# PrecMod: An Automated Precision SAS® Macro for Random Effects Models

Jesse A. Canchola, Roche Molecular Diagnostics, Pleasanton, California, USA
Pari Hemyari, Roche Molecular Diagnostics, Pleasanton, California, USA

## ABSTRACT

Typical random effects linear model estimation involves fitting a model with main effects that may or may not include nesting with other study factors. Part of the challenge comes when having to calculate the confidence intervals for the variance components using the correct effective degrees of freedom (Satterthwaite, 1946) and then iterating the macro over different grouping levels (if they exist). The current precision macro, PrecMod, surmounts these challenges and provides a clear and concise path towards efficient and timely calculations ready for reporting.

## INTRODUCTION

Random effects models are used in a typical randomized controlled trial for reproducibility analysis to estimate the variance component effects and total variability. For example, a typical registrational study in the medical/molecular diagnostics industry evaluating precision will measure the effect of site, reagent lot, instrument, operator and the totality of these effects on precision. The Clinical Laboratory Standards Institute's (CLSI) guidance, EP05-A3, defines the terms reproducibility and repeatability as they relate to the term precision. Specifically, precision or "total precision" refers to either within-site, within-laboratory or within-device precision and is generally termed "repeatability". On the other hand, the term "reproducibility" seeks to measure precision across sites or laboratories across time (usually at least 5 days) and includes the within-site or within-laboratory precision.

The PrecMod SAS macro for random effects models presented here provides a framework from which to estimate variation for multiple sources as required by regulatory agencies or as standard research requires.

## THEORY

The standard theory for variance components models is essentially an analysis of variance (ANOVA) using a random effects model as follows, as an example, for two-factor studies (Neter et al., 1996).

$$Y_{sr} = \mu + S_s + R_r + \varepsilon_{sre} \quad (1)$$

where

$$S_s \overset{\text{iid}}{\sim} N(0,\sigma_S^2);\ R_r \overset{\text{iid}}{\sim} N(0,\sigma_R^2);\ \varepsilon_{sre} \overset{\text{iid}}{\sim} N(0,\sigma_\varepsilon^2)$$

and where "$\overset{\text{iid}}{\sim}$" means "is independently and identically distributed as" and $N(0,\sigma_X^2)$ is the Normal or Gaussian probability distribution written generally as $N(\mu,\sigma_X^2)$, with mean $\mu = 0$ and variance $\sigma_X^2$ where X=S, R or e, representing *Site*, *Run* and *error*, respectively.

The total variance of Y, or $\sigma_Y^2$, is the sum of the individual variance components estimated from model (1). Specifically:

$$\sigma_Y^2 = \sigma_S^2 + \sigma_R^2 + \sigma_\varepsilon^2 \quad (2)$$

Different optimization methods exist for finding the variance component estimates with the two most common being maximum likelihood (ML) and restricted maximum likelihood (REML). The current macro allows for ML and REML with the default to REML if no estimation method is specified. An additional method using minimum variance quadratic unbiased estimation (MIVQUE) is also an option.

A typical random effects model required by a regulatory agency might include random effects due to lot, site, operator, day, and run or batch, so that model (1) is extended as follows.

$$Y_{lsodre} = \mu + L_l + S_s + O_o + D_d + R_r + \varepsilon_{lsodre} \quad (3)$$

where

$$L_l \overset{\text{iid}}{\sim} N(0,\sigma_L^2)\,;\; S_s \overset{\text{iid}}{\sim} N(0,\sigma_S^2)\,;\; O_o \overset{\text{iid}}{\sim} N(0,\sigma_O^2)\,;\; D_d \overset{\text{iid}}{\sim} N(0,\sigma_D^2)\,;\; R_r \overset{\text{iid}}{\sim} N(0,\sigma_R^2)\,;\; \varepsilon_{lsodre} \overset{\text{iid}}{\sim} N(0,\sigma_\varepsilon^2)\,.$$

So that the total variance of Y, or $\sigma_Y^2$, is the sum of the individual variance components estimated from model (3). Specifically:

$$\sigma_Y^2 = \sigma_L^2 + \sigma_S^2 + \sigma_O^2 + \sigma_D^2 + \sigma_R^2 + \sigma_\varepsilon^2 \qquad (4)$$

so that it follows that an estimate of the total standard deviation will be

$$SD = \sqrt{\hat{\sigma}_Y^2} \qquad (5)$$

where $\hat{\sigma}_Y^2$ is an estimate of the total variance from the sample.

## GENERAL EXAMPLE

The above ANOVA model will be performed on the logarithm base 10 (i.e., $\log_{10}$) transformed HIV RNA concentration results for positive panel members (i.e., panel members or assay levels for which a result is greater than zero). A typical reproducibility model in the medical device (diagnostics) industry includes lot, site/instrument, operator, day, run, and within-run (random error) as random effects. The assumption that the different effects are random allows their contribution to the percentage of variance to be estimated. Note that effects can be nested in other effects by design. Here we assume that nesting is negligible for simplification of exposition.

The random effects model to be used is described as follows.

$$y_{lsodre} = \mu + L_l + S_s + O_o + D_d + R_r + \varepsilon_{lsodre} \qquad (3)$$

where

$$L_l \overset{\text{iid}}{\sim} N(0,\sigma_L^2)\,;\; S_s \overset{\text{iid}}{\sim} N(0,\sigma_S^2)\,;\; O_o \overset{\text{iid}}{\sim} N(0,\sigma_O^2)\,;\; D_d \overset{\text{iid}}{\sim} N(0,\sigma_D^2)\,;\; R_r \overset{\text{iid}}{\sim} N(0,\sigma_R^2)\,;\; \varepsilon_{lsodre} \overset{\text{iid}}{\sim} N(0,\sigma_\varepsilon^2)\,.$$

The linear mixed effects model uses the SAS MIXED procedure and will provide the restricted maximum likelihood estimates (REML) of the variance components for the included factors: lot (L), site (S); operator (O); day (D); run (R); and sample within-run ($\varepsilon$). The dependent variable for the model is the $\log_{10}$ HIV RNA quantitation. HIV is the Human Immunodeficiency Virus and RNA (viz., ribonucleic acid) is the nucleic acid present within the virus that is amplified (i.e., quantitated) by the polymerase chain reaction (PCR) exponential (geometric) replication process. Because this quantitated outcome is exponential, it is linear in the logarithmic scale within the dynamic range of the assay [i.e., from the Lower Limit of Quantitation (LLoQ) to the Upper Limit of Quantitation (ULoQ)]. Therefore, the outcome is analyzed in $\log_{10}$ units. The original units will be different for different "targets" such as HIV in copies per milliliter (cp/mL), Hepatitis B or C viruses (HBV or HCV, respectively) in international units per milliliter (IU/mL), and so on.

The PrecMod macro includes the following SAS MIXED procedure specification.

```
proc mixed data = DataIn method = &method ;
 class &ClassVars ;
 model &DepVar  = / solution cl ddfm=satterth ;
 random &Random;
 where LevelVar = &index ;
run ;
```

with specifications in the macro call:

```
%PrecMod(InData    = ,      ❶
         ByVars    = ,      ❷
         ClassVars = ,      ❸
         DepVar    = ,      ❹
         IndepVars = ,      ❺
         Random    = ,      ❻
         Method    = REML , ❼
```

```
OutData   = ,        ❽
FlagVar   = ,        ❾
FlagValue = ,        ❿
EquTest   = N    ,   ⓫
Alpha     = 0.05 ,   ⓬
InsType   = ,        ⓭
Logged    = "Yes");  ⓮
```

and macro parameters defined in Table 1 below.

| Parameter # | Macro Parameter | General Macro Parameter Description | Specifications / Defaults | Example |
|---|---|---|---|---|
| ❶ | InData | SAS input data set (permanent or temporary) | < previous data set > | anadata.valid |
| ❷ | ByVars | "By" processing variable | < empty > | instrument cap_avg cpavglog |
| ❸ | ClassVars | Categorical or Class Variables | [Required] | site day run |
| ❹ | DepVar | Dependent Variable | [Required] | Logconc |
| ❺ | IndepVars | Independent Fixed Variables | < typically empty > | |
| ❻ | Random | Independent Random Variables | [Required] | site day run |
| ❼ | Method | Optimization method | REML, MLE, MIVQUE() / REML | REML |
| ❽ | OutData | Output SAS data set (permanent or temporary) | [Required] | anadata.mixed |
| ❾ | FlagVar | Any data flag or indicator variable | < empty > | dvlflag |
| ❿ | FlagValue | Value for FlagVar for subsetting | < empty > | DVL |
| ⓫ | EquTest | Test for Equivalency for Two Systems/Items Indicated | N | Y |
| ⓬ | Alpha | Statistical Significance Level | 0 < α < 1.0 / 0.05 | 0.05 |
| ⓭ | InsType | Declare a variable for instrument type, variables should match input variables. Required for ⓫ EquTest. | < empty > | Instrument |
| ⓮ | Logged | If "Yes", macro will use correct percent coefficient of variation (%CV) formula | "No" | "Yes" |

**Table 1. PrecMod SAS Macro Parameter Specifications and defaults**

## PRACTICAL EXAMPLE

We present an analysis of 1855 sample reproducibility data for a HIV-1 PCR assay test evaluated at six titer/concentration levels across the following factors:

- **Lot:** 3 manufactured reagent lots
- **Site/Instrument:** 3 test sites; 1 instrument per site

- **Operator:** 2 operators performing testing at each site
- **Day/Run:** 5 days per lot for each operator; 1 run per day
- **Within-Day/run:** 3 replicates for each HIV-1 RNA concentration

Two different operators are at each of 3 test sites each performed 5 days of testing with each of 3 lots of reagents.

For this example, the specification for the PrecMod SAS macro is as follows:

```
%PrecMod(InData    = DataSetIn ,                ①
         ByVars    = titer_n titer_c logexp  ,  ②
         ClassVars = dslot site operator day ,  ③
         DepVar    = logconc ,                  ④
         IndepVars = ,                          ⑤
         Random    = dslot site operator day ,  ⑥
         Method    = REML ,                     ⑦
         OutData   = ResultsDat ,               ⑧
         FlagVar   = dvlflag ,                  ⑨
         FlagValue = DVL ,                      ⑩
         EquTest   = N    ,                     ⑪
         Alpha     = 0.05 ,                     ⑫
         InsType   = ,                          ⑬
         Logged    = "Yes") ;                   ⑭
```

| Parameter # | Macro Parameter | General Macro Parameter Description | Example | Description |
|---|---|---|---|---|
| ① | InData | SAS input data set | DataSetIn | Temporary SAS Input Data Set |
| ② | ByVars | "By" processing variable | titer_n<br>titer_c<br>logexp | Numeric titer<br>Character titer<br>$Log_{10}$(expected conc) |
| ③ | ClassVars | Categorical or Class Variables | dslot<br>site<br>operator<br>day | Lot factor<br>Site factor<br>Operator factor<br>Day factor |
| ④ | DepVar | Dependent Variable | Logconc | Logarithm base 10 of Observed Concentration |
| ⑤ | IndepVars | Independent Fixed Variables | < empty > | No fixed variables |
| ⑥ | Random | Independent Random Variables | dslot, site, operator, day | Same variables as in ③ ClassVars |
| ⑦ | Method | Optimization method | REML | Restricted Maximum Likelihood |
| ⑧ | OutData | Output SAS data set | ResultsDat | Temporary SAS data set |
| ⑨ | FlagVar | Any data flag or indicator variable | dvlflag | Flag |
| ⑩ | FlagValue | Value for FlagVar for subsetting | DVL | Flag value for subset |
| ⑪ | EquTest | Test for Equivalency for Two | N | No equivalency testing |

| | | Systems Indicated | | requested |
|---|---|---|---|---|
| ⑫ | Alpha | Statistical Significance Level | 0.05 | Standard two-sided significance |
| ⑬ | InsType | Declare a variable for instrument type, variables should match input variables | < empty > | No "instrument" variable declared since equivalency testing is not requested (⑪ EquTest=N) |
| ⑭ | Logged | If "Yes", macro will use correct percent coefficient of variation (%CV) formula | "Yes" | Since ④ DepVar is logged in original data, choose "Yes" here. |

**Table 2. PrecMod SAS macro parameter inputs for example of HIV-1 PCR assay test evaluated at six titer/concentration levels across lot, site, operator, day.**

Additional coding can be added to format the results in a table using the SAS REPORT procedure (APPENDIX B).

## RESULTS

Table 3 shows the results as formatted using the DATA step and PROC REPORT code in APPENDIX B. Additional tables can be produce similarly for the percent coefficient of variation (%CV), standard deviation (SD) and variance.

| HIV-1 RNA Concentration ($log_{10}$ cp/mL) | | | Random Effects Model Components Contribution to Total Variance (%) | | | | | Total Precision |
|---|---|---|---|---|---|---|---|---|
| Expected | Observed (average) | No. of Tests[a] | Lot | Site | Operator | Day | Within-Day | Standard Deviation (Lognormal %CV)[b] |
| 1.699 | 1.624 | 263 | 5.3% | 0.7% | 0.0% | 2.5% | 91.4% | 0.26 (66.1%) |
| 1.699** | 1.646 | 263 | 4.4% | 2.7% | 0.0% | 0.9% | 92.0% | 0.25 (63.1%) |
| 2.301 | 2.260 | 267 | 1.1% | 1.1% | 1.1% | 0.0% | 96.7% | 0.13 (32.3%) |
| 2.602 | 2.538 | 267 | 2.3% | 1.3% | 11.4% | 0.5% | 84.5% | 0.10 (23.3%) |
| 3.000 | 2.983 | 266 | 0.5% | 0.0% | 4.6% | 0.0% | 94.9% | 0.12 (28.6%) |
| 5.000 | 4.961 | 265 | 0.9% | 17.2% | 2.7% | 0.4% | 78.7% | 0.09 (21.3%) |
| 6.699 | 6.673 | 264 | 1.7% | 71.5% | 2.8% | 0.3% | 23.7% | 0.16 (39.7%) |

Note: Results with detectable viral load are included in this table.

[a] Number of tests with detectable viral load.
[b] Lognormal %CV = 100% • square root of {10^[variance • ln(10)]-1}, where ln() is the natural logarithm
** Results < 3.40E+1 cp/mL were calculated based on the validated in-house software from ΔCt values for test results that were below LLoQ.

Data Source: Appendix X, Table Y.1, Table Y.2.

**Table 3. Attributable percentage of total variance, total precision standard deviation and lognormal %CV of HIV-1 RNA concentration (log10 cp/mL) from tests with detectable viral load.**

## CONCLUSION

We presented the random effects linear model that can be fit to the data for the estimation of variance components for each factor in the model. The variance estimates are then used to derive the standard deviation, percent coefficient of variation and attributable percentage of total variance. Using the MIXED procedure and additional coding allowed us to calculate confidence intervals for the variance components using the correct effective degrees of freedom (Satterthwaite, 1946). Since there may be more than one level in a study for the calculations, we coded the macro to iterate over any number of grouping levels. Finally, we provided an example that showed conclusive

evidence that the PrecMod SAS macro is easy to use and provides a clear and concise path towards efficient and timely calculations ready for reporting.

## REFERENCES

Agresti A, Coull BA. Approximate is better than “exact” for interval estimation of binomial proportions. The Americal Statistician. 1988; 52(2): 119-126.

Clinical Laboratory Standards Institute (CLSI) document EP05-A2, Evaluation of Precision Performance of Quantitative Measurement Methods; Approved Guideline-Second Edition. 2004. Wayne, PA.

Clincial Labroratory Standards Institute (CLSI) document EP12-A2. “User Protocol for Evaluation of Qualitative Test Performance; Approved Guideline - Second Edition”. January; 2008.

Clopper CJ, Pearson ES. The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika 1934; 26: 404–13.

Guidance for Industry and FDA: “Statistical Guidance on Reporting Results from Studies Evaluating Diagnostic Tests” March 13, 2007.

Nelson W. Applied Life Data Analysis. In. New York, NY: John Wiley and Sons; 1982.

Neter J, Kutner MH, Nachtsheim CJ, Wasserman W (1996). Applied Linear Statistical Models. pp. 976-977. Boston, MA: IRWIN.

SAS Enterprise Guide v5.1. Copyright © 2012 by SAS Institute Inc., Cary, NC, USA. All rights reserved. product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA.

Wilson EB. Probable inference, the law of succession and statistical inference, JASA. 1927; 22: 209-212.

Satterthwaite, F. E. (1946), "An Approximate Distribution of Estimates of Variance Components.", *Biometrics Bulletin* **2**: 110–114,doi:10.2307/3002019

SAS Institute Inc. 2011. SAS/STAT® 9.3 User’s Guide. Cary, NC: SAS Institute Inc.

## ACKNOWLEDGMENTS

The authors thank Enrique Marino and Merlin Njoya for their valuable comments as well as for testing and validating the macro.

## CONTACT INFORMATION

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

Jesse A. Canchola
Roche Molecular Systems, Inc
4300 Hacienda Drive
Pleasanton, California, 94588
eMail: Jesse.Canchola@Roche.Com
Web: http://usdiagnostics.roche.com/en/index.html

**APPENDIX A. PrecMod SAS Macro Code**

```
********************************************************************************************** ;
* Macro:  PrecMod.sas
*
* Objective:  Precision estimates from random effects / mixed model
*
* PLEASE SPECIFY CORRECT ITEMS TO THE RIGHT OF THE "=" SIGN ;
*  InData      =  * Input data set name ;
*  ByVars      =  * Declare "by" variables to stratify the mixed model, variables should match
*                   input vars ;
*  ClassVars   =  * Declare "class" variables in random effects model, variables should match
*                  input vars ;
*  DepVar      =  * Dependent variable, should match input var ;
*  IndepVar    =  * Independent variables, should match input var ;
*  Random      =  * Random Effects variables including nesting effects, should match input var ;
*  Method      =  * Estimation method for Proc Mixed: ML, MIVQUE(), or REML <default if nothing
*                   specified> ;
*  OutData     =  * Output data set name;
*  FlagVar     =  * Include only vaild observations ;
*  FlagValue   =  * Put an actual/right value to recognize  ;
*  EquTest     =  * For equivalency test of total variance between instruments;
*  Alpha       =  * Significant level;
*  InsType     =  * Declare a variable for instrument type, variables should match input vars;
*  Logged      =  * Switch for correct presentation of percent CVs: make sure to have quotes, if
*                   the DepVar  is in log10, then this is "Yes" ;
*  Example:

        %PrecMod(InData=anadata.valid,
                ByVars=instypn cap_avg cpavglog,
               ClassVars=rsinitn day run,
               DepVar=logconc,
               IndepVars=,
               Random=rsinitn day(rsinitn) run(rsinitn day),
                Method=REML,
               OutData=anadata.mixed,
                FlagVar=dvlflag,
                FlagValue=DVL,
                EquTest=Y,
                Alpha=0.05,
                InsType=Instypn,
               Logged="Yes");

* Created by:        Jesse A. Canchola (JAC)
* Creation date:     30-Sep-2015
********************************************************************************************** ;
%macro PrecMod(InData=,ByVars=,ClassVars=,DepVar=,IndepVars=,Random=,Method=,OutData=,FlagVar=,
              FlagValue=,EquTest=,InsType=,Alpha=,Logged=);

* For "BY" processing, we use a wrapper around the current code and spit out by level.  ;

****** Chose only valid tests/observations, if necessary.  ******;
%if &flagvar ^= %then %do;
    proc sort data=&InData out=DataIn;
        by &ByVars;
        where upcase(&flagvar)="&flagvalue";
    run;
```

```
%end;
%else %if &flagvar = %then %do;
    proc sort data=&InData out=DataIn;
        by &ByVars;
    run;
%end;

****** Read "by" statement variables and prepared for automatic iteration.  ******;
data _null_;
    byvar = tranwrd(trim("&ByVars"), " ", "*");
    call symputx('bivar',byvar);
run;

proc freq data=DataIn;table &bivar/list out=frequency(drop=count percent); run;

data vars;
    set frequency end=eof;
    LevelVar = _N_;
    if eof = 1 then call symput('ByVarLevels',_N_);
run;

proc sort data=vars;
    by &ByVars;
run;

****** Merged into the main input dataset.  ******;
data DataIn;
    merge DataIn vars;
    by &ByVars;
run;

%let Model = &Random ; * indicate random effects model nesting ;

%do index = 1 %to &ByVarLevels ; * iterate over any by variables ;
 ods output covparms  = cov
            dimensions= totobsv(where=(descr='Max Obs Per Subject') rename=(value=totobs))
            solutionf = solutionf(keep=estimate df lower upper) ;
 proc mixed data = DataIn method = &method ;
  class &ClassVars ;
  model &DepVar  = / solution cl ddfm=satterth ;
  random &Random;
  where LevelVar = &index ;
 run ;
 * grabbing the variance-covariance matrix ;
 ods output AsyCov=AsyCovA ;
 proc varcomp data = DataIn method = &method ;
  class &ClassVars ;
  model &DepVar  = &Model ;
  where LevelVar = &index ;
 run ;
 * grabbing naive mean ;
 proc means data = DataIn ;
  var &DepVar  ;
  output out=Mean_Naive Mean=Mean_Naive ;
  where LevelVar = &index ;
 run ;
```

```
 proc iml ;
  ** process variance-covariance matrix ** ;
  use AsyCovA ;
  read all var _ALL_ into X[colname=varNames] ; * "into" will read only the numeric variables in
AsyCovA into an X matrix ;
  close AsyCovA ;
  TotVarCov = sum( X ) ; * sum all elements of the variance-covariance matrix ;

  ** process variance estimates ** ;
  use cov ;
  read all var _ALL_ into Y[colname=varNames] into Y ;
  close cov ;
  TotVariance = sum( Y ) ; * sum the "Estimate" variable ;
  TotSD = sqrt( TotVariance ) ;

  ** process total sample size variable ** ;
  use TotObsv ;
  read all var _ALL_ into Z[colname=varNames] ;
  close TotObsv ;
  N = Z ;

  ** process model-based mean from proc mixed ** ;
  use SolutionF ;
  read all var _ALL_ into K[colname=VarNames] ;
  close SolutionF ;
  Mean_MB = K[,1] ; * Model-based mean ;

  ** process naive mean from proc means ** ;
  use Mean_Naive ;
  read all var _ALL_ into L[colname=VarNames] ;
  close Mean_Naive ;
  Mean_Naive = K[,1] ; * Model-based mean ;

  * Calculate Satterthwaite Degrees of Freedom ;
  ****** Removed a round function (i.e. round(xxx,1)) from the Satterthwaite degree freedom for
accuracy.  ******;
  DF = (2 * TotVariance**2) / TotVarCov; * Degrees of freedom using Satterthwaite approximation ;
  Chi1 = cinv(0.025,DF) ;
  Chi2 = cinv(0.975,DF) ;

  * Calculate the 95% CI for the sum of the variances ;
  Lower = ( DF * TotVariance ) / Chi2 ;
  Upper = ( DF * TotVariance ) / Chi1 ;

  * Calculate the Standard Error of the Mean ;
  SE = TotSD / sqrt( N ) ; * Define SE of mean ;

  create combineDat var {Mean_MB Mean_Naive TotVarCov TotVariance Lower Upper TotSD N DF SE} ;
  append ;
  close combineDat ;
 run ;

 ****************************************************** ;
 ****************************************************** ;
 * Using SQL, can use cov data rather than outvar to get variables dynamically ;
 * then can transpose that or grab the names using call symput ;
 proc sql noprint noerrorstop ;
```

```
   create table work.var as
   select compress("VAR_" || CovParm ), Estimate from cov ;
 quit ;
 proc sql noprint noerrorstop ;
   create table work.pc as
   select compress("PC_" || CovParm ), Estimate from cov ;
 quit ;
 proc sql noprint noerrorstop ;
   create table work.sd as
   select compress("SD_" || CovParm ), Estimate from cov ;
 quit ;
 proc sql noprint noerrorstop ;
   create table work.cv as
   select compress("CV_" || CovParm ), Estimate from cov ;
 quit ;

 * temp variance components to wide format- note these all have the same values but these ;
 * are temp to be calculated correctly in proc iml ;
 proc transpose data=var out=outvar_var (drop=_NAME_) ; id _TEMA001 ; var Estimate ; run ;
 proc transpose data=pc  out=outvar_pc  (drop=_NAME_) ; id _TEMA001 ; var Estimate ; run ;
 proc transpose data=sd  out=outvar_sd  (drop=_NAME_) ; id _TEMA001 ; var Estimate ; run ;
 proc transpose data=cv  out=outvar_cv  (drop=_NAME_) ; id _TEMA001 ; var Estimate ; run ;

 * merge combineDat with each outvar for PC, SD and CV ;
 data temp_Var ;  merge combineDat outvar_Var ; run ;
 data temp_PC  ;  merge combineDat outvar_PC  ; run ;
 data temp_SD  ;  merge combineDat outvar_SD  ; run ;
 data temp_CV  ;  merge combineDat outvar_CV  ; run ;

 proc iml ;
  ** process Variance ** ;
  use temp_Var ;
  read all var _ALL_ into A[colname=varNames] ; * "into" will read only the numeric variables in
temp_Var into an A matrix ;
  close temp_Var ;

  * save the bits and pieces into a data set ;
  create Var_Dat from A[colname=varNames]   ; * try to select only the last components? ;
  append from A ;
  close Var_Dat ;

  ** process PC ** ;
  use temp_PC ;
  read all var _ALL_ into X[colname=varNames] ; * "into" will read only the numeric variables in
temp_PC into an X matrix ;
  close temp_PC ;

  TotVariance = X[,4] ;
  do i = 11 to ncol(X) ; * take only the relevant dynamic columns from 11 to last ;
     X[,i] = 100 * ( X[,i] ) / TotVariance ;
  end ;

  * save the bits and pieces into a data set ;
  create PC_Dat from X[colname=varNames]   ; * try to select only the last components? ;
  append from X ;
  close PC_Dat ;
```

```
 ** process SD ** ;
 use temp_SD ;
 read all var _ALL_ into Y[colname=varNames] ;
 close temp_SD ;
 TotSD = Y[,7] ;
 do i = 11 to ncol(Y) ; * take only the relevant dynamic columns from 11 to last ;
    Y[,i] = sqrt( Y[,i] ) ; * TotSD can also be grabbed from the previous Proc iml above ;
 end ;

 * save the bits and pieces into a data set ;
 create SD_Dat from Y[colname=varNames]   ; * try to select only the last components? ;
 append from Y ;
 close SD_Dat ;

 ** process CV ** ;
 use temp_CV ;
 read all var _ALL_ into Z[colname=varNames] ;
 close temp_CV ;
 Mean_MB    = Z[,1] ;
 if upcase(&logged) = "NO" then do ;
   do i = 11 to ncol(Z) ; * take only the relevant dynamic columns from 11 to last ;
      Z[,i] = 100 * ( sqrt( Z[,i] ) / Mean_MB ) ;
   end ;
 end ;
 if upcase(&logged) = "YES" then do ;
   do i = 11 to ncol(Z) ; * take only the relevant dynamic columns from 11 to last ;
      Z[,i] = 100 * ( sqrt( 10**( Z[,i] * log(10)) - 1 ) ) ;
   end ;
 end ;

 * save the bits and pieces into a data set ;
 create CV_Dat from Z[colname=varNames]   ; * try to select only the last components? ;
 append from Z ;
 close CV_Dat ;
run ;

****************************************************** ;
****************************************************** ;

* merge combineDat with variance components in wide format ;
* calculate the remaining total variance and cv ;
data BigDat&index ;
retain DepVar  Mean_MB Mean_Naive TotVarCov TotVariance Lower Upper TotSD N DF SE ;
merge Var_Dat
      PC_Dat ( drop = Mean_MB Mean_Naive TotVarCov TotVariance Lower Upper TotSD N DF SE )
      SD_Dat ( drop = Mean_MB Mean_Naive TotVarCov TotVariance Lower Upper TotSD N DF SE )
      CV_Dat ( drop = Mean_MB Mean_Naive TotVarCov TotVariance Lower Upper TotSD N DF SE )
      ; * no by-statement since only one line - add later with multiple groups ;

* Creating percent contribution ;
PC_TotVariance      = 100                                          ;

*** conditional if logged or not logged *** ;
if upcase(&logged) = "NO" then do ;
 * Calculating percent CV (Unlogged) - using model-based mean ;
 CV_TotSD            = 100 * ( TotSD / Mean_MB )               ;
end ;
```

```
 else do ;
  * Calculating percent CV (log10) - mean does not matter here ;
  CV_TotSD            = 100 * ( sqrt( 10**(TotVariance * log(10)) - 1 ) ) ;
 end ;
 *** end conditional *** ;

 * Defining Dependent Variable ;
 length DepVar  $15. ;
 DepVar  = "&DepVar " ;

 * Defining Level ;
 ****** Changed to levelvar for merging with the final dataset.  ******;
 LevelVar = &index ;

 RUN ;
%END ;
* putting the datasets together ;
%macro combine ;

 data merged;
 set
 %do i = 1 %to &ByVarLevels ;
   BigDat&i
 %end ;
 ;
 run ;
%mend combine ;
%combine ;

****** Merge "by" variables into the final dataset.  ******;
data &OutData;
    merge vars merged;
    by LevelVar;
run;

****** Test equivalency for total variances between instruments.  ******;
%if &EquTest = Y %then %do;
    proc freq data=DataIn;table &instype/list out=freqinst(drop=count percent); run;

    data _null_;
        set freqinst end=eof;
        if eof = 1 then call symput('FinalObs',_N_);
    run;

    %do i=1 %to &finalobs;
        proc sort data=&outdata (keep=&byvars n mean_mb totvariance df) out=inst&i
(rename=(n=n&i mean_mb=mean&i._md df=df&i totvariance=totvar&i));
         by &byvars;
         where &instype = &i;
        run;
    %end;

    data &outdata._ratio (drop=&instype);
        retain levelvar &byvars ratio ci_l_95 ci_u_95;
        set inst1; set inst2;

        levelvar = _N_;
```

```
        ratio=round(totvar2/totvar1,.0001);

        ci_l_95=ratio*(1/finv((1-(&alpha/2)),(df2-1),(df1-1)));
        ci_u_95=ratio*(1/finv((&alpha/2),(df2-1),(df1-1)));

        label levelvar = 'Level of Variables'
              ci_l_95 = 'Lower Limit'
              ci_u_95 = 'Upper Limit'
              ratio = 'Ratio of Total Variance2 vs. Variance1';
    run;

%end;
%mend PrecMod ;
* END MACRO CODE APPENDIX A;
```

## APPENDIX B. Additional SAS code for formatting results from PrecMod SAS Macro

```
data Intext ;
        set ResultsDat ;

        * Contribution to Total Variance(%) ;
        PC_DSLOT1       = put(PC_DSLOT   ,4.1)    ||'%' ;
        PC_SITE1        = put(PC_SITE    ,4.1)    ||'%' ;
        PC_OPERATOR1    = put(PC_OPERATOR,4.1)    ||'%' ;
        PC_DAY1         = put(PC_DAY     ,4.1)    ||'%' ;
        PC_RESIDUAL1    = put(PC_RESIDUAL,4.1)    ||'%' ;
        PC_TotVariance1 = put(PC_TotVariance,4.1)||'%' ;

        * Standard Deviation Units ;
        SD_DSLOT1       = put(SD_DSLOT   ,8.4) ;
        SD_SITE1        = put(SD_SITE    ,8.4) ;
        SD_OPERATOR1    = put(SD_OPERATOR,8.4) ;
        SD_DAY1         = put(SD_DAY     ,8.4) ;
        SD_RESIDUAL1    = put(SD_RESIDUAL,8.4) ;
        TotSD1          = put(TotSD      ,8.4) ;

        * Variance Units ;
        VAR_DSLOT1      = put(VAR_DSLOT   ,8.4) ;
        VAR_SITE1       = put(VAR_SITE    ,8.4) ;
        VAR_OPERATOR1   = put(VAR_OPERATOR,8.4) ;
        VAR_DAY1        = put(VAR_DAY     ,8.4) ;
        VAR_RESIDUAL1   = put(VAR_RESIDUAL,8.4) ;
        TotVariance1    = put(TotVariance ,8.4) ;

        * Coefficient of Variation (CV) Units ;
        CV_DSLOT1      = put(CV_DSLOT   ,4.1) ||'%' ;
        CV_SITE1       = put(CV_SITE    ,4.1) ||'%' ;
        CV_OPERATOR1   = put(CV_OPERATOR,4.1) ||'%' ;
        CV_DAY1        = put(CV_DAY     ,4.1) ||'%' ;
        CV_RESIDUAL1   = put(CV_RESIDUAL,4.1) ||'%' ;
        CV_TotSD1      = put(CV_TotSD   ,4.1) ||'%' ;

        *Total Precision-STD;
        p_TotSD1 =  put(TotSD1,6.2)||' ('||put(CV_TotSD1,4.1)||'%)';

        * Expected concentration ;
        if titer_n=50.001 then
          logexp1 = put(logexp,5.3)||'**' ;
        else logexp1 = put(logexp,5.3)       ;
run ;

*-----------------------------------------------------------------
Output Reports:
----------------------------------------------------------------- ;

* Percent Contribution (PC) ;
ods listing close ;
ods escapechar = '~' ;
ods rtf file="C:\SAS\Precision\papers\Results\PrecMod_PC_Example_hiv216.rtf" style=print.chgRTF
bodytitle notoc_data ;

title ; footnote ;

proc report nowd data=Intext split='*'
     style(report)={just=center outputwidth=6.5 in}
     style(lines)=header {font_size=9pt font_face="Arial" font_weight=medium
           background=transparent just=left}
     style(header)=header{font_size=9pt font_face="Arial" font_weight=bold
           background=transparent}
     style(column)=header{font_size=9pt font_face="Arial" font_weight=medium
           background=transparent just=center} ;

     columns ( ("HIV-1 RNA Concentration*(log~{sub 10} cp/mL)" logexp1 Mean_Naive N)
                       ("Random Effects Model Components*Contribution to Total Variance (%)"
               PC_DSLOT1 PC_site1 PC_OPERATOR1 PC_DAY1 PC_RESIDUAL1)
                       ("Total Precision" p_TotSD1)
```

```
              ) ;

     define logexp1 / display "Expected" flow
                      style(header)={just=center}
                      style(column)={just=left protectspecialchars=off pretext="\qj\tqdec\tx350 "
cellwidth=0.60 in} ;

          define Mean_Naive / display "Observed*(average)"  flow f=7.3
                      style(header)={just=center}
                      style(column)={just=center protectspecialchars=off  cellwidth=0.60 in} ;

          define N / display "No. of*Tests~{super a}" flow
                      style(header)={just=center}
                      style(column)={just=center protectspecialchars=off  cellwidth=0.50 in} ;

          define PC_DSLOT1 /display "Lot" flow
                      style(header)={just=center}
                      style(column)={just=center protectspecialchars=off pretext="\qj\tqdec\tx350
" cellwidth=0.40 in} ;

          define PC_SITE1 /display "Site" flow
                          style(header)={just=center}
                          style(column)={just=center protectspecialchars=off
pretext="\qj\tqdec\tx550 " cellwidth=0.60 in} ;

          define PC_OPERATOR1 /display "Operator" flow
                          style(header)={just=center}
                          style(column)={just=center protectspecialchars=off
pretext="\qj\tqdec\tx450 " cellwidth=0.50 in} ;

          define PC_DAY1 /display "Day" flow
                          style(header)={just=center}
                          style(column)={just=center protectspecialchars=off
pretext="\qj\tqdec\tx350 " cellwidth=0.40 in} ;

           define PC_RESIDUAL1 /display "Within-*Day" flow
                          style(header)={just=center}
                          style(column)={just=center protectspecialchars=off
pretext="\qj\tqdec\tx350 " cellwidth=0.40 in} ;

           define p_TotSD1 /display 'Standard Deviation*(Lognormal %CV)~{super b}' flow
                          style(header)={just=center}
                          style(column)={just=center protectspecialchars=off  cellwidth=0.90 in}
;

         title1 j=c bold height=12pt f='Times' "Table X1.  Attributable Percentage of Total
Variance, Total Precision Standard" ;
         title2 j=c bold height=12pt f='Times' "Deviation, and Lognormal %CV of HIV-1 RNA
Concentration (log~{sub 10} cp/mL) from" ;
         title3 j=c bold height=12pt f='Times' "Tests With Detectable Viral Load" ;

     compute after _page_ / style=[protectspecialchars=off] ;
                line "Note: Results with detectable viral load are included in this table." ;
                line " " ;
                line "~{super a }Number of tests with detectable viral load." ;
                line "~{super b }Lognormal %CV = 100% x square root of{10^[variance x ln(10)]-1},
where ln() is the natural logarithm" ;
                line "~{super ** }Results < 3.40E+1 cp/mL were calculated based on the validated
in-house software from ~{unicode 0394}Ct values" ;
                line "~{super    }for test results that were below LLoQ." ;
                line " " ;
                line "Data Source: Appendix X, Table Y.1, Table Y.2." ;
     endcomp;
run ;
ods rtf close ;
ods listing ;
* END MACRO CODE APPENDIX B ;
```