Mixed Models for Hierarchical Data

This exercise is designed as a brief introduction to mixed models for hierarchical data. We will examine a data set from the High School and Beyond study concerned with accounting for scores on Math Achievement. The data set contains information for 7,185 students in 160 schools (with between 14—67 students per school) on the following variables:

Level 1 variables (student level)
```
   minority  minority student (0/1)?
   female    female(0/1)?
   ses       student SES (parent education, occupation, income)
   Cses      Centered SES = ses - meanSES
   mathach   Math achievement
```
Level 2 variables (school level)
```
   size      school enrollment
   sector    Public or Catholic: 0=Public, 1=Catholic
   pracad    proportion of students in academic track
   HImnty    more than 40% minority students?
   disclaim  scale of disciplinary climate
   meanSES   school mean of SES
```

Note that the Level 2 variables are measured for each school – they are constant for all students in a given school. The goal of this exercise is to compare simple models using PROC GLM (which treats all observations as independent) with several models using PROC MIXED (which take the clustering by school into account). The statements below are contained in **N:\psy6140\tutorials\hsbmix1-ex.sas**.

1. Read the data into SAS from the file `hsbmix.sas`

```
%include data(hsbmix);
proc print data=hsbmix(obs=20); run;
```

2. Fit a model using PROC GLM predicting `mathach` from `cses` and `school`. Since `school` is a class variable, this allows a separate intercept for each school

```
title 'Model glm1: Fixed-effects with PROC GLM: varying intercepts';
proc glm data=hsbmix;
   class school;
   model mathach = cses school ;
   output out=glm1 p=pred  r=resid;
run;
```

3. We can also allow the slopes for `mathach` on `school` to vary, by including an interaction of `cses * school`

```
title 'Model glm2: PROC GLM: varying intercepts & slopes';
proc glm data=hsbmix;
   class school;
   model mathach = cses | school ;
```

```
   output out=glm2 p=pred  r=resid;
run;
```

As usual, you can help yourself to understand the fitted model by plotting the predicted values. The statements below do this for a subset of the schools. (You could do the same for the output data set `glm1` from the first model.)

```
proc gplot data=glm2;
  where school < 4000;
  plot predict * cses = school/ vaxis=axis1 nolegend hm=1 vm=1;
  symbol1 v=none interpol=join r=100;
  axis1 label=(a=90) order=(0 to 30 by 10);
  label predict='Fitted math achievement';
run;
```

We could continue to add more predictors—either at the student level or school level. However, we should be worried that we are violating the assumption of independence, since all students in the same school share a common educational experience that would cause their residuals to be correlated.

4. As our first mixed model, we will fit a mixed model analog of model `glm1`, which allows intercepts to vary over schools, but treats these as a random factor.

```
title 'Model mix1: PROC MIXED: random intercepts';
proc mixed data=hsbmix noclprint covtest method=reml noitprint;
  class school;
  model mathach = cses / solution ddfm=bw outp=mix1;
  random intercept / sub=school type=un;
run;
```

The details of how the model is specified, and interpretation of the output is left for the lecture, but for now note:

- The MODEL statement specifies only the fixed effects (here, cses)
- The RANDOM statement specifies that the school level intercepts are to be treated as a random factor.
- The OUTP= option gives an output data set containing predicted values suitable for plots. You can use the same plot step as above with mix1 and mix2 (below).

5. A second mixed model, an analog of model `glm2`, contains random components for both intercepts (mathach at cses=0) and slopes for cses.

```
title 'Model mix2: PROC MIXED: random intercepts & slopes';
proc mixed data=hsbmix noclprint covtest method=reml noitprint;
  class school;
  model mathach = cses / solution ddfm=bw outp=mix2;
  random intercept cses / sub=school type=un;
run;
```

Mixed models in R

The same data set is available as a text file on my web site, with the link below. Some of the variables are dummy coded, and it is more convenient to turn them into factors in R. But it is *essential* that `school` be a factor. The script below is available on the lab server as **N:\psy6140\tutorials\hsbmix.R**.

```
hsbmix <- read.table(http://euclid.psych.yorku.ca/www/lab/psy6140/R/hsbmix.dat,
                     header=TRUE)
# make factors
hsbmix$sector <- factor(hsbmix$sector, labels=c("Public", "Catholic"))
hsbmix$female <- factor(hsbmix$female, labels=c("female", "male"))
hsbmix$minority <- factor(hsbmix$minority, labels=c("other", "minority"))
hsbmix$school <- factor(hsbmix$school)
str(hsbmix)
```

As noted above for SAS, you can fit the fixed effects models simply by including the school factor as a predictor in the model. This just uses the ordinary lm() function.

```
mod1 <- lm(mathach ~ cses + school, data = hsbmix)
anova(mod1)

# allow varying slopes too
mod2 <- lm(mathach ~ cses * school, data = hsbmix)
anova(mod2)
```

There are two main packages in R for fitting mixed-effects models: nlme (with the function lme()) and lme4 (with the function lmer()). Here, I'll illustrate the nlme package.

First, use lmList() to calculate the slopes and intercepts for the separate regressions on cses, by sector. Simple boxplots show the differences between the schools in each sector.

```
# plot intercepts and slopes by sector
library(nlme)
cat.mods <- lmList(mathach ~ cses | school, data=hsbmix, subset = sector=="Catholic")
pub.mods <- lmList(mathach ~ cses | school, data=hsbmix, subset = sector=="Public")
# get coefficients
cat.coef <- coef(cat.mods)
pub.coef <- coef(pub.mods)

boxplot(cat.coef[, 1], pub.coef[, 1], main="Intercepts",
      names=c("Catholic", "Public"))
boxplot(cat.coef[, 2], pub.coef[, 2], main="Slopes",
      names=c("Catholic", "Public"))
```

Now, fit some models using lme()

```
lme.1 <- lme(mathach ~ cses,
             random = ~ 1 | school, data=hsbmix)
lme.1

lme.2 <- lme(mathach ~ cses,
             random = ~ cses | school, data=hsbmix)
```

```
lme.2

lme.3 <- lme(mathach ~ meanses*cses + sector*cses,
             random = ~ cses | school, data=hsbmix)
lme.3

# Compare models (should really refit using mtehod="ML")
anova(lme.1, lme.2, lme.3)
```

Finally, effect plots are useful to visualize the fitted models.

```
library(effects)
plot(allEffects(lme.2))
plot(allEffects(lme.3))
```