Supplement 9.6. Estimating variance components and variance partition components on the probability scale for binary/binomial outcomes using the "differencing" approach

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) from user-specified values for binary/binomial outcomes for data from two-way cross-classified designs. The calculations implemented correspond to steps 1-6 in Appendix 9.2, with alternate steps 4 and 5 used. 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 using approximation (labeled "differencing", Appendix 9.2);
* CREATED: 28 Apr 2010 by Brian Gray;

/* THIS ALGORITHM IS PROVIDED WITHOUT WARRANTY, WHETHER EXPRESSED OR IMPLIED.  PLS SEND QUESTIONS OR 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
*	- p_jk is a population value and hence variances of f(p_jk) don't require correction for sampling variation.  see Wang
*		and Gelfand (2002) for an alternative
*	- Li et al 2008 calculate the L2 variance as the mean of the w/in-L3 unit sample variances, and the L3 variance as the
*		marginal variance at L2 [ie the analogue of var(pjk-p_bar)] _less_ the L2 variance. regardless, it seems
*		sensible to question whether estimation of var(p_j)^ as mean var(p_j|k) less mean var(p_jk|j,k)
*		and similar for var(p_k) (as done in this algorithm) is reasonable.  the differencing idea is
*		corroborated by Li et al.  however, averaging over "the other" main effect may introduce bias,
*		or at least the perception of bias. var(p) is a function of the mean, becoming smaller as the mean becomes
*		more extreme. p(.) become more skewed as mean p_jk decreases - assumes biases in marginal var(p_j|k)^
*		and var(p_jk) are similar--this may not be true
*	- yields smaller estimates, as expected (ie w/o conditioning), than when estimated separately using var(p) from
*		 var(logit(p)).sas
*	- with &sims=5000 took ~30 min on 2 GHz, 2 GB RAM laptop (albeit with other processes up);



* ENTER PARAMETER/DESIGN VALUES;
options nocenter;
%let logitmu = -2;	* supply logit-scale estimate of grand intercept;
%let varA =1;				* supply var(A) estimate on logit scale;
%let varB =0.5;			* supply var(B) estimate on logit scale;
%let varAB =0.25;			* supply var(AB) estimate on logit scale;
%let sims = 2000;			* define number of simulations;
title1 "Var component estimates, 2-way design, var_p_jk by difference. logit(mu)=&logitmu, var(A)=&varA, var(B) = &varB,
	var(AB)=&varAB, sims=&sims";


* GENERATE DATA;
* generate A effects;
data A_effects;
	do j = 1 to &sims;
		u0j = sqrt(&varA)*rannor(123);
		do k = 1 to &sims;
			output;
		end;
	end;
```

```
run;
proc sort data=A_effects out=A_effects_sort; by k j; run;

* generate B and AB effects;
data B_effects;
	do k = 1 to &sims;
		u0k = sqrt(&varB)*rannor(234);
		do j = 1 to &sims;
			u0jk = sqrt(&varAB)*rannor(345);
      output;
		end;
  end;
run;

* combine A and (B and AB) effects;
data twoway;
	merge A_effects_sort B_effects;
	eta_jk = &logitmu + u0j + u0k + u0jk; * calculate subject-specific mean on logit scale;
	p_jk=(1+exp(-eta_jk))**(-1); * calculate jkth mean on probability scale;
	var_y_ijk = p_jk*(1-p_jk); * variance of ith Bernoulli outcome, i = 1, ..., n, in cell jk;
	by k j;
run;


* ESTIMATE GROUP LEVEL VARIANCE ESTIMATES ON PROBABILITY SCALE;
* Estimate uncorrected (marginal) among-j(k) variance estimate on probability scale;
proc means data=twoway noprint;
	var p_jk u0j;
	output out=margvar_p_jINk var=margvar_p_jINk var_u0j;
	by k;
run;
proc means data=margvar_p_jINk noprint;
	var margvar_p_jINk var_u0j;
	output out=meanmargvar_p_jINk mean=meanmargvar_p_jINk meanvar_u0j;
run;
/*proc print data=meanmargvar_p_jINk(drop=_TYPE_) noobs; run; */

* Estimate uncorrected (marginal) among-k(j) variance estimate on probability scale;
proc sort data=twoway out=twowaysort; by j k; run;
proc means data=twowaysort noprint;
	var p_jk u0k;
	output out=margvar_p_kINj var=margvar_p_kINj var_u0k;
	by j;
run;
proc means data=margvar_p_kINj noprint;
	var margvar_p_kINj var_u0k;
	output out=meanmargvar_p_kINj mean=meanmargvar_p_kINj meanvar_u0k;
```

```
run;
/*proc print data=meanmargvar_p_kINj(drop=_TYPE_) noobs; run; */

* Estimate uncorrected (marginal) among-(jk) variance components, mean cell probability and mean sampling variance (for a
        Bernoulli outcome);
proc means data=twoway noprint;
        var p_jk u0jk var_y_ijk;
        output out=margvar_p_jk var=margvar_p_jk var_u0jk mean=mean_p_jk mean_u0jk mean_s2 median=median_p_jk;
run;
/*proc print data=margvar_p_jk noobs; format _numeric_ 8.4 n_main n_jk; var _FREQ_ margvar_p_jk varu0jk mean_p_jk
        median_p_jk mean_s2; run;*/


* CALCULATE AND PRINT ADJUSTED VARIANCE ESTIMATES ON PROBABILITY SCALE;
title2 "Variance estimates on probability scale";
data allvarests;
        merge meanmargvar_p_jINk(keep=meanmargvar_p_jINk meanvar_u0j _freq_ rename=_freq_=n_main)
                meanmargvar_p_kINj(keep=meanmargvar_p_kINj meanvar_u0k)
                margvar_p_jk(keep = margvar_p_jk var_u0jk mean_p_jk median_p_jk mean_s2 _freq_ rename=_freq_=n_int);
        var_p_j = margvar_p_jk - meanmargvar_p_kINj;
        var_p_k = margvar_p_jk - meanmargvar_p_jINk;
        var_p_jk = meanmargvar_p_jINk + meanmargvar_p_kINj - margvar_p_jk;
proc print data=allvarests noobs;
        format _numeric_ 8.4 n_main n_int;
        var var_p_j var_p_k var_p_jk mean_p_jk median_p_jk mean_s2 meanvar_u0j meanvar_u0k var_u0jk n_main n_int;
run;

* CALCULATE AND PRINT VARIANCE PARTITION COMPONENTS;
title2 "VPC estimates by simulation and latent variable ('L') methods";
data VPC;
        set allvarests;
        s2_L=constant('pi')**2/3;
        VPC_A = var_p_j / (var_p_j + var_p_k + var_p_jk + mean_s2);
        VPC_B = var_p_k / (var_p_j + var_p_k + var_p_jk + mean_s2);
        VPC_AB = (var_p_j + var_p_k + var_p_jk) / (var_p_j + var_p_k + var_p_jk + mean_s2);
        * calculate VPCs under latent logistic RV assumption;
        VPC_A_L = meanvar_u0j / (meanvar_u0j + meanvar_u0k + var_u0jk + s2_L);
        VPC_B_L = meanvar_u0k / (meanvar_u0j + meanvar_u0k + var_u0jk + s2_L);
        VPC_AB_L = (meanvar_u0j + meanvar_u0k + var_u0jk) / (meanvar_u0j + meanvar_u0k + var_u0jk + s2_L);
proc print data=VPC noobs;
        format _numeric_ 8.3 n_main n_int;
        var VPC_A VPC_B VPC_AB VPC_A_L VPC_B_L VPC_AB_L n_main n_int s2_L;
run;
```