Robyn Crook - 2008

Week 6, Week 7 and Week 8 – Analyses of Variance

In the next few weeks we will look at analyses of variance. This is an information-heavy handout so take your time reading it, and don't worry if you are lost! We will take several weeks to cover all of the information here. Once we get through this handout we will have covered all the code steps necessary for completion of the project.

In SAS there are several different methods for performing ANOVAs. Each uses a different procedure command and slight variation in the code, but the code fragments for a one-way fixed-effects anova (proc anova), random effects (proc varcomp) nested (proc nested) or factorial ANOVA (proc GLM) are quite similar. There are some important differences between each, both in terms of their indication and their code. (Do not panic of you don't know what these mean for now – Rocky will cover the theory in the coming weeks).

In the gosize program you will need to perform the appropriate anova for the data you have. In Component 2 you will be using Proc anova and Proc nested. In component 3 we will be using Proc GLM, a particularly powerful analysis tool, to perform factorial anovas. Proc GLM can also be used for simpler analyses too.

This handout covers each of these approaches, so we will refer to it over the following weeks, and you can work ahead if you choose. By now each of you should be familiar with the basics of codes such as 'proc sort', 'proc freq', if/then and data/set statements. You will be using these procedures among your anova codes to create the datasets to be used for each analysis, for sorting your data in order to perform partitioned analyses of variance, and for identifying and selecting which observations to include in your secondary files.

Because you have many examples of code for each of these basic procedures we will concentrate only on the new material.

Preliminary organisation for analyses of variance in the gosize program

Before you perform any ANOVAs, you need to follow the instructions on the project handout to create a dataset that is appropriate. Initially, you need one that contains observations for the year with the largest number of samples, and only includes those with no missing tarsus values (again, pick either tarsus b or tarsus t). To do this you will use a 'data/set' statement with an 'if' statement to screen observations without tarsus data (data gosizeX; set gosizeY; if tarsus_t). To find the year with the largest number of samples, you will use proc freq/tables. (proc freq data=gosizeY; tables bndyr;). Then you will create a further secondary set containing only observations from that largest sampling year. This is the dataset on which you will perform your first anova. Follow the instructions on the project description pages carefully to make sure the datasets you use at each step contain the correct observations.

Proc Anova – analysis of fixed effects.

'Proc anova' is used when you are working with fixed effects (as opposed to random effects, where we use proc varcomp). A 'fixed variable' is one that is assumed to be measured without error, such as treatment levels that we establish ourselves during an experiment. 'Random

variables' are assumed to be values that are drawn from a larger population of possible values. Random variables are commonly categories which we apply to variables we observe but did not create. Most of the time in anovas we assume the independent variables are fixed.

The SAS program makes some assumptions when you use the Proc anova code. The most important one for you to remember is that it assumes the sample sizes in the treatment groups are equal. You will need to ensure your data fit this assumption if you use Proc anova in your program.

Example Code

I will use an example dataset (you will modify this example code to fit into your program) to demonstrate the syntax for proc anova (the 'cattle' example from the past several weeks). In this example we will assume we have been feeding our cows on two different diets, diet 1 or diet 2. These are the treatments, which are fixed effects. Weight is our response variable.

Let's assume we wanted to investigate whether cows fed differently tended to be different weights. So, in other words, we are looking at the effect of diet on weight.

```
Title 'Anova: does weight differ between diets?';
Proc anova data=cattle;
class diet;
model weight=diet;
run;
```

The 'proc anova' command specifies a one-way, or single-factor ANOVA. Remember to specify the correct dataset. The 'class' statement lists the classification variable or the treatment levels. In this example we have a classification variable of 'diet'.

The 'model' statement is where we list the dependent, or response, variable, which in our case is weight. We are effectively telling the program that our null hypothesis (our model) is that weights are equal across the two diets. All effects in the 'model' statement must be shown above in the 'class' statement. Class variables can be either numeric or character. The output from the anova will produce an F statistic and p value based on the hypothesis that there is no difference in weight between the treatment groups.

You could also add an extra line to the Proc anova code to generate separate analyses for another classification variable. For example, in a study of the effect of diet on weight of cattle, it might be important to consider the effect separately for males and females. In this case, we add a 'by' statement to the code. This produces separate analyses for each sex.

```
title 'Anova: does weight differ between diets, separate analysis
for males and females';
Proc anova data=cattle;
class diet;
model weight=diet;
by sex;
run;
```

Some tips for Proc Anova: an important assumption of the proc anova code is that the design is balanced. This means there are equal numbers of subjects in each treatment group. If your sample is not equal for each treatment level, you should either select an equal number of observations for each classification variable (perhaps a random selection), or you could use a different method of running the ANOVA that does not assume a balanced design (proc glm, coming up). If you include a 'by' statement, remember your dataset must be sorted on that 'by' variable.

Proc Varcomp – analysis of random effects

Proc varcomp syntax is essentially the same as for proc anova, the difference is not in the code but in the indication. A proc varcomp is indicated when we want to test the effect of a random variable on a dependent variable. Most real-life analyses are on fixed effects, so proc varcomp is not used that frequently.

Example Code

Using the same example dataset (cattle), we will use proc varcomp to see if there is an effect of 'colour' on weight of cows. 'Colour' acts as a random effect because 1. We didn't make our cows brown, or white, or black, they just came that way, and 2. We are assuming that there are other possible colours a cow can be, but these happen to be the three colours we have (as though we have randomly selected these colours from a wider distribution). Thus 'colour' fits the definition of a random effect. Again, weight is our response variable.

So, here we are looking at the effect of colour on weight.

```
Title 'Anova: does weight differ between colour?';
Proc varcomp data=cattle;
class colour;
model weight=colour;
run;
```

Again, we might choose to perform separate analyses on each sex and each diet, since we can foresee that either of those variables may have an effect on weight too. So we add a 'by' statement with both these variables included.

```
Title 'Anova: does weight differ between colour, separate analyses
for sexes and diets';
Proc varcomp data=cattle;
class colour;
model weight=colour;
by sex diet;
run;
```

This will create four separate anovas, assuming we have two sexes and two diets (one for females on diet 1, one for females on diet 2, one for males on diet 1 and one for males on diet 2). As you can imagine, this can get unwieldy and also can cut the power of the anova down as we reduce the sample size in each analysis.

We can get around this problem by using a factorial anova design that will compute for us whether there is an interaction between colour, diet and sex in a single analysis step.

Proc GLM – factorial anova.

GLM stands for general linear model. It is one of the two most powerful analysis tools in SAS (the other is Proc Mixed, which we will not cover here). Proc GLM is more versatile than either proc anova or proc varcomp, and allows you to perform analyses on effects and interactions between multiple effects, all in a single step (a factorial anova). Proc GLM can also be used for a single, fixed-factor model anova where the groups are not balanced, nested anovas and several other types of analysis.

Example Code

For this example, we will follow on from where we left off with proc varcomp. Instead of producing separate analyses for each sex or each diet, we can use proc GLM to give us statistical support for splitting by, or pooling over, these variables that might confound the results of our analyses. The identification of possible interactions is the basis for a factorial anova.

So, here we are asking the program to look at 1. whether weight varies with colour, 2. whether weight varies with diet, and 3. if the weights of different coloured cows are affected by each diet differently (an interaction of two variables, colour and diet, on weight).

We can list both variables of interest in the 'class' statement, and we specify them again in the 'model' line. This time we also write colour*diet). This will produce separate F statistics and p values for each of these effects and for the interaction term. If we get a significant effect on our interaction term, this tells us we need to split our analysis by either colour or diet to eliminate the interaction. If the p value is not significant, we can pool over the different levels of colour or diet.

```
Title 'Factorial analysis of weight by colour, diet and combination
of colour and diet';
proc glm data=cattle;
class colour diet;
model weight=colour diet colour*diet/ss3;
lsmeans colour diet/stderr pdiff;
run;
```

There are a couple of new things to note in this code. At the end of the 'model' statement we have '/ss3', which is 'sums of squares'. The 3 tells SAS we want to calculate the sums of squares in a particular way (there is also ss1). If you omit this line the output window will list separate F stats and p values for 'sums of squares I' and 'sums of squares III'. The distinction is something you can google for at your leisure. For now, we will use ss3.

The 'lsmeans' statement is also new, and is an abbreviation for 'least-squares means'. The 'stderr pdiff' option at the end requests calculations of p values associated with each possible pair-wise comparison for colours and diets. Multiple pair-wise comparisons are a standard post-hoc test for identifying which levels of each variable are responsible for the overall effect. So, if we had

three colours (black, brown and white), and we knew there was an effect of colour based on the significant p value outputted from our anova, this line of code will show us whether that overall difference is coming from a difference between black and brown and/or black and white, and/or white and brown. Where there are only two levels of treatment, the lsmeans statement tells us little; we already know that if there is a difference it is between level 1 and level 2.

Proc Nested and Proc GLM – nested analyses of variance

A nested analysis is indicated where we have multiple replicates drawn from a single individual or sampling unit. For example, if we were to conduct an experiment where we assay the level of some enzyme in rat lungs, where the rats were given one of two drug treatments, this would be a nested design if we were to take more than one sample from each lung.

For the sake of minimising the rats we need to kill (a common motivation), we might take ten slices from each lung. In this example, we have a treatment of 'drug', a response variable of 'enzyme' and our sampling unit is the rat. In this case, 'slices' are nested within 'lung'. This requires an analysis that reflects the sampling method where slices are not independent if they are derived from the same rat.

There are two main ways to perform a nested anova in SAS, 'proc nested' and 'proc GLM'.

Example Code – Proc nested.

In this example we are testing to see if the enzyme concentration differs between the treatment levels (drug 1 and drug 2). Because we have a nested design we need to analyse with this in mind, so we don't pseudo-replicate by treating every lung slice as an independent observation.

```
Title 'Nested analysis of enzyme concentrations for two drugs using
Proc nested';
proc nested data=drugstudy;
class drug lung slice;
var enzyme;
run;
```

The 'class' statement is a list of the classification variables. SAS assumes that the second class variable is nested in the first, the third is nested within the second and so on. (so, here we have 'slice' which is nested within 'lung', which is nested within 'drug', which is the main effect we are looking for. The 'var' statement is where we list the response variable. In our case, this is enzyme concentration.

The 'proc nested' procedure assumes that the data are balanced (equal numbers in each sampling group), and that the data set is sorted by the class variables (so proc sort data=drugstudy; by drug lung slice; run;)

Example Code – Proc GLM

The other alternative it to use proc GLM, which is better if you suspect your groups are unbalanced. In this code the way we specify nested variables is to place the nesting variable inside parentheses.

```
Title 'Nested analysis of enzyme concentrations for two drugs using
proc GLM';
proc glm data=drugstudy;
class drug lung slice;
model enzyme=drug lung slice(lung)/ss3;
random slice(lung);
run;
```

The 'class' line, like above, has a list of the classification variables. Because we are using Proc GLM which is a general command, (as opposed to proc nested, which is only for nested designs), we have to specify that this is a nested anova by describing the nesting structure. The nested variables go in the 'model' statement. As always with proc GLM, the response variable is on the left, then an equal sign, and then we specify the nested variables. In this case, like above, we have drug, with slices nested within individual lungs. This is specified by 'slice(lung)'.

Under the 'model' line we now specify that 'slices' is a random effect within lungs. Because we sampled our slices from a theoretically infinite number of possible slices, it acts as a random variable here.

In the project description page on the biostats website, there are a number of embedded links in the instructions that provide examples and tips for working through this section. I recommend you look at each of these as you go. They are particularly useful for understanding the output of each analysis.