Supplement 9.5. SAS commands for evaluations summarized in Tables 9.4-9.5

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

This supplement provides SAS commands for generating variance component (VC) and variance partition component (VPC) estimates on the probability scale (after Goldstein et al. 2002 and Browne et al. 2005) for binary/binomial outcomes for data from two-way cross-classified designs. Estimates are produced using approximation (i.e., the alternate or "differencing" approach presented in Appendix 9.2). Uses input from SAS routines presented in Supplement 9.2. Evaluations using these SAS calculations are summarized in Tables 9.4-9.5. A .sas format file containing the following commands is available on request from gitzenr@missouri.edu.

```
* PURPOSE: generate variance component (VC) and variance partition coefficient (VPC) estimates from for binary/binomial
outcomes on probability scale (after Goldstein et al 2002 and Browne et al 2005) for data from 2-way cross-classified designs
using approximation (labeled "differencing" (see Appendix 9.2).  Uses input from SAS routines presented in Supplement 9.2;
* CREATED: 5 Jul 2010 by Brian Gray from analogous "user input" sas file;

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

* notes:
*       - this code requires a dataset in the user's SAS work directory that contains variance component estimates (on the
*              logit scale, eg
*       - this code generates variance components from a random effects model only. Goldstein et al 2002 postulate and Browne
*              et al 2005 demonstrate VPCs with fixed covariates
*       - this code calculated var(p_ab) by differencing on the probability scale.  While differencing is standard
*              (cf Li et al 2008), this method furthers this approach. Whether estimation of var(p_a) by marginal var(p_ab)
*              less marginal var(p_b) and similar for var(p_b) is reasonable under some or any settings is an open question.
*              A method that estimates var(p_ab) using replication for a single set of variance components is provided by
*              in Supplement 9.4.
*       - p_ab is a population value and hence variances of f(p_ab) don't require correction for sampling variation.
*              See Wang and Gelfand (2002) for an alternative;


* ENTER PARAMETER/DESIGN VALUES;
%let maxmodelsim = 500;                                 * define maximum number of external estimates to use;
%let groups = 100;                                      * define number of simulated random effects per set of model estimates;
%let filename = covparmsplus;  * supply name of file (in work directory) containing variance component estimates;

options nocenter;
title1 "2-way factorial design: sites = &sites, years=&years, index = &n. var(yr)=&var_yr, var(site)=&var_site,
var(sitebyyr)=&var_sitebyyr, sims=&simulations";
title2 "Var component and VPC estimates on probability scale. Groups per replicated dataset = &groups.";


* GENERATE INPUT DATASET;
data A;
        set &filename;
        varAhat=varhat;
        where subject="site" and status ne 1 and sim le &maxmodelsim;
        keep method sim varhat varAhat inthat reason status pdG;
run;
data B;
        set &filename;
        varBhat=varhat;
        where subject="year" and status ne 1 and sim le &maxmodelsim;
        keep method sim varhat varBhat;
run;
data C;
```

```
        set &filename;
        varABhat=varhat;
        where subject="year*site" and status ne 1 and sim le &maxmodelsim;
        keep method sim varhat varABhat;
run;
data allvar;
        merge A B C;
        by method sim;
run;


* SIMULATE RANDOM EFFECTS;
* generate A effects;
data A_effects;
        set allvar;
        do a = 1 to &groups;
                u0a = sqrt(varAhat)*rannor(123);
                do b = 1 to &groups;
                        output;
                end;
        end;
run;
proc sort data=A_effects out=A_effects_sort; by method sim b a; run;

* generate B and AB effects;
data B_effects;
        set allvar;
        do b = 1 to &groups;
                u0b = sqrt(varBhat)*rannor(234);
                do a = 1 to &groups;
                        u0ab = sqrt(varABhat)*rannor(345);
      output;
                end;
  end;
run;

* combine A and (B and AB) effects;
data twoway;
        merge A_effects_sort B_effects;
        eta_ab = inthat + u0a + u0b + u0ab;    * calculate subject-specific mean on logit scale;
        p_ab=(1+exp(-eta_ab))**(-1); * calculate SS mean on probability scale;
        var_y_abk = p_ab*(1-p_ab); * variance of kth Bernoulli outcome, k = 1, ..., n, in cell ab;
        by method sim b a;
run;


* ESTIMATE GROUP LEVEL VARIANCE ESTIMATES ON PROBABILITY SCALE;
```

```
* Estimate uncorrected (marginal) among-a(b) variance estimates on probability scale;
proc means data=twoway noprint;
        var p_ab u0a;
        output out=margvar_p_aINb var=margvar_p_aINb var_u0a;
        by method sim b;
run;
* average among-a(b) variance estimates over b (irrelevant for u0a);
proc means data=margvar_p_aINb noprint;
        var margvar_p_aINb var_u0a;
        output out=meanmargvar_p_aINb mean=meanmargvar_p_aINb meanvar_u0a;
        by method sim;
run;

* Estimate uncorrected (marginal) among-b(a) variance estimate on probability scale;
proc sort data=twoway out=twowaysort; by method sim a b; run;
proc means data=twowaysort noprint;
        var p_ab u0b;
        output out=margvar_p_bINa var=margvar_p_bINa var_u0b;
        by method sim a;
run;
* average among-b(a) variance estimates over a (irrelevant for u0a);
proc means data=margvar_p_bINa noprint;
        var margvar_p_bINa var_u0b;
        output out=meanmargvar_p_bINa mean=meanmargvar_p_bINa meanvar_u0b;
        by method sim;
run;

* Estimate uncorrected (marginal) among-(ab) variance components, mean cell probability and
*       mean sampling variance (for a Bernoulli outcome);
proc means data=twoway noprint;
        var p_ab u0ab var_y_abk;
        output out=margvar_p_ab var=margvar_p_ab var_u0ab mean=mean_p_ab mean_u0ab mean_s2;
        by method sim;
run;


* ESTIMATE VARIANCE COMPONENT STATISTICS ON LOGIT SCALE;
* combine datasets. calculate variances and VPCs on probability scale;
data allests;
        merge meanmargvar_p_aINb(keep=method sim meanmargvar_p_aINb meanvar_u0a _freq_ rename=_freq_=n_main)
                meanmargvar_p_bINa(keep=method sim meanmargvar_p_bINa meanvar_u0b)
                margvar_p_ab(keep=method sim margvar_p_ab var_u0ab mean_p_ab mean_s2 _freq_ rename=_freq_=n_int);
        var_p_a = margvar_p_ab - meanmargvar_p_bINa;
        var_p_b = margvar_p_ab - meanmargvar_p_aINb;
        var_p_ab = meanmargvar_p_aINb + meanmargvar_p_bINa - margvar_p_ab;
        * calculate VPCs;
        VPC_A = var_p_a / (var_p_a + var_p_b + var_p_ab + mean_s2);
```

```
        VPC_B = var_p_b / (var_p_a + var_p_b + var_p_ab + mean_s2);
        VPC_AB = (var_p_a + var_p_b + var_p_ab) / (var_p_a + var_p_b + var_p_ab + mean_s2);
        * calculate VPCs under latent logistic RV assumption;
        s2_L = constant('pi')**2/3;
        VPC_A_L = meanvar_u0a / (meanvar_u0a + meanvar_u0b + var_u0ab + s2_L);
        VPC_B_L = meanvar_u0b / (meanvar_u0a + meanvar_u0b + var_u0ab + s2_L);
        VPC_AB_L = (meanvar_u0a + meanvar_u0b + var_u0ab) / (meanvar_u0a + meanvar_u0b + var_u0ab + s2_L);
        by method sim;
run;
title3 "Variance estimates on logit scale";
proc means data=allests noprint;
        var meanvar_u0a meanvar_u0b var_u0ab;
        output out=varstatslogit mean=gmeanvar_u0a gmeanvar_u0b meanvar_u0ab stddev=SDgmeanvar_u0a SDgmeanvar_u0b
                SDmeanvar_u0ab n=sims;
        by method;
run;
proc print data=varstatslogit noobs;
        format _numeric_ 8.4 sims;
        var method gmeanvar_u0a SDgmeanvar_u0a gmeanvar_u0b SDgmeanvar_u0b meanvar_u0ab SDmeanvar_u0ab sims;
run;


* ESTIMATE VARIANCE COMPONENT STATISTICS ON PROBABILITY SCALE;
title3 "Variance component estimates on probability scale";
proc means data=allests noprint;
        var mean_p_ab mean_s2 var_p_a var_p_b var_p_ab;
        output out=varstatsprob mean=gmean_p_ab gmean_s2 meanvar_p_a meanvar_p_b meanvar_p_ab stddev=SDgmean_p_ab
                SDgmean_s2 SDmeanvar_p_a
                SDmeanvar_p_b SDmeanvar_p_ab n=sims;
        by method;
run;
proc print data=varstatsprob noobs;
        format _numeric_ 8.4 sims;
        var method meanvar_p_a SDmeanvar_p_a meanvar_p_b SDmeanvar_p_b meanvar_p_ab SDmeanvar_p_ab gmean_s2
                SDgmean_s2 gmean_p_ab SDgmean_p_ab sims;
run;


* ESTIMATE VARIANCE PARTITION COEFFICIENTS;
title3 "VPC estimates by simulation and latent variable ('L') methods";
proc means data=allests noprint;
        id s2_L;
        var VPC_A VPC_B VPC_AB VPC_A_L VPC_B_L VPC_AB_L;
        output out=VPCstats mean=meanVPC_A meanVPC_B meanVPC_AB meanVPC_A_L meanVPC_B_L meanVPC_AB_L
                stddev=SDmeanVPC_A SDmeanVPC_B
                SDmeanVPC_AB SDmeanVPC_A_L SDmeanVPC_B_L SDmeanVPC_AB_L n=sims;
        by method;
```

```
run;
proc print data=VPCstats noobs;
        format _numeric_ 8.4 sims;
        var method meanVPC_A SDmeanVPC_A meanVPC_B SDmeanVPC_B meanVPC_AB SDmeanVPC_AB meanVPC_A_L SDmeanVPC_A_L
                meanVPC_B_L SDmeanVPC_B_L meanVPC_AB_L SDmeanVPC_AB_L s2_L sims;
run;
```