Supplement 9.4. Estimating variance components on the probability scale for binary/binomial outcomes

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) estimates on the probability scale from user-specified values for binary/binomial outcomes for data from two-way cross-classified designs (after Goldstein et al. 2002 and Browne et al. 2005). The calculations implemented correspond to steps 1-6 in Appendix 9.2. 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 user-specified values 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 (cf., Appendix 9.2);
* CREATED: 28 Apr 2010 by Brian Gray;

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


* notes:
*       - 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 replicates on ab, and then estimates var_ab from w/in ab cells. estimates may also be obtained by taking
*             differences among variance component means (w/o replicating).
*             however, differencing risks running afoul of Jensen's inequality (and, for a ltd set of scenarios, yields
*             estimates that differ from this replication approach)
*       - p_ab is a population value and hence variances of f(p_ab) don't require correction for sampling variation;


* ENTER PARAMETER/DESIGN VALUES;
%let logitmu = -2;            * supply logit-scale estimate of median;
%let varA = 1;                * supply logit-scale estimate;
%let varB = 0.5;              * supply logit-scale estimate;
%let varAB = 0.25;            * supply logit-scale estimate;
%let sims = 1000;             * supply number of a and b effects;
%let absims = 15;             * supply number of ab effects per ab cell (a device to allow direct estimation of var(ab));

options nocenter;
proc datasets library=work kill; run;
       * delete all datasets (prevents using results from previous scenario when all models for a given method
       *fail to converge);
title1 "Var component estimates, 2-way design. var_p_ab by replication. logit(mu)=&logitmu, var(A)=&varA, var(B) = &varB,
       var(AB)=&varAB, sims=&sims";


* GENERATE DATA;
* generate A effects;
data A_effects;
       do a = 1 to &sims;
              u0a = sqrt(&varA)*rannor(123);
              do b = 1 to &sims;
                     output;
              end;
       end;
run;
proc sort data=A_effects out=A_effects_sort; by b a; run;
```

```
* generate B and AB effects;
data B_effects;
        do b = 1 to &sims;
                u0b = sqrt(&varB)*rannor(234);
                do a = 1 to &sims;
                        do ab = 1 to &absims;
                        * replication here is a concession to estimation below, and does not imply analogous replication in a
                        *typical space-time design;
                                u0ab = sqrt(&varAB)*rannor(345);
        output;
                        end;
        end;
        end;
run;

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


* ESTIMATE GROUP LEVEL VARIANCE ESTIMATES ON PROBABILITY SCALE;
* estimate marginal among-a(b) variance on probability scale;
title3 "marginal among-a(b) variance estimate on probability scale";
proc means data=twoway noprint;
        var p_ab u0a;
        output out=varmarg_p_aINb var=varmarg_p_aINb var_u0a;
        by b;
        where ab=1;
run;
proc means data=varmarg_p_aINb noprint;
        var varmarg_p_aINb var_u0a;
        output out=meanvarmarg_p_aINb mean=meanvarmarg_p_aINb meanvar_u0a;
run;
/*proc print data=meanvarmarg_p_aINb noobs; run; */

* estimate marginal among-b(a) variance on probability scale;
title3 "marginal among-b(a) variance estimate on probability scale";
proc sort data=twoway out=twowaysort; by a b; where ab=1; run;
proc means data=twowaysort noprint;
        var p_ab u0b;
        output out=varmarg_p_bINa var=varmarg_p_bINa var_u0b;
        by a;
```

```
run;
proc means data=varmarg_p_bINa noprint;
        var varmarg_p_bINa var_u0b;
        output out=meanvarmarg_p_bINa mean=meanvarmarg_p_bINa meanvar_u0b;
run;
/*proc print data=meanvarmarg_p_bINa noobs; run; */

* estimate among-ab variance on probability scale (using replicates);
title3 "Among-ab variance estimate on probability scale";
proc means data=twoway noprint;
        var p_ab u0ab;
        output out=var_p_ab var=var_p_ab var_u0ab;
        by b a;
run;
proc means data=var_p_ab noprint;
        var var_p_ab var_u0ab;
        output out=meanvar_p_ab mean=meanvar_p_ab meanvar_u0ab;
run;
/*proc print data=meanvar_p_ab noobs; run; */

* estimate mean_p.. and mean_s2
*       (could estimate using previous means statements but conditioning on ab=1 avoids liberal precisions);
title3 "Mean p.. and mean_s2";
proc means data=twoway noprint;
        var p_ab var_y_abk;
        output out=mean_p_ab mean=mean_p_ab mean_s2 median=median_p_ab;
        *by b a;
        where ab=1;
run;
proc print data=mean_p_ab noobs; run;


* CALCULATE, PRINT ADJUSTED VARIANCE ESTIMATES ON PROBABILITY SCALE;
title3 "Group-level variance estimates on probability scale";
data allvarests;
        merge meanvarmarg_p_aINb(keep=meanvarmarg_p_aINb meanvar_u0a _freq_ rename=_freq_=n_main)
                meanvarmarg_p_bINa(keep=meanvarmarg_p_bINa meanvar_u0b)
                meanvar_p_ab(keep=meanvar_p_ab meanvar_u0ab)
                mean_p_ab(keep = mean_p_ab median_p_ab mean_s2 _freq_ rename=_freq_=n_int);
        var_p_a = meanvarmarg_p_aINb - meanvar_p_ab;
        var_p_b = meanvarmarg_p_bINa - meanvar_p_ab;
proc print noobs; format _numeric_ 8.4 n_main n_int;
        var var_p_a var_p_b meanvar_p_ab mean_p_ab median_p_ab mean_s2 meanvar_u0a meanvar_u0b meanvar_u0ab n_main n_int; run;
```