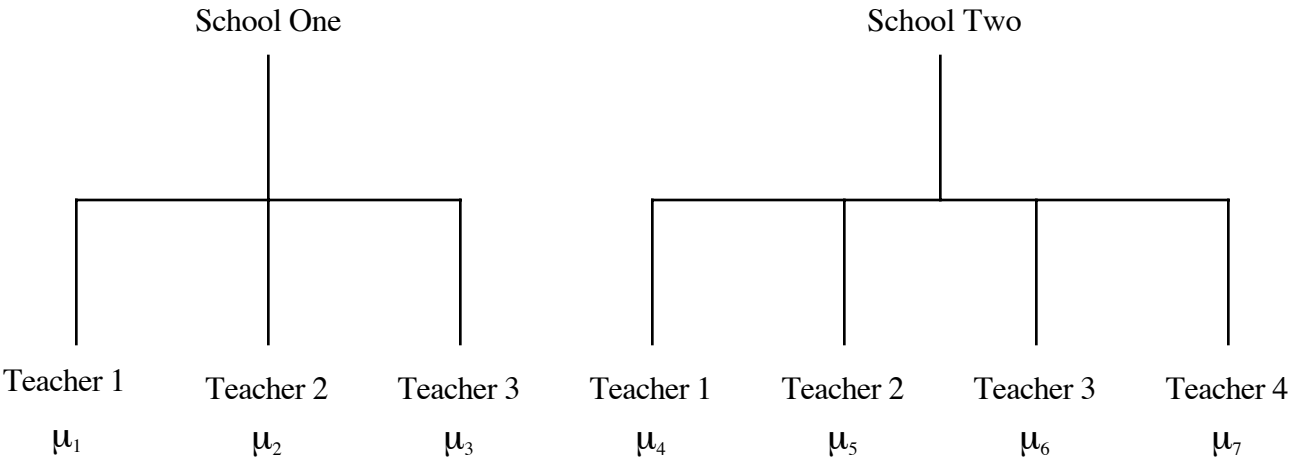
Chapter Four: Nested and Random Effects Models

Nested Designs

Suppose a chain of commercial business colleges is teaching a software certification course. After 6 weeks of instruction, students take a certification exam and receive a score ranging from zero to 100. The owners of the business school chain want to see whether performance is related to which school students attend, or which instructor they have -- or both. They compare two schools; one of the schools has three instructors teaching the course, and the other school has 4 instructors teaching the course. A teacher only works in one school.

There are two independent variables, school and teacher. But it's not a factorial design, because ``Teacher 1" does not mean the same thing in School 1 and School 2; it's a different person. This is called a **nested** design. By the way, it's also **unbalanced**, because there are different numbers of teachers withing each school. We say that *teacher is nested within school*. The diagram below shows what is going on, and give a clue about how to conduct the analysis.



To compare schools, we want to test $\frac{1}{3}(\mu_1+\mu_2+\mu_3) = \frac{1}{4}(\mu_4+\mu_5+\mu_6+\mu_7)$.

To compare instructors within schools, we want to test $\mu_1=\mu_2=\mu_3$ and $\mu_4=\mu_5=\mu_6=\mu_7$ simultaneously.

The first test involves one contrast of μ_1 through μ_7; the second test involves five contrasts. There really is nothing to it.

You can specify the contrasts yourself, or you can take advantage of `proc glm`'s syntax for nested models.

```
proc glm;
     class school teacher;
     model score = school teacher(school);
```

The notation `teacher(school)` should be read ``teacher within school."

- It's easy to extend this to more than one level of nesting. You could have climate zones, lakes within climate zones, fishing boats within lakes, ...

- There is no problem with combining nested and factorial structures. You just have to keep track of what's nested within what. Factors that are not nested are sometimes called ``crossed."

Random Effect Models The preceding discussion (and indeed, the entire course to this point) has been limited to ``fixed effects" models. In a **random effects** model, *the values of the categorical independent variables represent a random sample from some population of values*. For example, suppose the business school had 200 branches, and just selected 2 of them at random for the investigation. Also, maybe each school has a lot of teachers, and we randomly sampled teachers within schools. Then, teachers within schools would be a random effects factor too.

It's quite possible to have random effect factors and fixed effect factors in the same design; such designs are called ``mixed." SAS `proc mixed` is built around this, but it does a lot of other things too.

Nested models are often viewed as random effects models, but there is no necessary connection between the two concepts. It depends on how the study was conducted. Were the two schools randomly selected from some population of schools, or did someone just pick those two (maybe because there are just two schools)?

Random effects, like fixed effects, can either be nested or not; it depends on the logic of the design. An interesting case of nested and purely random effects is provided by **sub-sampling**. For example, we take a random sample of towns, from each town we select a random sample of households, and from each household we select a random sample of individuals to test, or measure, or question.

In such cases the population variance of the DV can truly be partitioned into pieces -- the variance due to towns, the variance due to households within towns, and the variance due to individuals within households. These components of variance can be estimated, and they are, by a program called proc nested, a specialized tool for just exactly this design. All effects are random, and each is nested within the preceding one.

Another example: Suppose we are studying waste water treatment, specifically the porosity of "flocks," nasty little pieces of something floating in the tanks. We randomly select a sample of flocks, and then cut each one up into very thin slices. We then randomly select a sample of slices (called "sections") from each flock, look at it under a microscope, and assign a number representing how porous it is (how much empty space there is in a designated region of the section). The independent variables are flock and section. The research question is whether section is explaining a significant amount of the variance in porosity -- because if not, we can use just one section per flock, and save considerable time & expense.

The SAS syntax for this would be

```
proc sort; by flock section; /* Data must be sorted */
proc nested;
     class flock section;
     var por;
```

The F tests on the output are easy to locate. The last column of output ("Percent of total") is estimated percent of total variance due to the effect. It's fairly close to R^2, but not the same. To include a covariate (say "window"), just use `var window por`; instead of `var por;.` You'll get an analysis of `por` with `window` as the covariate (which is what you want) and an analysis of `window` with `por` as the covariate (which you should ignore).

Of course lots of the time, nothing is randomly selected -- but people use random effects models anyway. Why pretend? Well, sometimes they are thinking that in a better world, lakes *would* have been randomly selected. Or sometimes, the scientists are thinking that they really would like to generalize to the entire population of lakes, and

therefore should use statistical tools that support such generalization -- even if there was no random sampling. (By the way, no statistical method can compensate for a biased sample.) Or sometimes it's just a tradition in certain sub-areas of research, and everybody expects to see random effects models.

In the traditional analysis of models with random or mixed effects and a normal assumption, F-tests are often possible, but they don't always use Mean Squared Error in the denominator of the F statistic. Often, it's the Mean Square for some interaction term or other. The choice of what error term to use is relatively mechanical for balanced models with equal sample sizes (and SAS will do it for you), but even then, sometimes (especially when it's a mixed model) a valid F-test for an effect of interest just doesn't exist.

The following shows how one can obtain classical F tests for random effects and mixed models using `proc glm`. Some things to bear in mind are:

° The interaction of any random factor with another factor (whether fixed or random) is random. But you have to tell proc glm this explicitly.

° You have to tell `proc glm` that you want significance tests, using `/ test`.

° Regardless of what you specify in the random statement, the output from proc glm starts with tests that assume all effects are fixed. If you believe that one or more effects are random, then these tests are meaningless, and should be ignored.

° The tests for random and mixed effects are preceded by expected mean squares, in a notation one can get used to. This part of the output can be a blessing, especially in courses that go into nitty-gritty detail about the classical tests. We will ignore it.

Here is the program `mixed3.sas`, which has no content but shows the syntax.

```
/*************  mixed3.sas ********************
Three levels of factor A, four levels of B
      Pretend both fixed
      Pretend both random
      Pretend A fixed, B random
************************************************/

options linesize=79 noovp formdlim=' ';

data mixedup;
     infile 'ch19pr14.data';
     input Y A garbage B;

/* By default, both are considered fixed */
proc glm;
     title 'Both effects Fixed';
     class A B ;
     model y = a | b;

/* Now both random */
proc glm;
     title 'Both effects random';
     class A B ;
     model y = a | b;
     random a b a*b / test; /* Have to specify interaction random too! */

/* Now A fixed, B random */
proc glm;
     title 'A fixed, B random';
     class A B ;
     model y = a | b;
     random b a*b / test;

/* Now B fixed, A random */
proc glm;
     title 'B fixed, A random';
     class A B ;
     model y = a | b;
     random a a*b / test;
```

Here is the output in `mixed.lst`:

The GLM Procedure

Class Level Information

Class	Levels	Values
A	3	1 2 3
B	4	1 2 3 4

Number of Observations Read 36
Number of Observations Used 36

The GLM Procedure

Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	220.2833333	20.0257576	3.11	0.0097
Error	24	154.4466667	6.4352778		
Corrected Total	35	374.7300000			

R-Square	Coeff Var	Root MSE	Y Mean
0.587845	35.31487	2.536785	7.183333

Source	DF	Type I SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

The GLM Procedure

Class Level Information

Class	Levels	Values
A	3	1 2 3
B	4	1 2 3 4

Number of Observations Read	36
Number of Observations Used	36

The GLM Procedure

Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	220.2833333	20.0257576	3.11	0.0097
Error	24	154.4466667	6.4352778		
Corrected Total	35	374.7300000			

R-Square	Coeff Var	Root MSE	Y Mean
0.587845	35.31487	2.536785	7.183333

Source	DF	Type I SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

The GLM Procedure

Source	Type III Expected Mean Square
A	Var(Error) + 3 Var(A*B) + 12 Var(A)
B	Var(Error) + 3 Var(A*B) + 9 Var(B)
A*B	Var(Error) + 3 Var(A*B)

The GLM Procedure

Tests of Hypotheses for Random Model Analysis of Variance

Dependent Variable: Y

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	220.020000	110.010000	3453.80	<.0001
B	3	0.072222	0.024074	0.76	0.5582
Error: MS(A*B)	6	0.191111	0.031852		

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A*B	6	0.191111	0.031852	0.00	1.0000
Error: MS(Error)	24	154.446667	6.435278		

The GLM Procedure

Class Level Information

Class	Levels	Values
A	3	1 2 3
B	4	1 2 3 4

Number of Observations Read 36
Number of Observations Used 36

The GLM Procedure

Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	220.2833333	20.0257576	3.11	0.0097
Error	24	154.4466667	6.4352778		
Corrected Total	35	374.7300000			

R-Square	Coeff Var	Root MSE	Y Mean
0.587845	35.31487	2.536785	7.183333

Source	DF	Type I SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

A fixed, B random 9

The GLM Procedure

Source	Type III Expected Mean Square
A	Var(Error) + 3 Var(A*B) + Q(A)
B	Var(Error) + 3 Var(A*B) + 9 Var(B)
A*B	Var(Error) + 3 Var(A*B)

A fixed, B random 10

The GLM Procedure

Tests of Hypotheses for Mixed Model Analysis of Variance

Dependent Variable: Y

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	220.020000	110.010000	3453.80	<.0001
B	3	0.072222	0.024074	0.76	0.5582
Error: MS(A*B)	6	0.191111	0.031852		

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A*B	6	0.191111	0.031852	0.00	1.0000
Error: MS(Error)	24	154.446667	6.435278		

The GLM Procedure

Class Level Information

Class	Levels	Values
A	3	1 2 3
B	4	1 2 3 4

Number of Observations Read	36
Number of Observations Used	36

The GLM Procedure

Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	220.2833333	20.0257576	3.11	0.0097
Error	24	154.4466667	6.4352778		
Corrected Total	35	374.7300000			

R-Square	Coeff Var	Root MSE	Y Mean
0.587845	35.31487	2.536785	7.183333

Source	DF	Type I SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	220.0200000	110.0100000	17.09	<.0001
B	3	0.0722222	0.0240741	0.00	0.9997
A*B	6	0.1911111	0.0318519	0.00	1.0000

```
                              B fixed, A random                              13

                              The GLM Procedure

  Source                   Type III Expected Mean Square

  A                        Var(Error) + 3 Var(A*B) + 12 Var(A)

  B                        Var(Error) + 3 Var(A*B) + Q(B)

  A*B                      Var(Error) + 3 Var(A*B)


                              B fixed, A random                              14

                              The GLM Procedure
          Tests of Hypotheses for Mixed Model Analysis of Variance

Dependent Variable: Y

Source                       DF    Type III SS    Mean Square   F Value   Pr > F

A                             2     220.020000     110.010000   3453.80   <.0001
B                             3       0.072222       0.024074      0.76   0.5582

Error: MS(A*B)                6       0.191111       0.031852

Source                       DF    Type III SS    Mean Square   F Value   Pr > F

A*B                           6       0.191111       0.031852      0.00   1.0000

Error: MS(Error)             24     154.446667       6.435278
```

When the design is unbalanced or has unequal sample sizes, the classical approach based on expected mean squares fails, and a valid F-test rarely exists. It's a real pain. Sometimes, you can find an error term that produces a valid F-test *assuming* that some interaction (or maybe more than one interaction) is absent. Usually, you can't test for that interaction either. But people do it anyway and hope for the best.

SAS `proc mixed` goes a long way toward solving these problems. It's a great piece of software, based on recent, state-of the-art research as well as more venerable stuff. Examples will be given in lecture.