# ENV 710 – Applied Data Analysis
# Fall 2014
# Lab 8: More ANOVA models

In this lab, we explore ANOVA designs that are a bit more complicated than a one-way ANOVA. We are going to explore three different designs: factorial design, block design, and repeated measures design. There are many others, but this should get you comfortable with their analysis so that you can learn others by yourself as you need.

The learning goals of the lab are to:

- Recognize different study designs and the underlying reasoning behind them
- Understand how to adapt your analysis for each design
- Reinforce your ability to conduct ANOVA, post-hoc tests, and interpret the results
- Practice writing about the results of your analyses.

At the end of the lab, there are a two datasets to analyze. Please type your answers to each problem, including any requested graphs, in a Word document. *Submit your answers and your* ***R****-code to the class Sakai site under the folder Assignments before 5 pm on Wed., Oct. 29 (Sections 01, 02, and 07).*

## More functions in R

In this lab, we introduce a few new R commands.

`aov()` - Fits an ANOVA model by a call to `lm` for each stratum.
`with()` - Tells a function the data to use, possibly modifying the original data; avoids the need to use `attach()`.
`lm()` - Fits a linear model to the data; used for regression and ANOVA (when the independent variable is a factor).
`interaction.plot()` – Plots the mean of the response for the two-way combinations of factors, thereby illustrating possible interactions.
`bartlett.test()` - Provides a parametric K-sample test of the equality of variances.
`points()` - Draws a sequence of points at user-specified coordinates.
`*` - Symbol used to define a full first order model (all main effects and interactions): $y \sim \beta_0 + \beta_1 A + \beta_2 B + \beta_3 AB$
`:` - Symbol used in defining an interaction in a model.

## Factorial Design

A factorial design has two or more factors, each with two or more levels. The test subjects for a factorial design are assigned to treatment levels of every factor combination at random. This means that we can investigate statistical interactions, in which the response to one factor depends on the level of another factor.

Let's use an example of animal diets from Crawley (2005). In this dataset (download "Growth.csv"), the response variable is weight gain after 6 weeks. There are two factors: diet and supplement.

```
feed <- read.csv("Growth.csv", header = T)
```

How many levels are there of each factor? Use `levels()` to check it out. Before we get started, also check to make sure diet and supplement are defined as factors.

### Getting a feel for the data

The first thing we want to do is get a picture of the data. Note here that I use the function `with()` to tell **R** which data to use for the boxplot. This function can be useful when you do not want to attach data permanently to the **R** search path.

```
with(feed, boxplot(gain ~ diet + supplement, las = 1,
     ylab = c("Weight gain, pounds"),
     xlab = c("Supplements")))
```

We could also view this with a barplot, although I prefer boxplots.

```
cols <- c("darkblue", "darkred", "darkgreen")
with(feed, barplot(tapply(gain, list(diet, supplement),
     mean), ylim =c(0,35), beside = T,
     col = cols, las=1, ylab = c("Weight gain, pounds"),
     xlab = c("Supplements")))
labs <- c("Barley", "Oats", "Wheat")
legend(2.3, 35, labs, fill = cols)
```

### Running the model

There are a couple ways to run the factorial ANOVA in **R**, let's start with `aov()`.

```
mod0 <- aov(gain ~ diet*supplement, data = feed)
```

Here the `*` specifies that we want estimates for the main effect of each level of diet and supplement and the interaction between diet and supplement. We could also write out each of the main effects and the interaction effect. Note that I do this below using two different functions: `aov()` and `lm()`, using `anova()` to get the ANOVA table. There are often multiple ways to do things in **R**, but the below lines of code should give the same results.

```
mod0 <- aov(gain ~ diet + supplement + diet:supplement,
      data = feed)
summary(mod0)


mod0a <- lm(gain ~ diet + supplement + diet:supplement,
      data = feed)
anova(mod0a)
```

The results provide no support of a significant interaction between diet and supplement. We can use an interaction plot to get a better idea of what is going on. An interaction plot displays the levels of one factor on the x-axis and the mean response for each treatment on the y-axis. In addition, it shows a separate line connecting the means corresponding to each level of the second factor. When no interaction is present the lines should be roughly parallel.

These types of plots can be used to determine whether an interaction term should be included in our ANOVA model.

```
with(feed, interaction.plot(diet, supplement, gain,
     col=c(1,2,3,4)))
```

The consistent decline in weight gain from barley to oats to wheat demonstrates the significance of the main effect of diet.  Similarly, the separation of the lines for supplement also suggests a difference along that main effect. The lines are parallel to each other, indicating no interaction.

## Refining the model

In many cases, the goal of analysis is to look for the simplest (most parsimonious) model for our data. Therefore, we can refine our model taking out non-important, here interpreted as non-significant, variables. Let's simplify our ANOVA model by taking out the interaction term.

```
mod1 <- aov(gain ~ diet + supplement, data = feed)
```

Like the previous model, the ANOVA results tell us that *diet* and *supplement* are statistically significant main effects, but where do the differences in weight gain lie?

```
require(graphics)
TukeyHSD(mod1, "diet", ordered = TRUE)
plot(TukeyHSD(mod1, "diet"))
```

The Tukey test tells us that there are significant differences between the means of each of the diets. In other words, the mean weight gain is greatest on barley, then oats, and then wheat, with significant differences in mean weight gain between each pair of diets at the 0.05 level. Modify the above code to examine where the differences lie for different levels of *supplement*.

## Checking the fit of the model

When conducting any statistical analysis it is important to evaluate how well the model fits the data and that the data meet the assumptions of the model. There are numerous ways to do this and a variety of statistical tests to evaluate deviations from model assumptions. Generally statisticians examine various diagnostic plots after running their regression models.  Here is an introduction to diagnostics, which we will be talking about more when we get into regression.

```
par(mfrow=c(2,2))
plot(mod1)
```

Plotting of the model provides four diagnostic plots.

The first is plot of the residuals (distance of the data points from the expected value) versus the fitted data. Points should be randomly scattered around the centerline. Any pattern indicates either a violation of linearity or homoscedasticity.

The second plot is a q-q plot, which we have already used to evaluate the normality of a variable. Significant departures from the line suggest violations of normality. If the pattern were S-shaped or banana shaped, we would need a different model. You can also perform a Shapiro-Wilk test of normality with the `shapiro.test()` function.

The third plot is a plot of standardized residuals versus the fitted values. It repeats the first plot, but on a different scale. It shows the square root of the standardized residuals (where all the residuals are positive). If there was a problem, the points would be distributed inside a triangular shape, with the scatter of the residuals increasing as the fitted values increase.

The fourth plot is a residuals-leverage plot that shows the Cook's distance for each of the observed values of the factor levels. Cook's distance measures relative change in the coefficients as each replicate is deleted. So the point is to highlight those *y* (response) values that have the biggest effect on parameter estimates. The idea is to verify that no single data point is so influential that leaving it out changes the structure of the model.

The `bartlett.test()` function provides a parametric K-sample test of the equality of variances. The null hypothesis of the test is that the variances are all equal.

```
bartlett.test(gain~diet, data=feed)
bartlett.test(gain~supplement, data=feed)
```

## Block Design

The basic idea behind blocking is to group the experimental units into blocks of similar units and carry out the treatment assignment separately within each block. With every treatment included at least once in every block the design is called a complete block design. For example, say you want to compare the effectiveness of three fertilizers for increasing tomato growth. You would apply each of the fertilizers (treatment) to randomly chosen tomato plants in several different gardens (block). You are really interested in which fertilizer has the greatest affect on growth, but take into account variation across gardens because there may be unmeasured factors (soil fertility, exposure to the sun) that led plants in some gardens to have more or less growth than other gardens.

Blocking removes as much variability as possible from the random error so that the differences among the groups are more evident. The focus of the analysis is on the difference among groups (or treatments), not the blocks.

In the completely randomized design (e.g. one-way ANOVA), the total variation ($SS_T$) is subdivided into variation due to differences *among* the *a* treatment groups ($SS_A$) and variation *within* the *a* groups ($SS_W$). Within-group variation is considered to be random variation, and among-group variation is due to differences from group to group and random variation. To remove the effects of the blocking from the random variation component in the randomized block design, the within-group variation ($SS_W$) is subdivided into variation due to differences among the blocks ($SS_{BL}$) and random variation ($SS_E$).

In this example, we want to test whether four different antibiotics result in different levels of antibodies in the blood. Sixteen people are randomly assigned one of the four antibiotics, and samples of their blood are taken for analysis. To process the blood samples as quickly as possible, the samples are sent to four different laboratories – each lab receives one blood sample treated with one of the four antibiotics.

```
lab <- c(rep(1:4, each = 4))
antibiotic <- rep(c(1:4), 4)
results <- (c(9.3, 9.4, 9.6, 10, 9.4, 9.3, 9.8, 9.9, 9.2,
              9.4, 9.5, 9.7, 9.7, 9.6, 10, 10.2))

dlabs <- data.frame(cbind(lab = as.factor(lab),
          antibiotic = as.factor(antibiotic), results))
dlabs$antibiotic <- as.factor(dlabs$antibiotic)
```

Each laboratory has its own instruments and personnel that might result in variation in the results across laboratories. Our variable of interest is the level of antibodies in the blood samples; laboratories are blocks whose uncontrolled effects we want to separate from the main effect.

First, let's do the analysis without the block effect. What is your interpretation?

```
mod2 <- aov(results ~ factor(antibiotic), data = dlabs)
```

Now, let's include the block effect. How does this change your interpretation? We could do this using either the `lm()` function or the `aov()` function.

```
mod3 <- lm(results ~ factor(antibiotic) + factor(lab),
        data = dlabs)
anova(mod3)


mod3 <- aov(results ~ factor(antibiotic) + factor(lab),
        data = dlabs)
summary(mod3)
```

So, up to now, this should all seem pretty straightforward. `mod3` looks pretty much like a factorial ANOVA without the interaction, and we are treating the effect of *antibiotic* as a nuisance. If it were only so easy…

Recall that there are two types of ANOVA. Model I, or fixed effects ANOVA, applies when the treatments have been specifically chosen. For example, when you are interested in the effects of the particular antibiotics above. In other words, you want to know how antibiotics 1, 2, 3 and 4 specifically impact the results. Model II, or random effects ANOVA, applies to hypotheses that are more general. Instead of examining the effects of four specific antibiotics, your null hypothesis might be: “There is no effect of antibiotics, in general, on results.” Therefore, the antibiotics chosen are merely representatives of a wider range of antibiotics, even though your random selection might be antibiotics 1, 2, 3, and 4.

If we treat the blocking variable as a fixed effect, then the inference will only apply to those particular blocks (or samples). If we treat the blocking variable as a random effect, then inference can be made to the population of all possible blocks. The second option is what we are after; however, the rule of thumb is that you need at least six subjects (six samples) to estimate a random effect (i.e. variance) or the precision on the estimate will not be able to be estimated. Note that this also depends upon the assumption that the blocks are chosen randomly from a normal distribution of blocks.

The take-away message is that it is preferable to treat the blocking factor as a random effect, but may not always be possible. Let’s see what happens when we set up the model for the Model II ANOVA. Note that the term `Error` below classifies *lab* as a random effect, rather than a fixed effect.

```
mod4 <- aov(results ~ antibiotic + Error(factor(lab)),
        data = dlabs)
```

It looks like we got lucky, and the results do not seem to have changed even though we only had four samples (blocks). To do the Tukey *post-hoc* test, we will use `mod3`.

```
TukeyHSD(mod3, "factor(antibiotic)")
```

In full transparency, using `TukeyHSD()` on `mod4` results in errors. The `TukeyHSD()` function doesn’t work with random effects. Below we download the package `agricolae` to do the post-hoc test, but here it doesn’t seem to work. I am working on finding a work around for this problem – stay tuned.

Incidentally, the above ANOVA model could be used for situations where you have a fixed effect and a random effect, where the random effect is not necessarily a block, but rather some other nuisance variable.

## Repeated Measures Design

A repeated measure ANOVA is used when all members of a random sample are measured under a number of different conditions. As the sample is exposed to each condition in turn, the measurement of the dependent variable is repeated. Using a standard ANOVA would not be appropriate because it fails to model the correlation between the repeated measures: the data violate the ANOVA assumption of independence.

We look at an example of two advertising campaigns in ten different cities. For each city, the sales before, during and after the advertising campaign are measured. We want to know if the campaign has a significant effect on sales.

```
ad <- read.csv("Sales.csv", header=T)
ad <- with(ad, data.frame(sales, city=factor(city),
         campaign = factor(campaign), time =
         factor(time)))
```

Let's first proceed without including repeated measures. It looks like there is not a significant effect of campaign, time, or their interaction.

```
mod5 <- lm(sales ~ campaign * time, data = ad)
anova(mod5)
```

Now let's add in the repeated measures of city (i.e. the sales measurements were taken at the same city before, during, and after the ad campaign). Does this change your interpretation?

```
mod6 <- aov(sales ~ campaign*time + Error(city),
        data = ad)
```

Note here that the campaign results are listed under the city error. This is because city is nested within campaign. There is no significant main effect of campaign, but there is a main effect of time in the sales.

Again, `TukeyHSD()` function doesn't work with random effects, so we use the `HSD.test()` from the `agricolae` package to do the post-hoc test.

```
out <- with(ad, HSD.test(sales, time, DFerror = 2,
     MSerror = 358))
```

The `HSD.test()` function calculates the "honestly significant difference" (the level of difference between pairwise comparisons at which they are significantly different). It represents the differences among groups by labeling different groups of means with different letters ("a" or "b" in this case). The package can also make us a nice bar graph of the differences in group means.

```
bar.group(out$groups, ylim=c(0,850), density=4,
          border="darkblue", las=1)
with(ad, tapply(sales, list(time), mean))
```

## Your assignment

Analyze the data for the two examples below.  Make sure to use the appropriate ANOVA design for each study. For each problem, write a 1-page description of your analysis and the results. Each write-up should include the following information:

- Null and alternative hypotheses of your tests
- Justification for your choice of ANOVA model
- A description of how you checked the assumptions of your statistical test
- Results of your statistical test, interpreting your test in 2-3 sentences that include the appropriate reporting of the statistics
- An interpretation of any necessary *post-hoc* tests
- An interpretation of diagnostic figures

### Example 1

In the first example, we evaluate the effect of salt on plant biomass growth in 24 experimental vegetation plots. The assigned treatment is applied to the soil of a plot and at the end of the experiment the biomass of plants in each plot is measured. The experimental units are grouped into four blocks of six plots each, based on geographic proximity, and the treatments are assigned completely at random within each block. Thus, each treatment occurs exactly once in each block.

```
obs salt block biomass
 1   10     1    11.8
 2   15     1    21.3
 3   20     1     8.8
 4   25     1    10.4
 5   30     1     2.2
 6   35     1     8.4
 7   10     2    15.1
 8   15     2    22.3
 9   20     2     8.1
10   25     2     8.5
11   30     2     3.3
12   35     2     7.3
13   10     3    22.6
14   15     3    19.8
15   20     3     6.1
16   25     3     8.2
17   30     3     6.1
18   35     3     5.2
19   10     4     7.1
20   15     4     9.9
21   20     4     1.0
22   25     4     2.8
23   30     4     0.7
24   35     4     2.2
```

## Example 2

Pangolins are scaly anteaters that inhabit the tropical forests of Asia and Africa, and that are hunted for meat and for their scales, which are made of hair. A researcher wants to evaluate the affect of diet on the thickness of pangolin scales, with the idea that thicker scales would better protect pangolins from predators. She rears pangolins and provides them with identical diets, but different doses (0.5, 1, 2 milligrams) of supplements (Vitamin B and zinc). Download the “ScaleThickness.csv” file from Sakai and analyze the data to determine the potential effects of *doses* and *supplements* and whether there is an interaction between the two of them.