixed model: Large-sample asymptotic theory and small-sample simulation results. *Biometrics*, *41*, 477-486.

Hess, M.R., Ferron, J.M, Bell-Ellison, B.A., Dedrick, R.F., & Lewis, S.E. (2006, April). *The effects of sample size on fixed effect inferences*. Paper presented at the American Educational Association's Annual Meeting, San Francisco, CA.

Kenward, M., & Roger, J. (1997). Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, *53*, 983-997.

Kromrey, J. D., Coraggio, J. T., Phan, H., Romano, J. T., Hess, M. R., Lee, R. S., Hines, C. V., and Luther, S. L. (2007, April). *Fallible regressors in multilevel models: Impact of measurement error on parameter estimates and hypothesis tests*. Paper presented at the American Educational Association's Annual Meeting, Chicago.

Maas, C.J.M. & Hox, J.J. (2002). Sample sizes for multilevel modeling. In: J. Blasius, J. Hox , E. de Leeuw & P. Schmidt (eds.) (2002). *Social Science Methodology in the New Millennium. Proceedings of the Fifth International Conference on Logic and Methodology*. Second expanded edition. Opladen, RG: Leske + Budrich Verlag (CD-ROM).

McCulloch, C. E. & Searle, S. R. (2001). *Generalized, linear, and mixed models*. New York: Wiley

Raudenbush S.W. & Bryk, A.S. (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods* (2nd ed.). Sage Publications: Thousand Oaks, CA.

Raymond, M. R., & Roberts, D. M. (1987). A comparison of methods for treating incomplete data in selection research. *Educational and Psychological Measurement*, *47*, 13-26.

Roy, J., & Lin, X. (2002). Analysis of multivariate longitudinal outcomes with nonignorable dropouts and missing covariates: Changes in methadone treatment practices. *Journal of the American Statistical Association, 97*, 40-52.

Schwartz, G. (1978). Estimating the dimensions of a model. *Annals of Statistics*, *6*, 461-464.

CONTACT INFORMATION

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

James T. Coraggio
University of South Florida
Department of Educational Measurement and Research
4202 E. Fowler Ave., EDU 162
Tampa, FL 33620
(813) 974-4933
Fax: (813) 974-4495
coraggio@tempest.coedu.usf.edu
http://www.coedu.usf.edu/main/departments/me/me.html

GLMM_SIM1 MACRO

```
%macro GLMM_SIM1 (_tau0=.1, _tau1=.1,_N2=120,_gamma0=1, _gamma1=0.5,
_n1min=20, _n1max=100, _k1 = 3, _k2 = 2, _r1 = 0.0, _r2 = 0.0, _r_xx = 1.00, _total_reps
= 1000, _fal = 0, _zero_end= 0);
* +-------------------------------------------------------------------------------------------------------------+
  Arguments to the macro
  _tau0= variance in intercepts,  _tau1= variance in slopes
  _n2= number of classes,  _gamma0= fixed effects for intercept
  _gamma1= fixed effects for slopes,  _n1min= minimum students in class
  _n1max= maximum students in class, _k1 = number of level 1 predictors
  _k2 = number of level 2 predictors, _r1 = correlation between level 1 predictors
  _r2 = correlation between level 2 predictors,  _r_xx = reliability of predictors
  _total_reps = Number of replications to simulate,
  _fal = generate a column of zeros in the error e1 matrix
  _zero_end= set last element of the last row to zero: W2 has no effect on X3
* +-------------------------------------------------------------------------------------------------------------+;

* +-------------------------------------------------------------------+
   Create file with appropriate RENAME commands
  +-------------------------------------------------------------------+;
data _null_;
        file rname;
        %do count = 1 %to &_k1;
          %let a = %eval(&count+16);
          %let state = "Rename col&&a = x&&count %STR(;)";  put &state;
        %end;
        %do count = 1 %to &_k2;
          %let a = %eval(&count+&_k1+16);
          %let state = "Rename col&&a = w&&count %STR(;)";  put &state;
        %end; stop;  run;
* +-------------------------------------------------------------------------------------+
   Create file with appropriate MODEL and RANDOM statements
  +-------------------------------------------------------------------------------------+;
data _null_;
        file mixed;
        put "model y = " @;
        %do count = 1 %to &_k1;
                %let state = "x&&count ";  put &state @;
        %end;
        %do count = 1 %to &_k2;
                %let state = "w&&count "; put &state @;
        %end;
        %do count1 = 1 %to &_k1;
                %do count2 = 1 %to &_k2;
                        %let state = "w&&count2%STR(*)x&&count1 ";
                        put &state @;
                %end;
        %end;
        put "/ s cl;";
        put "random int " @;
        %do count = 1 %to &_k1;
                %let state = "x&&count "; put &state @;
        %end;
        put "/ sub = ID2 type = un;";
        stop; run;
proc iml;
* +-------------------------------------------------------------------------------------------------------------+
  Subroutine to generate a random sample.
  User specifies the population means and standard deviations, as well as the correlation
  matrix. For population shapes, Fleishman constants are used.

  Inputs to the subroutine are
   NN - desired sample size, mu - row vector of population means
   variance - row vector of population variances, bb,cc,dd - Fleishman constants
   r_matrix - population correlation matrix

  Outputs are
     Rawdata - matrix of NN observations from the specified population
 +-------------------------------------------------------------------------------------------------------------+;
start gendata(NN,seed1,variance,bb,cc,dd,mu,r_matrix,rawdata);
 COLS = NCOL(r_matrix); G = ROOT(r_matrix); rawdata=rannor(repeat(seed1,nn,COLS));
 do r = 1 to NN;
  do c = 1 to COLS;
   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;
 rawdata = rawdata*G;
finish;
* Definition of condition to be simulated;
tau0=&_tau0; tau1=&_tau1;  n2=&_n2;   gamma0=&_gamma0;   gamma1=&_gamma1;
n1min=&_n1min;  n1max=&_n1max;
k1 = &_k1; k2 = &_k2; r1 = &_r1;  r2 = &_r2;  r_xx = &_r_xx;  total_reps = &_total_Reps; fal
= &_fal;  zero_end= &_zero_end;
RN_seed = round(ranuni(0)#100000000);
* +-------------------------------------------------------------------------------------------------------------+
 Compute error variance to give specified reliability (assume that true score variance = 1.0
 +-------------------------------------------------------------------------------------------------------------+;
 error_var = (1-r_xx)/r_xx;
* +-------------------------------------------------------------------------------------+
   Create k1 X k1 correlation matrix for level 1 data generation (error)
 +-------------------------------------------------------------------------------------+;
 rmatrix1_e1 = J(k1,k1,0);
 do element = 1 to k1;  rmatrix1_e1[element,element] = 1;   end;
* +---------------------------------------------------------------------------------------------+
   Create k1 X k1 correlation matrix for level 1 data generation (true scores)
 +---------------------------------------------------------------------------------------------+;
 rmatrix1 = J(k1,k1,r1);
 do element = 1 to k1; rmatrix1[element,element] = 1;   end;
```

```
* +--------------------------------------------------------------------------------------------+
    Create k2 X k2 correlation matrix for level 2 data generation (error)
  +--------------------------------------------------------------------------------------------+;
  rmatrix2_e2 = J(k2,k2,0);
  do element = 1 to k2; rmatrix2_e2[element,element] = 1;   end;
* +--------------------------------------------------------------------------------------------+
    Create k2 X k2 correlation matrix for level 2 data generation (true scores)
  +--------------------------------------------------------------------------------------------+;
  rmatrix2 = J(k2,k2,r2);
  do element = 1 to k2;  rmatrix2[element,element] = 1;   end;
* +--------------------------------------------------------------------------------------------+
    Generate correlation matrix for simulation of level 2 residuals (U0,U1,...Uk1)
  +--------------------------------------------------------------------------------------------+;
  resid_r = J(k1+1,k1+1,0);
  do element = 1 to k1+1; resid_r[element,element] = 1;  end;
  pop_varr = J(1,k1+1,tau0);   pop_mnr = J(1,k1+1,0);

do replicat = 1 to total_reps;
* +------------------------------------------------------------------+
    Generate sample of level 2 measurement errors
  +------------------------------------------------------------------+;
  pop_var2_e2 = J(1,k2,error_var);* vector of variances of measurement errors;
  pop_mn2_e2 = J(1,k2,0);*vector of means of measurement errors;

  run gendata(N2,RN_seed,pop_var2_e2,1,0,0,pop_mn2_e2,rmatrix2_e2,E2);
  E2 = J(N2,1,1)||E2;
* +-------------------------------------------------------+
    Generate sample of level 2 true scores
  +-------------------------------------------------------+;
  pop_var2 = J(1,k2,1);   pop_mn2 = J(1,k2,0);
  run gendata(N2,RN_seed,pop_var2,1,0,0,pop_mn2,rmatrix2,W);
  W = J(N2,1,1)||W;
* +----------------------------------------------------------------------+
    Generate matrix of level 2 residuals (U0,U1,...Uk1)
  +----------------------------------------------------------------------+;
  run gendata(N2,RN_seed,pop_varr,1,0,0,pop_mnr,resid_r,U);
  pop_var1 = J(1,k1,1);   pop_mn1 = J(1,k1,0);
  pop_var1_e1 = J(1,k1,error_var);* vector of variances of measurement errors;
  pop_mn1_e1 = J(1,k1,0);*vector of means of measurement errors;
 do ID=1 to n2;
   n1range=n1max-n1min;*sample size within classroom;
   n1=n1min+round(uniform(0)*n1range);
   IDlevel2=j(n1,1,ID);
* +--------------------------------------------------------------------------------------+
    Generate sample of level 1 measurement errors in classroom ID
  +--------------------------------------------------------------------------------------+;
   run gendata(N1,RN_seed,pop_var1_e1,1,0,0,pop_mn1_e1,rmatrix1_e1,E1);
   E1 = J(N1,1,1)||E1;
   if fal = 1 then do;  E1[,K1+1] =J(N1,1,0); end;
* +---------------------------------------------------------------------------+
    Generate sample of true level 1 values in classroom ID
  +---------------------------------------------------------------------------+;
  run gendata(N1,RN_seed,pop_var1,1,0,0,pop_mn1,rmatrix1,X);
  X = J(N1,1,1)||X;
* +---------------------------------------------------------------------+
    Generate sample of level 1 residuals in classroom ID
  +---------------------------------------------------------------------+;
  run gendata(N1,RN_seed,1,1,0,0,0,1,r);
* +---------------------------------------------------------------------+
    Generate level 1 regression equation for classroom ID
  +---------------------------------------------------------------------+;
  gamma_mtx = J(k2+1,1,gamma0)||J(k2+1,K1,gamma1);
if zero_end=1 then do;
    gamma_mtx[k2+1,K1+1] = 0;   * set last element of the last row to zero: W2 has no
effect on X3;  end;
  B_i = W[ID,]*gamma_mtx + U[ID,];
* +---------------------------------------------------------------+
    Generate observed Y values for classroom ID
  +---------------------------------------------------------------+;
  Y = X*B_i` + r;
* +---------------------------------------------------------+
     Assemble matrix to send to regular SAS
  +---------------------------------------------------------+;
  W_out = repeat(W[ID,]+E2[ID,],N1); replicat_out = repeat(replicat,N1);
  r_xx_out = repeat(r_xx,N1); k1_out = repeat(k1,N1); k2_out = repeat(k2,N1);
  tau0_out = repeat(tau0,N1); tau1_out = repeat(tau1,N1); N2_out = repeat(N2,N1);
  mean_n1 = (n1max+n1min)/2;  N1_out = repeat(mean_n1,N1);
  gamma0_out = repeat(gamma0,N1);gamma1_out = repeat(gamma1,N1);
  r1_out = repeat(r1,N1); r2_out = repeat(r2,N1);
  fal_out = repeat(fal,N1); zero_out =repeat(zero_end,N1);
  outmatrix =
                 replicat_out||r_xx_out||k1_out||k2_out||tau0_out||tau1_out||N2_out||N1_out||
                 gamma0_out||gamma1_out||r1_out||
                 r2_out||IDlevel2||y||fal_out||zero_out||X[,2:k1+1]+E1[,2:K1+1]||W_out[,2:k2+1
                 ];
   if (replicat = 1 & id = 1) then do; create j1 from outmatrix; end;
   append from outmatrix; end; end;
close j1;
data j1; set j1;
 rename col1 = replicat col2 = r_xx col3 = k1 col4 = k2 col5 = tau0 col6 = tau1 col7 = n2
            col8 = n1 col9 = gamma0 col10 = gamma1
            col11 = r1 col12 = r2 col13 = id2 col14 = y col15 =fal col16 =zero_end;
              %include rname;
proc mixed data = j1 CL COVTEST;
             by replicat n1 n2 r_xx k1 k2 fal zero_end r1 r2; class id2;
             %include mixed;  ods output SolutionF = output1 CovParms = output2;
*+-------------------------------------------------+
    Identify samples that did not converge
 +-------------------------------------------------+;
data nonconverge;
```

```
        merge output1 output2; by replicat; if effect ='' ; tryagain = 1;
proc means noprint data=nonconverge;
        by replicat;
        var tryagain;
        output out = try2 mean = tryagain;
data back;
        merge j1 try2; by replicat; if tryagain = 1;
* +-----------------------------------------------------------------------------------+
   Run PROC MIXED again with looser criteria for convergence
  +-----------------------------------------------------------------------------------+;
proc mixed data = back CL COVTEST maxfunc = 500 scoring = 5 convh = .002;
        by replicat n1 n2 r_xx k1 k2 fal zero_end r1 r2;  class id2;
        %include mixed;  ods output SolutionF = output1B CovParms = output2B;
run;
* +-----------------------------------------------------------------------------------------------------------+
   Count number of non-converging samples in first attempt with PROC MIXED
  +-----------------------------------------------------------------------------------------------------------+;
proc means noprint data=try2;
         var tryagain;
        output out = count1 n = noconverge1;
proc datasets; delete nonconverge try2;
*+-----------------------------------------------------------------------------------------------------+
   Identify samples that did not converge in second try with PROC MIXED
 +-----------------------------------------------------------------------------------------------------+;
data nonconverge;
        merge output1B output2B;  by replicat; if effect ='' ; tryagain = 1;
proc means noprint data=nonconverge; by replicat;
        var tryagain;
        output out = try2 mean = tryagain;
* +----------------------------------------------------------------------------------------------------------------+
   Count number of non-converging samples in second attempt with PROC MIXED
  +----------------------------------------------------------------------------------------------------------------+;
proc means noprint data=try2; var tryagain;
        output out = count2 n = noconverge2;
data count3a;
        merge count1 count2;
        if noconverge1 = . then noconverge1 = 0;
        if noconverge2 = . then noconverge2 = 0;
data count3c;  set count3a count3b;
proc means noprint data=count3c; var noconverge1 noconverge2;
        output out = count3 sum = noconverge1 noconverge2;
proc datasets;  delete nonconverge try2 count1 count2 count3a count3c;
*+-----------------------------------------------+
   Turn on the output window again
 +-----------------------------------------------+;
proc printto print = print;
proc sort data=output1; by replicat effect;
proc sort data=output1B; by replicat effect;
data fixed_effect_compare; merge output1 output1B;
        by replicat effect; if replicat = 0 then delete;
* +---------------------------------------------------------+
   Parameter values from data generation
  +---------------------------------------------------------+;
true_gamma0=&_gamma0;    true_gamma1=&_gamma1;
true_tau0=&_tau0;        true_tau1=&_tau1;
* +------------------------------------+
   Initialize counters to zero
  +------------------------------------+;
In_CL_gamma = 0;
Ho_test = 0;
If Effect = 'Intercept' or (INDEX(Effect,'w') ^=0 and INDEX(Effect,'*') = 0) then do;
        diff_gamma = Estimate - true_gamma0;
        diff_gamma_sq = diff_gamma * diff_gamma;
        if true_gamma0 > Lower and true_gamma0 < Upper then In_CL_gamma = 1;
end;
If INDEX(Effect,'x') ^=0 then do;
        diff_gamma = Estimate - true_gamma1;
        diff_gamma_sq = diff_gamma * diff_gamma;
        if true_gamma1 > Lower and true_gamma1 < Upper then In_CL_gamma = 1;
end;
CI_range_gamma = Upper - Lower;
if Probt < .05 then Ho_test = 1;
keep replicat n1 n2 r1 r2 r_xx k1 k2 fal zero_end Effect Estimate Lower Upper
    true_gamma0 true_gamma1 true_tau0 true_tau1 diff_gamma diff_gamma_sq
    In_Cl_gamma Ho_test Probt CI_range_gamma ;
proc sort data = fixed_effect_compare;   by Effect;
Proc univariate data=fixed_effect_compare noprint;  by Effect;
        var diff_gamma diff_gamma_sq In_Cl_gamma Ho_test CI_range_gamma
            estimate true_gamma0 true_gamma1 true_tau0 true_tau1 n1 n2 r_xx k1 k2
            fal zero_end r1 r2;
        output    out=fixed_effect_compare_out
                  mean=meandiff_gamma meandiff_gamma_sq CI_Coverage
                  rejection_rate meanCI_range estimate true_gamma0 true_gamma1
                  true_tau0 true_tau1 n1 n2 r_xx k1 k2 fal zero_end r1 r2
                  N=N_reps;
data fixed_effect_compare_out;
        if _n_ = 1 then set count3; retain noconverge1 noconverge2;
        set fixed_effect_compare_out;
        meandiff_gamma_sq = sqrt(meandiff_gamma_sq);
        n1min = &_n1min; n1max = &_n1max;
 file print header = h notitles;
    put @1 effect @20 estimate BEST12. @35 meandiff_gamma BEST12. @50
        meandiff_gamma_sq BEST12. @65 CI_coverage 12.10 @80 meanCI_range
        BEST12. @95 rejection_rate 12.10;  return;
        h: put @1 'Multilevel Data Simulation: Fixed Effects' /
            @1 '-----------------------------------------'//
        @1 'n1 (min):' @40 n1min 5. / @1 'n1 (max):' @40 n1max 5. /
        @1 'n2:' @40 n2 5. // @1 'k1:' @40 k1 5. / @1 'k2:' @40 k2 5. //
        @1 'r1:' @41 r1 4.2 / @1 'r2:' @41 r2 4.2 / @1 'rxx:' @41 r_xx 4.2 //
        @1 'True Value of Gamma0:' @41 true_gamma0 4.2 /
```

```
            @1 'True Value of Gamma1:' @41 true_gamma1 4.2 //
            @1 'Number of Samples Simulated:' @40 N_reps 5. /
            @1 'Non-convergence (default):' @40 noconverge1 5. /
            @1 'Non-convergence (loose criteria):' @40 noconverge2 5. //
            @1 '---------------------------------------------------------------------------------------------------' /
            @1 '                                                    95% Confidence Interval' /
            @1 '                                                  ---------------------------' /
            @1 'Effect          Mean        Bias        RMSE        Coverage        Width
            Rejection Rate' /
            @1 '---------------------------------------------------------------------------------------------------' ;
run;
proc sort data=output2; by replicat CovParm;
proc sort data=output2B; by replicat CovParm;
data variance_compare;
            merge output2 output2B; by replicat CovParm;
            if replicat = 0 then delete;
* +-------------------------------------------------------+
     Parameter values from data generation
  +-------------------------------------------------------+;
true_gamma0=&_gamma0;     true_gamma1=&_gamma1;
true_tau0=&_tau0;         true_tau1=&_tau1;
* +-----------------------------------+
     Initialize counters to zero
  +-----------------------------------+;
In_CL_tau = 0;
Ho_test = 0;
If CovParm = "UN(1,1)" then do;
            diff_tau = Estimate - true_tau0;
            diff_tau_sq = diff_tau * diff_tau;
            if true_tau0 > Lower and true_tau0 < Upper then In_CL_tau = 1; end;
If CovParm ^= "UN(1,1)" and SUBSTR(CovParm,4,1) = SUBSTR(CovParm,6,1) then do;
            diff_tau = Estimate - true_tau1;
            diff_tau_sq = diff_tau * diff_tau;
            if true_tau1 > Lower and true_tau1 < Upper then In_CL_tau = 1;
end;
If diff_tau = . then Delete;  * remove covariance terms;
CI_range_tau = Upper - Lower; if ProbZ < .05 then Ho_test = 1;
keep replicat CovParm Estimate Lower Upper true_tau0 true_tau1 true_gamma0
        true_gamma1 diff_tau diff_tau_sq In_Cl_tau Ho_test Probz CI_range_tau n1 n2 r_xx
        k1 k2 r1 r2 fal zero_end;
proc sort data = variance_compare; by CovParm;
Proc univariate data=variance_compare noprint; by CovParm;
            var estimate diff_tau diff_tau_sq In_Cl_tau Ho_test CI_range_tau true_tau0
                true_tau1 true_gamma0 true_gamma1 n1 n2 r_xx k1 k2 fal zero_end r1 r2;
            output out=variance_compare_out
                    mean=estimate meandiff_tau meandiff_tau_sq CI_coverage
                        rejection_rate meanCI_range_tau true_tau0 true_tau1
                        true_gamma0 true_gamma1 n1 n2 r_xx k1 k2 fal zero_end r1 r2
                    N = N_reps;
data variance_compare_out;
```

```
            if _n_ = 1 then set count3;  retain noconverge1 noconverge2;
            set variance_compare_out ;
            meandiff_tau_sq = sqrt(meandiff_tau_sq);
            n1min = &_n1min;   n1max = &_n1max;
 file print header = h notitles;
      put @1 CovParm @20 estimate BEST12. @35 meandiff_tau BEST12. @50
            meandiff_tau_sq BEST12. @65 CI_coverage 12.10 @80 meanCI_range_tau
            BEST12. @95 rejection_rate 12.10;
            return;
            h: put @1 'Multilevel Data Simulation: Random Effects' /
                @1 '-----------------------------------------' //
            @1 'n1 (min):' @40 n1min 5. / @1 'n1 (max):' @40 n1max 5. /
            @1 'n2:' @40 n2 5. //  @1 'k1:' @40 k1 5. /  @1 'k2:' @40 k2 5. //
            @1 'r1:' @41 r1 4.2 / @1 'r2:' @41 r2 4.2 / @1 'rxx:' @41 r_xx 4.2 //
            @1 'True Value of Tau0:' @41 true_tau0 4.2 /
            @1 'True Value of Tau1:' @41 true_tau1 4.2 //
            @1 'Number of Samples Simulated:' @40 N_reps 5. /
            @1 'Non-convergence (default):' @40 noconverge1 5. /
            @1 'Non-convergence (loose criteria):' @40 noconverge2 5. //
            @1 '---------------------------------------------------------------------------------------------------' /
            @1 '                                                    95% Confidence Interval' /
            @1 '                                                  ---------------------------' /
            @1 'CovParm          Mean        Bias        RMSE        Coverage        Width
            Rejection Rate' /
            @1 '---------------------------------------------------------------------------------------------------' ;
*+-------------------------------------------------------------------------+
    Erase all SAS datasets used in this execution of the macro
 +-------------------------------------------------------------------------+;
proc datasets;
 delete variance_compare variance_compare_out fixed_effect_compare
          fixed_effect_compare_out j1 output1 output2 back output1B output2B;
*+-----------------------------------------+
    Turn on the log window again
 +-----------------------------------------+;
proc printto log = log; run;
%mend GLMM_SIM1;
* +-------------------------------------------------------------------------------+
      Define 'dummy' files to reroute the log and/or output windows
  +-------------------------------------------------------------------------------+;
filename junk dummy;  filename junk2 dummy;
* +-------------------------------------------------------------------------------------------+
   Define disk locations for macro generated rename and model statements
  +-------------------------------------------------------------------------------------------+;
filename rname 'c:\temp1.txt';
filename mixed 'c:\temp2.txt';
* +----------------------------------+
    Sample call to the macro
  +----------------------------------+;
%GLMM_SIM1 (_r1 = 0.0, _r2 = 0.0); Run;
```