Using SAS for Statistical Modeling: MONTE CARLO Simulations

Sean W. Mulvenon, University of Arkansas, Fayetteville, AR
M. Austin Betz, Arizona State University, Tempe, AZ

Abstract

The development of new statistical formulae generally requires various mathematical derivations to clearly prove the accuracy and usefulness of the new formulae. However, an additional verification of the new formulae is also becoming standard and that is Monte Carlo simulation. Monte Carlo simulations provide a procedure for comparing new formulae versus current formulae and a forum to assess their relative effectiveness. The SAS® statistical computing software is very useful in this process and the current paper provides a model of how this can be completed. A series of new formulae for computing statistical power are generated and compared with a current procedure using Monte Carlo simulation. The procedures used in SAS for this process are PROC IML, PROC APPEND, PROC MEANS, and PROC ANOVA. The programs and statistical analyses are appropriate for all levels of SAS users and in particular those interested in statistical modeling.

Power Analysis

Muller and Barton (1989) recommended a procedure to calculate power estimates using an expected value of the estimator $\hat{\epsilon}$ to compute the degrees of freedom for a noncentral F. Following this procedure, the noncentral F (in terms of the Greenhouse-Geisser ϵ) becomes

$$F_{GG} \approx F[(ab)\epsilon, b(N - r)\epsilon, (ab)\epsilon F_A] \quad \textbf{(1)}$$

where $F_A = \dfrac{tr(H)}{tr(E)}$ computed using population values for μ and Σ, $a = g - 1$ (g represents the number of levels of the between group variable), $b = p - 1$ (p represents the number of repeated trials or levels of the within group variable), and $(ab)\epsilon F_A$ represents the noncentrality parameter, δ_ϵ, defined as

$$\delta_\varepsilon = (ab)\epsilon F_A. \quad \textbf{(2)}$$

An approximation for the power of the Greenhouse-Geisser test can be performed by computing the probability that a noncentral F with $(ab)\hat{\epsilon}$ and $b(N - r)\hat{\epsilon}$ degrees of freedom, and noncentrality $(ab)\hat{\epsilon}F_A$ exceeds the corresponding critical value (Muller & Barton, 1989). It should be noted that for the current study only $\hat{\epsilon}$ was used in the calculation of power estimates. Muller and Barton (1989) recommended using the Greenhouse-Geisser estimator $\hat{\epsilon}$ because the Huynh-Feldt estimator, $\tilde{\epsilon}$, was found to consistently overestimate the actual power. The expected value of $\hat{\epsilon}$ is calculated by solving the following equation:

$$E[f(\hat{\lambda})] \approx f(\lambda) + \frac{g_1}{(N - r)}, \quad \textbf{(3)}$$

with

$$g_1 = \Sigma f_{ii}\lambda_i^2 + \Sigma\Sigma f_i\lambda_i\lambda_j(\lambda_i - \lambda_j)^{-1}m_i m_j \quad \textbf{(4)}$$

where m_i is the multiplicity of the ith distinct eigenvalue, $f(\lambda) = \epsilon$, f_i is $\partial f / \partial\lambda_i$, and f_{ii} is $\partial f^{(2)} / \partial\lambda^{(2)}$, r is the rank of the matrix (equal to g), and $\lambda_i = \lambda_1, \lambda_2, \ldots, \lambda_b$, are the distinct ordered eigenvalues of Σ_* $(\Sigma_* = U' \cdot \Sigma \cdot U)$. The first and second derivatives with respect to λ_i are reported in Muller and Barton (1989).

In practice, population values of the variance covariance matrix will not be known. Therefore, it is necessary to use sample values generated from the data set under investigation. Equation 5 derived by Muller and Barton (1989) and equations 7, 8, and 10 derived by Betz and Thompson (1990) are analytic formulae which employ sample values for estimating power.

When the null hypothesis, H_0, is false, Muller and Barton (1989) proposed that power could be approximated using

$$1 - \beta = P\{F[(ab)\epsilon, b(N - r)\epsilon; \delta_\varepsilon] > F^{\alpha}[(ab)\epsilon, b(N - r)\epsilon]\} \quad \textbf{(5)}$$

where

$$\delta_{\varepsilon} = (ab)\varepsilon \cdot F_A = \frac{(b)\varepsilon \cdot tr(\Delta_*)}{tr(\Sigma_*)}. \quad \textbf{(6)}$$

Equation 6 represents the noncentrality parameter with population values replacing sample estimates, i.e., F_A, Δ_*, and Σ_* replace F_{univ}, $\hat{H}$ and $\hat{E}$, respectively. The power of a univariate mixed model F-test using 5 will be exact when the assumption of sphericity holds (ϵ = 1.0).

Betz and Thompson Power Formulae

The distribution of $y = tr(\hat{E})$, will be a linear combination of independent central χ^2 random variables, regardless of whether H_0 is true or false. However, when H_0 is false, $x = tr(\hat{H})$ is distributed as a linear combination of independent noncentral χ^2 random variables (Muller & Barton, 1989). The first of three new approximations (the proofs of the new approximations can be found in an unpublished masters thesis by Barry Thompson, Arizona State University, 1990) can be derived by approximating x with a multiple of a central χ^2 and produces

$$1 - \beta = P[F(h,h') > \frac{f_o h_o}{fh} F^{\alpha}(h_o,h')], \quad \textbf{(7)}$$

where $f = (m_x^* / l_x^*)$, $h = (l_x^* / f)$, $f' = (m_y / 2l_y)$, $h' = (l_y / f')$, and $f_o h_o = a \cdot tr(\hat{\Sigma}_*)$ are all constants defined as follows:

$$l_x^* = a \cdot tr(\Sigma_*) + tr(\Delta_*)$$
$$m_x^* = a \cdot tr(\Sigma_*) + 2 \cdot tr(\Sigma_*\Delta_*)$$
$$l_y = a(n - 1) \cdot tr(\Sigma_*)$$
$$m_y = 2a(n - 1) \cdot tr(\Sigma_*)$$

The second new formulae is equation 8

$$1 - \beta = P[F'(h^*,h';\lambda^*) > \frac{fh}{f^*h^*} F^{\alpha}(h,h')] \quad \textbf{(8)}$$

where

$$f^* = \frac{m^* - \sqrt{w^*}}{l^*},$$
$$h^* = \frac{l^{*2}(m^* - 2\sqrt{w^*})}{(m^* - \sqrt{w^*})^2}, \text{ and}$$
$$\lambda^* = \frac{l^{*2}\sqrt{w^*}}{(m^* - \sqrt{w^*})^2} \quad \textbf{(9)}$$

The third new formulae is equation 10.

$$1 - \beta \doteq P\left\{F\left[\frac{(ab\varepsilon + \delta_{\varepsilon})^2}{ab\varepsilon + 2\delta_{\varepsilon}}, b(N - r)\right] > \left\{\left(\frac{ab\varepsilon}{ab\varepsilon + \delta_{\varepsilon}}\right) F^{\alpha}[ab\varepsilon, b(N - \right. \right. \quad \textbf{(10)}$$

Equations 7, 8, and 10 do not require estimates of $E(\hat{\epsilon})$ based on a population variance covariance matrix, but can instead be computed using sample estimates. Results from equations 5, 7, 8 and 10 were compared with the result generated using the following equation:

$$1 - \beta = 1 - FPROB[F_{crit};(ab)\varepsilon, b(N - r)\varepsilon; abF_A\varepsilon] \quad \textbf{(11)}$$

where

$$F_{crit} = Finv[1 - \alpha, abE(\hat{\varepsilon}), b(N - r)E(\hat{\varepsilon})] \quad \textbf{(12)}$$

and

$$F_A = tr(\Delta_*)tr^{-1}(\Sigma_*)[b(N - r) / (ab)] \quad \textbf{(13)}$$

are used in the procedure recommended by Muller and Barton (1989) and amended by Muller and Barton (1991).

All of the above equations were derived using population values. To demonstrate that the new power formulae produce results similar to population estimates, each population estimate was replaced by its sample analog (i.e., Σ will be replaced by $\hat{\Sigma}$) in Monte Carlo simulations.

Procedure

Data Generation

A program was written to replicate the results of Muller and Barton (1989), using SAS Interactive

Matrix Language (SAS/IML) version 6.12 (1997). Random sample variance-covariance matrices were generated using independent normally distributed observation vectors, Z, and spectral decomposition of Σ_*, i.e., $X_{ij} = \mu + T \cdot Z$, where T is a lower triangular matrix satisfying the equality $\Sigma_* = T \cdot T'$, in the method developed by Schuer and Stoller (1966). Additionally, these matrices were contrived to approximate the 'population' matrices (conditions 111, 113, 121, 122, 123, 211, 213, 221, 221 and 223) used by Schuer and Stoller and reported in Table 1. There were two purposes underlying this section, 1) to replicate the work of the above authors as a means of checking the data generation algorithms, and 2) to compare of their results with results of those using the new power formulae.

Table 1

Description of Data from Muller and Barton

		Variance, σ_{ii}				Covariances, σ_{ij}					
Cond	N	1	2	3	4	12	13	14	23	24	34
111	15	1	1	1	1	0.600	0.600	0.600	0.600	0.600	0.600
112	15	1	1	1	1	0.600	0.360	0.216	0.600	0.360	0.600
113	15	1	1	1	1	0.800	0.500	0.200	0.800	0.500	0.800
121	15	1	2	3	4	0.849	1.039	1.200	1.470	1.697	2.078
122	15	1	2	3	4	0.849	0.624	0.432	1.470	1.020	2.080
123	15	1	2	3	4	1.130	1.230	0.400	1.960	1.410	2.770
211	30	1	1	1	1	0.600	0.600	0.600	0.600	0.600	0.600
212	30	1	1	1	1	0.600	0.360	0.216	0.600	0.360	0.600
213	30	1	1	1	1	0.800	0.500	0.200	0.800	0.500	0.800
221	30	1	2	3	4	0.849	1.039	1.200	1.470	1.697	2.078
222	30	1	2	3	4	0.849	0.624	0.432	1.470	1.020	2.080
223	30	1	2	3	4	1.130	1.230	0.400	1.960	1.410	2.770

Note. ϵ = 1.00, .814, .533, .897, .757, and .532 for conditions 111, 112, ..., 123, respectively. Repeated for 211, 212, ..., 223.

Power estimates were calculated and the Mean Absolute Difference (MAD) for both sets of conditions were compared to the results reported by Muller and Barton. The MAD is calculated by subtracting the mean of the simulated values from the predicted value. The predicted value represents the desired level of power and is denoted by PRED in the various tables.

The current study provides a replication of the Monte Carlo simulations conducted by Muller and Barton (1989) and demonstrates the use of Monte Carlo simulations to evaluate the effectiveness of the new power formulae. The first data generation program was used to generate random sample variance-covariance matrices for conditions 111, 112, ..., 223 using data from Muller and Barton (1989). In order to control estimated power, a procedure employed by Muller and Barton was used to generate μ's that result in Geisser-Greenhouse power estimates of .2, .5, and .8 (value for PRED). The structure of μ followed the model $\mu_{jk} = m_{jk} \cdot c = j(k-1)c$, (j = 1, 2, ..., g, k = 1, 2, ..., p) i.e.,

$$\underset{gxp}{\mu} = \begin{bmatrix} 0 & 1 & 2 & \cdots & 1(p-1) \\ 0 & 2 & 4 & \cdots & 2(p-1) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & g & 2g & \cdots & g(p-1) \end{bmatrix} \cdot c. \qquad \textbf{(14)}$$

A value for the constant c can be derived by solving

$$\lambda_* = tr(\Sigma_*) \cdot NCP = z \cdot n \cdot (p-1) \cdot \varepsilon \cdot c^2 \qquad \textbf{(15)}$$

for c, where n is the group size and NCP represents the noncentrality parameter, δ_ϵ. Sums of Squares can be used to determine a value for z, which can also be derived by using the formula $(p^3 - p) / 6$. The value for the constant, c, can then be used to provide the desired power by multiplying m_{jk} by c to obtain $\mu_{jk} = m_{jk} \cdot c$. Power estimates were calculated interactively for every sample matrix generated and the results stored in a separate file using the SAS procedure PROC APPEND. The MAD's using the summary statistics of this output file were used to compare the accuracy of the new power equations compared to those calculated using the procedure recommended by Muller and Barton (1989).

Results and Conclusion

The results for the power formulae, equation 5 (MB1), equation 7 (BT1), equation 8 (BT2), and equation 10 (BT3) when computed in a Monte Carlo simulation (5,000 replications), all provided reliable estimates of power. The MAD for the replication portion of the present study was .023 for the formulae used in the simulations and .014 for MBE Although the overall MAD's were slightly higher for the present study, the overall consistency of the power estimates is comparable to the results reported by Muller and Barton (1989). A closer examination of the MAD's reveals that the variability of the results in this study was greater, but if conditions 111 (power = .5 and .8) and 121 (power = .5 and .8) are omitted, the MAD's

are equivalent for both studies. However, it should be noted that the estimates generated by these two conditions, although more variable, produced reliable results. These values were just slightly greater than those reported by Muller and Barton (1989).

The results from this study demonstrate the viability of the new power formulae, using sample estimators, as procedures to estimate power without having to rely on the technique proposed Muller and Barton (1989). The statistical theory provides the basis for proposing formulae with sample estimators, however, the use of SAS/IML to complete Monte Carlo simulations allows for empirical validation of the accuracy of these new formulae.

References

Muller, K. E., & Barton, C. N. (1989). Approximate power for repeated measures ANOVA lacking sphericity. Journal of the American Statistical Association, 84, 549-555.

SAS Institute Inc. (1990). SAS/IML user's guide (Version 6.06 ed.). Cary, NC: Author.

SAS Institute Inc. (1991). SAS Manual for Linear Models (Version 6.06 ed.). Cary, NC: Author.

Schuer, E. M. & Stoller, D. S. (1966). On the generation of normal random vectors. Technometrics, 4, 279-280.

Thompson, B. L. (1990). A comparison of new power approximations in repeated measures analyses. An unpublished master's thesis, Arizona State University.

Appendix

SAS Monte Carle Simulation Program

```
OPTIONS FULLSTATS NONOTES;
PROC IML;
START MCARLO (NREPS, POPSIGMA, POP_MU, U_I, C_I, ALPHA, E2, POWER);

*********************************************************************
*    ANALYTIC FORMULAE FOR UNIVARIATE POWER ANALYSIS:
*          A MONTE CARLO SIMULATION
*
*    MB1:      THE FIRST MULLER-BARTON EQUATION
*    BT1:      THE FIRST BETZ-THOMPSON APPROXIMATION
*    BT2:      THE SECOND BETZ-THOMPSON APPROXIMATION
*    BT3:      THE THIRD BETZ-THOMPSON APPROXIMATION
*    MBE:      THE MULLER-BARTON PROCEDURE FOR CALCULATING
*              E(EPSILON)
*   SMBE:      MBE CALCULATED FOR EACH SIMULATION
*    U_F:      THE UNCORRECTED F-TEST
*    C_F:      THE CONSERVATIVE F-TEST USING GEISSER-GREENHOUSE
*              LOWER BOUND ESTIMATE FOR EPSILON
*
*    PARAMETERS:      EPSILON  = .814, .533, .757, and .532
*                         P = 4
*                         Total_n  = 15, 30
*
*    GENERATE A SERIES OF PXP COVARIANCE MATRICIES
*    (P= 4) SUCH THAT BOX'S EPSILON IS FIXED AT  ETA= .814, .533, .757, AND
*                                                    .532
*
*    NOTATION:
*              E2          IS A SCALAR REPRESENTING THE PRESET
*                          VALUE FOR EPSILON
*          EVAL            IS THE VECTOR OF EIGENVALUESFOR
*                          SIGMA
*          L1              IS THE VECTOR OF EIGENVALUES FOR S
*          L2              IS THE VECTOR OF EIGENVALUES FOR S2
*          L3              IS THE VECTOR OF THE Q= P-1
*                          EIGENVALUES OF S2
*          L_EPS           IS THE VECTOR OF P EIGENVALUES
*                          COMPUTED SUCH THAT EPSILON= .5087 and .5365
*          LB_EPS          LOWER BOUND ESTIMATE OF EPSILON,
*                          FOLLOWING GEISSER-GREENHOUSE PROCEDURE
*                        1/(P - 1)
*          N               IS THE SAMPLE SIZE
*          P               IS THE DIMENSION OF THE COVARIANCE
*                          MATRIX
*          POPSIGMA     IS THE POPULATION VARIANCE-COVARIANCE
*                          MATRIX
*          S1              IS THE REFERENT VARIANCE-COVARIANCE
*                          MATRIX (EPSILON=1.00)
*          S2              IS A RANDOMLY GENERATED
*                          VARIANCE-COVARIANCE
*                          MATRIX THAT IS USED TO GENERATE S FOR
*                          EPSILON= .814, .533, .757, AND .532
*          T  IS AN ORTHONORMAL MATRIX GENERATED VIA A
*                          GRAM-SCHMIDT ORTHONORMALIZATION OF S2
*          U1              IS THE UNIT VECTOR
*          X1              IS A RANDOMLY GENERATED UNIT NORMAL N X
*                          P DATA
*                          MATRIX USED TO COMPUTE SIGMA
*          POWER           POWER TO BE SET AT .2, .5, AND .8
*
*********************************************************************;

  R= NROW(POP_MU);
  P= NCOL(POP_MU);
  N= 5;
  TOTAL_N= N*R;
  A= NROW(C_I);
  B= P-1;
  TYP_II= 1-POWER;

/*   GIVEN MU AND SIGMA FOR THE COVARIANCE MATRIX WITH
     SPECIFIED EPSILON */

   CALL GSORTH(T1,M,LINDEP,POPSIGMA);
   SIGMA= (T1*DIAG(L_EPS)*T1`);                    /* GENERATE SIGMA_ST*/
   X_POP= I(R)@SHAPE(1,N,1);                       /* ESSENCE MATRIX */
   EVAL1= EIGVAL(SIGMA);
   EPS1= (SUM(EVAL1)*SUM(EVAL1))/((P)*EVAL1`*EVAL1);

*****************************************************************
*  COMPUTE APPROXIMATIONS NEEDED FOR POWER FORMULAE
*****************************************************************;

   PSIGMA= U_I`*POPSIGMA*U_I;
   EVAL= EIGVAL(PSIGMA);
   EPSILON= (SUM(EVAL)*SUM(EVAL))/((B)*EVAL`*EVAL);

*********************************************************************
*  COMPUTE MEAN MATRIX SUCH THAT POWER CAN BE SET TO A
*  SPECIFIED VALUE WHICH IS SELECTED A PRIORI (.2, .5, AND .8)
*********************************************************************;

   DF1= A*B*EPSILON;                        /* NUMERATOR DF */
   DF2= B*(TOTAL_P-A)*EPSILON;                        /* DENOMINATOR DF */
CRIT= FINV(1-ALPHA,DF1,DF2);                  /* CRITICAL VALUE FOR F */     ZNCP=
FNONCT(CRIT,DF1,DF2,TYP_II);      /* NONCENTRAL PARAMETER */     Z= (P**3-P)/6;
                                  /* CONSTANT Z */
  C= SQRT((TRACE(SIGMA)*ZNCP)/ (Z*POP_N*B*EPSILON));
MEAN= C*POP_MU;                             /* CREATE MATRIX OF MEANS */
   POP_MU1= MEAN(|1,|);                     /* MEAN VECTOR FOR GROUP 1 */
   POP_MU2= MEAN(|2,|);                     /* MEAN VECTOR FOR GROUP 2 */
   POP_MU3= MEAN(|3,|);                     /* MEAN VECTOR FOR GROUP 3 */
  PTHETA= C_I*MEAN*U_I;
  EDLTA_ST= PTHETA`*INV(C_I*INV(X_POP`*X_POP)*C_I`)*PTHETA
  ETR_SS= TRACE(PSIGMA);
  ETR_DS= TRACE(EDLTA_ST);

*************************************************************
 *  GENERATE RANDOM VARIANCE COVARIANCE MATRICES
*************************************************************;

DO I= 1 TO NREPS;

  U1= J(N,1,1);
  Z1= J(N,P,0);        */ INITIALIZE THE Z1-MATRIX;
```

```
 Z2= J(N,P,0);          */ INITIALIZE THE Z2-MATRIX;
 Z3= J(N,P,0);          */ INITIALIZE THE Z3-MATRIX;
  X= J(N,P,0);          */ INITIALIZE THE X-MATRIX;
 DO KK= 1 TO N;
  DO LL= 1 TO P;
   Z1(|KK,LL|)= RANNOR(0);
  END;
 END;
 DO MM= 1 TO N;
  DO NN= 1 TO P;
   Z2(|MM,NN|)= RANNOR(0);
  END;
 END;
 DO QQ= 1 TO N;
  DO RR= 1 TO P;
   Z3(|QQ,RR|)= RANNOR(0);
  END;
 END;

*****************************************************************
*    GENERATE RANDOM SAMPLES FROM NORMAL(MU,SIGMA)
*    POPULATION
*****************************************************************;

 MU1= REPEAT(POP_MU1,N,1);
 MU2= REPEAT(POP_MU2,N,1);
 MU3= REPEAT(POP_MU3,N,1);
 T= ROOT(POPSIGMA);                    /* SPECTRAL DECOMPOSITION */
 X1= MU1 + Z1*T;
 X2= MU2 + Z2*T;
 X3= MU3 + Z3*T;
 X_MATRIX= X1//X2//X3;
 X_SAMPLE= I(R)@SHAPE(1,N,1);   /* ESSENCE MATRIX FOR SAMPLE */
 MEAN1= U1`*X1/N;
 MEAN2= U1`*X2/N;
 MEAN3= U1`*X3/N;
 S1= (1/(N - 1))*(X1`*X1 - (U1`*X1)`*(U1`*X1)/N);
 S2= (1/(N - 1))*(X2`*X2 - (U1`*X2)`*(U1`*X2)/N);
 S3= (1/(N - 1))*(X3`*X3 - (U1`*X3)`*(U1`*X3)/N);
 S1_ST= U_I`*S1*U_I;
 S2_ST= U_I`*S2*U_I;
 S3_ST= U_I`*S3*U_I;
 SIGMA_ST= (S1+S2+S3);  */ POOLED VAR-COV MATRIX;
 EVAL_S= EIGVAL(SIGMA_ST);
 EPS_HAT= (SUM(EVAL_S)*SUM(EVAL_S))/(B*(EVAL_S`*EVAL_S));
 S_DF1= A*B*EPS_HAT;
 S_DF2= B*(TOTAL_N-R)*EPS_HAT;
 S_CRIT= FINV(1-ALPHA,S_DF1,S_DF2);
 S_ZNCP= B*EPS_HAT*ETR_DS/ETR_SS;
 Z= (P**3-P)/6;
 S_C= SQRT((TRACE(SIGMA_ST)*S_ZNCP)/(Z*N*B*EPS_HAT));
 MU_HAT= S_C*POP_MU;

*******************************************
*     CALCULATE POWER FORMULAE
*******************************************;

 THETA= C_I*MU_HAT*U_I;
 DELTA_ST=THETA`*INV(C_I*INV(X_SAMPLE`*X_SAMPLE)*C_I`)*THETA;
 TR_SS= TRACE(SIGMA_ST);
 TR_SS2= TRACE(SIGMA_ST**2);
 TR_SS3= TRACE(SIGMA_ST**3);
 TR_DS= TRACE(DELTA_ST);
 TR_SSDS= TRACE(SIGMA_ST*DELTA_ST);
 TR_SS2DS= TRACE(SIGMA_ST**2*DELTA_ST);
 L= A*TR_SS;
 L_ST= L+TR_DS;
 M_ST= (A*TR_SS2) + (2*TR_SSDS);
 N_ST= (A*TR_SS3) + (3*TR_SS2DS);
 W_ST= MAX(0, M_ST**2 - L_ST * N_ST);
 RW_ST= SQRT(W_ST);
 H= L_ST**2 / M_ST;
 HO=A*B*EPS_HAT;
 HP=B*(TOTAL_N-R)*EPS_HAT;
 G_ST=(M_ST - RW_ST) / L_ST;
 H_ST=(L_ST**2*(M_ST-2*RW_ST))/((M_ST-RW_ST)**2);
 FCRIT=FINV(1-ALPHA, HO, HP);
 ADJ_FAC=L / L_ST;
 ADJ_F=FCRIT*ADJ_FAC;
 BT1_PWR=1-PROBF (ADJ_F, H, HP);
 LMBDA_ST=(L_ST**2 * RW_ST) / ((M_ST - RW_ST)**2);
 ADJ_FAC2=L/(G_ST*H_ST);
 ADJ_F2=FCRIT*ADJ_FAC2;
 BT2_PWR=1-PROBF(ADJ_F2, H_ST, HP, LMBDA_ST);
 NCP=B*EPS_HAT*TR_DS/TR_SS;
 MB1_PWR=1-PROBF(FCRIT, HO, HP, NCP);
 DF1A=((HO+NCP)**2)/ (HO + 2*NCP);
 ADJ_FAC3=HO /(HO + NCP);
 ADJ_F3=FCRIT * ADJ_FAC3;
 BT3_PWR=1-PROBF(ADJ_F3, DF1A, HP);

*****************************************************************
*  THIS NEXT PORTION CALCULATES THE UNCORRECTED AND
*  CONSERVATIVE F-TESTS
*****************************************************************;

 LB_EPS= (1/(P-1));
 LB_NDF= A*B*LB_EPS;
 LB_DDF= B*(TOTAL_N-R)*LB_EPS;
 U_NDF= A*B;
 U_DDF= B*(TOTAL_N-R);
 LB_FCRIT= FINV(1-ALPHA, LB_NDF, LB_DDF);
 U_FCRIT= FINV(1-ALPHA, U_NDF, U_DDF);
 LB_NCP= B*LB_EPS*TR_DS/TR_SS;
 U_NCP= B*TR_DS/TR_SS;
 C_F= 1-PROBF(LB_FCRIT, LB_NDF, LB_DDF, LB_NCP);
 U_F= 1-PROBF(U_FCRIT, U_NDF, U_DDF, U_NCP);

*****************************************************************
* THE NEXT PORTION OF THE ALGORITHM CALCULATES THE
* POWER USING THE PROCEDURE RECOMMENDED BY MULLER AND
* BARTON FOR EACH OF THE SAMPLE VARIANCE-COVARIANCE
* MATRICES GENERATED IN THE MONTE CARLO SIMULATION.
*****************************************************************;

 SF1= 2*SUM(EVAL_S)*INV((EVAL_S`*EVAL_S)*B) -
      (2*EVAL_S(|1,1|)*(SUM(EVAL_S)**2) * INV(((EVAL_S`*EVAL_S)**2)*B));
SF11= 2*INV((EVAL_S`*EVAL_S)*B) - 8*EVAL_S(|1,1|)*(SUM(EVAL_S))*
     INV(((EVAL_S`*EVAL_S)**2)*B) +
        8*(EVAL_S(|1,1|)**2)*(SUM(EVAL_S)**2)*
        INV(((EVAL_S`*EVAL_S)**3)*B) -
        2*(SUM(EVAL_S)**2)*INV(((EVAL_S`*EVAL_S)**2)*B);
 SF2= 2*SUM(EVAL_S)*INV((EVAL_S`*EVAL_S)*B) -
      (2*EVAL_S(|2,1|)*(SUM(EVAL_S)**2) * INV(((EVAL_S`*EVAL_S)**2)*B));
SF22= 2*INV((EVAL_S`*EVAL_S)*B) - 8*EVAL_S(|2,1|)*(SUM(EVAL_S))*
     INV(((EVAL_S`*EVAL_S)**2)*B)
        +8*(EVAL_S(|2,1|)**2)*(SUM(EVAL_S)**2) *
        INV(((EVAL_S`*EVAL_S)**3)*B) -
        2*(SUM(EVAL_S)**2)*INV(((EVAL_S`*EVAL_S)**2)*B);
 SF3= 2*SUM(EVAL_S)*INV((EVAL_S`*EVAL_S)*B) -
      (2*EVAL_S(|3,1|)*(SUM(EVAL_S)**2) *INV(((EVAL_S`*EVAL_S)**2)*B));
SF33= 2*INV((EVAL_S`*EVAL_S)*B) - 8*EVAL_S(|3,1|)*(SUM(EVAL_S))*
     INV(((EVAL_S`*EVAL_S)**2)*B) +
        8*(EVAL_S(|3,1|)**2)*(SUM(EVAL_S)**2) *
        INV(((EVAL_S`*EVAL_S)**3)*B) -
        2*(SUM(EVAL_S)**2)*INV(((EVAL_S`*EVAL_S)**2)*B);

SPART1=    SF11*(EVAL_S(|1,1|)**2) + SF22*(EVAL_S(|2,1|)**2)
           SF33*(EVAL_S(|3,1|)**2);
SPART2=    SF1*(EVAL_S(|1,1|)*EVAL_S(|2,1|))  / (EVAL_S(|1,1|)-EVAL_S(|2,1|))     +
           SF1*(EVAL_S(|1,1|)*EVAL_S(|3,1|))  /
           (EVAL_S(|1,1|)-EVAL_S(|3,1|)) +
           SF2*(EVAL_S(|2,1|)*EVAL_S(|3,1|))  / (EVAL_S(|2,1|)-EVAL_S(|3,1|))     +
           SF2*(EVAL_S(|2,1|)*EVAL_S(|1,1|))  /
           (EVAL_S(|2,1|)-EVAL_S(|1,1|)) +
           SF3*(EVAL_S(|3,1|)*EVAL_S(|1,1|))  / (EVAL_S(|3,1|)-EVAL_S(|1,1|))     +
           SF3*(EVAL_S(|3,1|)*EVAL_S(|2,1|))  / (EVAL_S(|3,1|)-EVAL_S(|2,1|));

IF EVAL_S(|1,1|)>EVAL_S(|2,1|)>EVAL_S(|3,1|) THEN
 SG1= SPART1 + SPART2;
 ELSE SG1= SPART1;

 SEXP_EPS= EPS_HAT + (SG1/(TOTAL_N-R));
 SEHO= A*B*SEXP_EPS;
 SHO= A*B*EPS_HAT;
 SEHP= B*(TOTAL_N-A)*SEXP_EPS;
 SHP= B*(TOTAL_N-A)*EPS_HAT;
 SEFCRIT= FINV(1-ALPHA, SEHO, SEHP);
 SENCP= B*EPS_HAT*(TR_DS/TR_SS);
 SMBE_PWR= 1-PROBF(SEFCRIT, SHO, SHP, SENCP);

*****************************************************************
*   THE NEXT PORTION OF THE ALGORITHM CALCULATES THE
*   POWER USING THE PROCEDURE RECOMMENDED BY MULLER AND
*   BARTON FOR THE VARIANCE-COVARIANCE WHICH IS BEING
*   TREATED AS A 'POPULATION MATRIX' IN THE MONTE CARLO
*   SIMULATION.
*****************************************************************:

  F1= 2*SUM(EVAL)*INV((EVAL`*EVAL)*B) -(2*EVAL(|1,1|)*(SUM(EVAL)**2)*
      INV(((EVAL`*EVAL)**2)*B));
 F11= 2*INV((EVAL`*EVAL)*B) - 8*EVAL(|1,1|)*(SUM(EVAL))*
      INV(((EVAL`*EVAL)**2)*B) + 8*(EVAL(|1,1|)**2)*(SUM(EVAL)**2)*
      INV(((EVAL`*EVAL)**3)*B) -
      2*(SUM(EVAL)**2)*INV(((EVAL`*EVAL)**2)*B);
  F2= 2*SUM(EVAL)*INV((EVAL`*EVAL)*B) -(2*EVAL(|2,1|)*(SUM(EVAL)**2)*
      INV(((EVAL`*EVAL)**2)*B));
 F22= 2*INV((EVAL`*EVAL)*B) - 8*EVAL(|2,1|)*(SUM(EVAL))*
      INV(((EVAL`*EVAL)**2)*B) + 8*(EVAL(|2,1|)**2)*(SUM(EVAL)**2)*
      INV(((EVAL`*EVAL)**3)*B) -
      2*(SUM(EVAL)**2)*INV(((EVAL`*EVAL)**2)*B);
```

```
   F3= 2*SUM(EVAL)*INV((EVAL`*EVAL)*B) -(2*EVAL(|3,1|)*(SUM(EVAL)**2)*
       INV(((EVAL`*EVAL)**2)*B));
  F33= 2*INV((EVAL`*EVAL)*B) - 8*EVAL(|3,1|)*(SUM(EVAL))*
       INV(((EVAL`*EVAL)**2)*B) + 8*(EVAL(|3,1|)**2)*(SUM(EVAL)**2)*
       INV(((EVAL`*EVAL)**3)*B) -
       2*(SUM(EVAL)**2)*INV(((EVAL`*EVAL)**2)*B);

 PART1=    F11*(EVAL(|1,1|)**2) + F22*(EVAL(|2,1|)**2) +
                F33*(EVAL(|3,1|)**2);
 PART2=    F1*(EVAL(|1,1|)*EVAL(|2,1|)) / (EVAL(|1,1|)-EVAL(|2,1|)) +
   F1*(EVAL(|1,1|)*EVAL(|3,1|)) / (EVAL(|1,1|)-EVAL(|3,1|)) +
       F2*(EVAL(|2,1|)*EVAL(|3,1|)) / (EVAL(|2,1|)-EVAL(|3,1|)) +
       F2*(EVAL(|2,1|)*EVAL(|1,1|)) / (EVAL(|2,1|)-EVAL(|1,1|)) +
       F3*(EVAL(|3,1|)*EVAL(|1,1|)) / (EVAL(|3,1|)-EVAL(|1,1|)) +
       F3*(EVAL(|3,1|)*EVAL(|2,1|)) / (EVAL(|3,1|)-EVAL(|2,1|));

 IF EVAL(|1,1|)>EVAL(|2,1|)>EVAL(|3,1|) THEN
  G1= PART1 + PART2;
  ELSE G1= PART1;

  EXP_EPS= EPSILON + (G1/(TOTAL_N-R));
  EPHO= A*B*EXP_EPS;
  PHO= A*B*EPSILON;
  EPHP= B*(TOTAL_N-A)*EXP_EPS;
  PHP= B*(TOTAL_N-A)*EPSILON;
  EFCRIT= FINV(1-ALPHA, EPHO, EPHP);
  ENCP= B*EPSILON*(ETR_DS/ETR_SS);
  MBE_PWR= 1-PROBF(EFCRIT, PHO, PHP, ENCP);

************************************************************
*  CREATE DATA VECTOR WHICH I CAN APPEND RESULTS
*  FROM EACH SIMULATION TO GENERATE A N x P NUMBER
*  OF OUTPUT VALUES FOR THE MONTE CARLO SIMULATION
************************************************************;


    DATA_ONE=   TOTAL_N||N||P||EPS_HAT||EXP_EPS||U_F||
                C_F||MB1_PWR||BT1_PWR ||BT2_PWR||
                BT3_PWR||SMBE_PWR||MBE_PWR||EPSILON||EPS1;
    APPEND FROM DATA_ONE;
END;
FINISH;

************************************************************
*    PROCESS MULLER AND BARTON DATA
************************************************************;

 /*  VARIANCE-COVARIANCE MATRICES  */

COND112= {1.000   .600  .360  .216,
          .600  1.000  .600  .360,
          .360   .600 1.000  .600,
          .216   .360  .600 1.000};

COND113= {1.000  .800  .500  .200,
          .800 1.000  .800  .500,
          .500  .800 1.000  .800,
          .200  .500  .800 1.000};

COND122= {1.000  .600  .360  .216,
          .600 2.000  .600  .360,
          .360  .600 3.000  .600,
          .216  .360  .600 4.000};

COND123= {1.000  .800  .500  .200,
          .800 2.000  .800  .500,
          .500  .800 3.000  .800,
          .200  .500  .800 4.000};

* GENERATE HELMERT MATRIX AND NORMALIZE IT;

U_I1={-3  0  0, 1 -2  0, 1  1 -1, 1  1  1};
U_I1=U_I1/SHAPE(SQRT(U_I1[##,]),NROW(U_I1),NCOL(U_I1));
C_RM={ 1  1  1};   /* REPEATED MEASURES */
C_I1={ -2  1  1, 0 1 -1};    /* INTERACTION */

*  GENERATE MEAN MATRIX;

MU_POP= { 0 1 2, 0 2 4, 0 3 6};

*********************************************************************
*  CREATE OUTPUT MATRIX TO BE USED IN SUMMARY ANALYSIS
*********************************************************************;

  CREATE DATA_TWO VAR {TOTAL_N N P EPS_HAT EXP_EPS U_F C_F
                       MB1_PWR BT1_PWR BT2_PWR BT3_PWR
                       SMBE_PWR MBE_PWR EPSILON EPS1};

RUN MCARLO(5000, COND112, MU_POP, U_I1, C_I1, 5, .814, .05, .80);

QUIT;

PROC PRINT DATA=DATA_TWO;

**************************************
*        SUMMARY ANALYSIS
**************************************;

PROC MEANS MEAN STD VAR STDERR;
 CLASS N;
 VAR EPS_HAT EXP_EPS U_F C_F MB1_PWR BT1_PWR BT2_PWR BT3_PWR
      SMBE_PWR MBE_PWR EPSILON EPS1;
RUN;
```


Sean W. Mulvenon, Ph.D.
241 Graduate Education Building
University of Arkansas
Fayetteville, AR 72701
E-mail: seanm@comp.uark.edu