Bootstrapping

Written by Elizabeth Wall-Wieler for child_ses project (Dr. Leslie Roos) on May 6, 2014

Problem: Want to bootstrap estimates of multilevel modelling output to derive new standard errors (based on actual distribution as opposed to theoretical distribution). Siblings must remain together (random sampling is done at the family level).

Solution: Can create and run bootstrap macro in SAS.

Step 1: Create two datasets, one child from each family in each dataset

```
data mb_cohort;
	set project.child_ses_f21_externalizing (keep = ALL VARIABLES REQUIRED FOR THE MODEL);
run;

proc sort data = mb_cohort;
	by sib_group;
run;

data family_first;
	set mb_cohort;
	by sib_group;
	if first.sib_group;
run;

data family_last;
	set mb_cohort;
	by sib_group;
	if last.sib_group;
run;
```

Step 2: set up the Macro

```
/*************************************************************************
HS Grad Estimate Bootstrap – Siblings Risk Assessment
*************************************************************************/
**generate bootstrap sampling by resampling the data.
NOTE: the seed changes with each iteration (or else you would be taking the same sample each time);

%macro bootstrap;
	options nonotes;
	data grad_bstrap; run;  /*blank dataset*/

	%do e = 1 %to 500; /*here insert however many iterations you would like, I have chosen 500*/
```

```
        /*we are going to randomly select (with replacement) individuals from the family_first dataset.
        We will do the same thing with the family_last dataset, since we have specified the same seed
        for each procedure, the individuals from the same families will be selected*/

        proc surveyselect data = family_first out=bootstrap_1 seed = 1789011&e
                method = urs samprate=1 outhits rep=1 noprint;
        run;

        proc surveyselect data = family_last out=bootstrap_2 seed = 1789011&e
                method = urs samprate=1 outhits rep=1 noprint;
        run;
        options notes;

        /*combine the two random samples – this way we have both siblings for each randomly
        selected family*/

        data bootstrap_sample;
                set bootstrap_1 bootstrap_2;
        run;

/*run the model – the estimates of each effect will be stored in the 'bout' dataset*/
ods listing close;
ods output Glimmix.ParameterEstimates=bout;
proc glimmix data = bootstrap_sample noclprint noitprint ic=q pconv=0.00001;
        class sib_group;
        model gradhs_4yr =
                VARIABLE IN THE MODEL
                /dist=binary link=logit ddfm=bw solution;
run;
ods output close;

/*here we combine the new dataset with the previous datasets*/
data grad_bstrap;
        set grad_bstrap bout (keep = estimate effect);
run;

%end;
%mend bootstrap;
```

Step 3 – Call Macro

```
%bootstrap;
```

Step 4 – Transpose last dataset

```
/*we currently have a vertical dataset (effects are presented 500 times with their unique estimates), we
want a horizontal dataset (effects are presented once with 500 estimates)*/
proc sort data = grad_bstrap;
        by effect;
```

```
run;

proc transpose data = grad_bstrap
        out = g1
        prefix = estimate;
        by effect;
run;
```

Step 5 – get the standard deviation of all 500 estimates of each variable in the model – this is the bootstrapped standard error of the estimate

```
data g1_means;
        set g1 (keep = effect estimate:);
        new_standarderror = std(of estimate1-estimate500); /*here you would take the standard
        deviation of the number of iterations initially specified in the macro*/
run;
```

Step 6 – Print the new standard error

```
proc print data = g1_means;
        var effect new_standarderror;
run;
```