# Supplement 9.1. SAS commands for evaluations summarized in Tables 9.1 and 9.2

*Brian R. Gray, US Geological Survey, Upper Midwest Environmental Sciences Center, La Crosse, Wisconsin, USA.*

This supplement provides SAS commands for evaluating estimators of variance components (VCs) and variance partition components (VPCs) using linear models of data from two-way cross-classification designs. Corresponding routines were used for all scenarios summarized in Tables 9.1 and 9.2 of Chapter 9. A .sas format file containing the following commands is available on request from gitzenr@missouri.edu.

```
* PURPOSE: estimate variance components (VCs) and variance partition coefficients (VPCs) from linear models of data from
              2-way crossed classification designs;
* CREATED: 2 Apr 2010 by Brian Gray;
* REVISIONS: ;


/* THIS ALGORITHM IS PROVIDED WITHOUT WARRANTY, WHETHER EXPRESSED OR IMPLIED.  PLS SEND QUESTIONS AND CRITICISMS TO
 *BRIAN GRAY AT brgray@usgs.gov */


* NOTES:
       * computer code for fitting cross-classified linear models in languages other than SAS (ie, HLM, MLwiN, R, STATA, as
       * well as SAS) using a common dataset is provided at http://www.ats.ucla.edu/stat/sas/examples/mlm_ma_hox/
       * "some cells empty" unbalanced scenarios may (rather crudely) be effected by manually replacing n=1 with n=0 under
       * the "data twoway" data step;


* DEFINE PARAMETER/DESIGN VALUES;
%let var_y = 1;
%let var_site = .3;
%let var_yr = .3;
%let var_sitebyyr = .15;
%let n = 5;
%let sites = 10;
%let years = 5;
%let siteproportion = .3;      * assign proportion sites w/in odd years with n=1 (see note above to move from 'all cells
                               * filled' to 'some cells empty' scenarios;
%let u0normal = 1;             * determine whether random group effects are normal (1) or random uniform (else) distributed;
%let simulations = 50;

options nocenter;
* delete all datasets (prevents using results from a previous scenario when all models for a given method fail to converge);
proc datasets library=work kill; run;
title1 "2-way design: sites = &sites, years=@years, reps = &n. var(yr)=&var_yr, var(site)=&var_site,
       var(sitebyyr)=&var_sitebyyr, var(yijk)=&var_y";


* GENERATE DATA;
* generate year effects;
data yr_effects;
       do sim = 1 to &simulations;
       do year=1 to &years;
              if &u0normal=1 then yr_effect=sqrt(&var_yr)*rannor(123);
                             else yr_effect = sqrt(&var_yr)*(sqrt(12)*ranuni(234) - sqrt(12)/2);
                                                  * random uniform with unit variance;
              do site=1 to &sites;
                     output;
```

```
                end;
        end;
        end;
run;
proc sort data=yr_effects out=yr_effects_sort; by sim site year; run;

* generate site effects;
data site_effects;
        do sim = 1 to &simulations;
        do site = 1 to &sites;
                if &u0normal=1 then site_effect = sqrt(&var_site)*rannor(234);
                        else site_effect = sqrt(&var_site)*(sqrt(12)*ranuni(234) - sqrt(12)/2);
                                        * random uniform with unit variance;
                do year= 1 to &years;
                        if &u0normal=1 then sitebyyr_effect = sqrt(&var_sitebyyr)*rannor(345);
                                else sitebyyr_effect = sqrt(&var_sitebyyr)*(sqrt(12)*ranuni(234) - sqrt(12)/2);
                                        * random uniform with unit variance;
                output;
                end;
  end;
        end;
run;

data twoway;
        merge yr_effects_sort site_effects;
        by sim site year;
        if mod(year, 2) then oddyear=1; else oddyear=0;
        sites=&sites; n=&n; siteproportion=&siteproportion;
        if oddyear and site gt siteproportion*sites then n=1;
        do i=1 to n;
                y = yr_effect + site_effect + sitebyyr_effect + sqrt(&var_y)*rannor(456);
                output;
        end;
run;


* ESTIMATE VARIANCE COMPONENTS BY METHOD;
ods listing close;
* estimate variance components using Type 3 SS's;
proc mixed data=twoway method=type3;* covtest cl;
        class year site;
        model y = ;
        random site year site*year; * slower format;
        by sim;
  ods output covparms=covparmstype3;
run;
```

```
* estimate variance components using ML (default = observed inverse Fisher information matrix);
proc mixed data=twoway method=ml;* covtest cl;
        class year site;
        model y = ;
        random int / sub=site;
        random int / sub=year;
        random int / sub=year*site;
        by sim;
  ods output covparms=covparmsml;
run;

* estimate variance components using REML (default = observed inverse Fisher information matrix);
proc mixed data=twoway method=reml;* covtest cl;
        class year site;
        model y = ;
        random int / sub=site;
        random int / sub=year;
        random int / sub=year*site;
        by sim;
  ods output covparms=covparmsreml;
run;
ods listing;


* CALCULATE SUMMARY STATISTICS;
* append datasets;
data covparms_row;
        set covparmstype3(in=type3) covparmsml(in=ML) covparmsreml(in=REML);
        if type3 then method="1Type3"; else if ML then method="2ML"; else method="3REML";
        if covparm="Residual" then covparm="z_residual";
        if method in ("2ML", "3REML") then covparm=subject;
        if method in ("2ML", "3REML") and subject="" then covparm="z_residual";
        * variance assignments/calculations;
        varhat=estimate; if method="1Type3" and estimate lt 0 then varhat=0;
        if 0 le estimate lt 1E-4 then varhat0=1; else varhat0=0;
        if estimate < 0 then varhatneg=1; else varhatneg=0;
        SDhat=sqrt(varhat);
        * precision assignment;
        *SEhat=stderr; * requires unremarking covtest options;
        * calculate VPC estimates;
        VPC_site = lag3(varhat) / (varhat + lag(varhat)+lag2(varhat) + lag3(varhat));
        if covparm ne "z_residual" then VPC_site=.;
        VPC_yr = lag2(varhat) / (varhat + lag(varhat)+lag2(varhat) + lag3(varhat));
        if covparm ne "z_residual" then VPC_yr=.;
        VPC_siteyr = (lag(varhat)+lag2(varhat) + lag3(varhat)) / (varhat + lag(varhat)+lag2(varhat) + lag3(varhat));
        if covparm ne "z_residual" then VPC_siteyr=.;
        keep sim method covparm varhat varhat0 varhatneg SDhat VPC_site VPC_yr VPC_siteyr;* SEhat;
```

```
run;
proc sort data=covparms_row out=covparms_row_sort; by covparm method sim; run;

* calculate variance component stats;
title1 "2-way design: sites = &sites, years=&years, n = &n, var_yij=&var_y, var(yr)=&var_yr, var(site)=&var_site,
var(sitebyyr)=&var_sitebyyr, sims=&simulations";
title2 "Variance component stats";
proc means data=covparms_row_sort noprint;
        var varhat SDhat varhat0 varhatneg /*SEhat*/ ;
        output out=varcompstats mean=meanvarhat meanSDhat meanvarhat0 meanvarhatneg /*meanSEvarhat*/ stddev=SE_varhat SE_SDhat
                /*SESEvarhat*/;
        by covparm method;
run;
* add pct convergence;
data varcompstats2;
        set varcompstats(drop=_TYPE_);
        pctconverge=_freq_/&simulations;
proc print data=varcompstats2 noobs;
        format _numeric_ 8.3 pctconverge 8.2 _freq_;
        var covparm method _freq_ pctconverge meanvarhat /*meanSEvarhat*/ SE_varhat meanvarhat0 meanvarhatneg meanSDhat
                SE_SDhat;
run;

* calculate VPC stats;
title2 "VPC stats";
proc means data=covparms_row noprint;
        var VPC_site VPC_yr VPC_siteyr;
        output out=VPCstats mean=meanVPCsite meanVPCyr meanVPCsiteyr stddev=SE_VPCsite SE_VPCyr SE_VPCsiteyr
                median=medVPCsite medVPCyr medVPCsiteyr qrange= iqrVPCsite iqrVPCyr iqrVPCsiteyr;
        by method;
        where covparm = "z_residual";
run;
proc print data=VPCstats noobs;
        format _numeric_ 8.2 _freq_;
        var method _freq_ meanVPCsite SE_VPCsite medVPCsite iqrVPCsite meanVPCyr SE_VPCyr medVPCyr iqrVPCyr meanVPCsiteyr
                SE_VPCsiteyr medVPCsiteyr iqrVPCsiteyr;
run;


* PLOT ESTIMATES;
* plot variance/SD estimates;
proc boxplot data=covparms_row_sort; plot SDhat*method; by covparm; run;
* plot VPC estimates;
proc boxplot data=covparms_row; plot (VPC_site VPC_yr VPC_siteyr)*method; run;
```