R Demonstration – Randomized Block and Factorial ANOVA

Objective: The purpose of this week's session is to demonstrate how to perform ANOVA on randomized block and nested designs in R. We will also explore some of the limitations and caveats for analyzing these types of designs.

Part I. Performing an ANOVA on a Randomized Block design

NOTE: This part of the exercise assumes that you have downloaded the dataset that reports the dry weight of flour beetles (*Tribolium castaneum*), grouped by genotype and blocked by the day, and saved it in your `PCB6466` folder as a tab-delimited text file named *`Tribolium.txt`*. You also need to download the *`ANOVA2.R`* script and save it in your `PCB6466` folder.

After starting R, change the directory to your `PCB6466` folder and open the *`ANOVA2.R`* script. The first two lines of the script read and attach the *Tribolium* dataset:

```
## read and attach the data for the randomized block example
block_data <- read.table("Tribolium.txt", header=T)
attach(block_data)
```

Next, we use the *`factor`* function to indicate that the variable named `blocks` is a categorical variable and store the result in a new variable named `block`:

```
## use the factor() function to tag blocks as a categorical variable
block <- factor(blocks)
```

As mentioned above, each of the blocks in this example is a different day of the experiment (NOTE: this differs from a repeated-measures design, because new groups of flour beetles were measured each day). We could also use the *`factor`* function to designate `genotype` as a categorical variable, but it is not necessary to do this because all the genotypes are represented as character strings ("`bb`", "`plusb`", and "`pluplus`") and R automatically considers such data as categorical variables.

Once again, we use the familiar *`lm`* function to create a linear model, only this time we relate the response variable (`dryweight`) to both the treatment factor (`genotype`) and the block number (`block`). Then we use the *`anova`* function to produce summary statistics, including an ANOVA table, for our model. Here is the R code:

```
## create and execute the randomized block ANOVA model
block_model <- lm(dryweight ~ genotype + block)
anova(block_model)
```

The call to the *anova* function produces the following output:

```
Analysis of Variance Table

Response: dryweight
          Df    Sum Sq   Mean Sq F value   Pr(>F)
genotype   2 0.0097172 0.0048586  6.9682 0.027259 *
block      3 0.0213910 0.0071303 10.2264 0.008967 **
Residuals  6 0.0041835 0.0006973
---
```

As discussed in class, two null hypotheses can be tested with the randomized block design. The first hypothesis is that there are no differences among treatments (genotype in this example). Based on the calculated F-ratio and the associated P-value of approximately 0.027, we would reject this null hypothesis. The second null hypothesis that could be tested with a randomized block design is that there are no differences among blocks. While we could also reject this null hypothesis based on the calculated F-ratio and P-value, this hypothesis is seldom of interest. We should only use a randomized block design when we have *a priori* evidence of a strong spatial or temporal gradient in our study environment. Thus, we should always expect to find differences among blocks if we have used the randomized block design properly.

Part II. Performing an ANOVA on a Nested design

NOTE: This part of the exercise assumes that you have downloaded the hypothetical dataset that reports the number of seeds, grouped by treatment and with sub-sampling within each replicate, and saved it in your PCB6466 folder as a tab-delimited text file named *pollination.txt*. You also need to download the *ANOVA2.R* script and save it in your PCB6466 folder.

The following lines of code read and attach the hypothetical pollination experiment data:

```
## read and attach the data for the nested example
nested_data <- read.table("pollination.txt", header=T)
attach(nested_data)
```

We once again use the *factor* function to indicate that the variable named replicate is a categorical variable:

```
## use the factor() function to tag replicate as a categorical variable
rep <- factor(replicate)
```

Now we will use the *lm* function to specify our ANOVA model, only this time we will use the "/" symbol in our linear model formula to indicate that our replicates are nested within our treatments:

```
## create and execute the nested ANOVA model
nested_model <- lm(seeds ~ treatment/rep)
anova(nested_model)
```

The call to the *anova* function produces the following output:

```
Analysis of Variance Table

Response: seeds
               Df  Sum Sq Mean Sq F value    Pr(>F)
treatment       2  7389.9  3694.9 23.1896 1.443e-08 ***
treatment:rep  12  5400.5   450.0  2.8245  0.003049 **
Residuals      75 11950.2   159.3
---
```

As noted in both the lecture and the Gotelli & Ellison text (see page 304), a common error with nested ANOVA models is to treat the subsamples as independent (even though they are not) and to divide the mean square (MS) among groups by the mean square residual. Unfortunately, even R is not immune to this error and **the F-ratio reported in the table above is not correct**! The correct analysis involves dividing the MS among groups by the MS among replicates within groups. Applied to the example above, this would produce an F-Ratio of (3694.9/450.0) ≈ 8.21 (not 23.2 as reported above). While this would not affect the statistical significance of the current example (i.e. both F-ratios yield P-values that are less than 0.05), it could definitely do so for other datasets.

The easiest way to get the correct F-ratio is to calculate it "by hand," using the MS values returned from the ANOVA. First, we need to store the results generated by the *anova* function in a variable, which we'll call `nested_anova`:

```
## to construct the proper F-ratio, we need the anova() return value
nested_anova <- anova(nested_model)
```

Now, we use the familiar bracket notation to extract the MS among groups from the first row of the third column and the MS among replicates within groups from the second row of the third column. Then we simply divide the former by the latter to produce the appropriate F-ratio:

```
## construct F-ratio: (MS_among_groups/MS_among_replicates)
MS_among_groups <- nested_anova[1,3]
MS_among_replicates <- nested_anova[2,3]
F_ratio <- MS_among_groups/MS_among_replicates
```

There is another approach we can take that avoids this problem entirely. Rather than using a nested ANOVA design, we can average all of the sub-samples and then perform a one-way ANOVA of our response variable against the treatment groups. To do this, we will use the R function ***aggregate*** to generate the means for all of the sub-samples within each replicate:

```
aggregate_list <- list("treatment"=treatment, "replicate"=rep)
replicate_means <- aggregate(seeds, by=aggregate_list, FUN=mean)
```

The first line of code above creates a list indicating that we wish to create a mean grouped by both treatment and replicate. The second line of code uses the ***aggregate*** function to calculate the mean of each of the groupings designated by our `aggregate_list` variable.

Now that we have averaged all of the sub-samples within each replicate, we can perform a simple one-way ANOVA that links the number of seeds (designated by `replicate_means$x`) to the treatment (designated by `replicate_means$treatment`):

```
## finally, do the analysis again as a one-way ANOVA
oneway_model <- lm(replicate_means$x ~ replicate_means$treatment)
anova(oneway_model)
```

This produces the following ANOVA table output:

```
Analysis of Variance Table

Response: replicate_means$x
                          Df  Sum Sq Mean Sq F value   Pr(>F)
replicate_means$treatment  2 1231.64  615.82  8.2103 0.005666 **
Residuals                 12  900.08   75.01
---
```

As shown above in green, our calculated F-ratio now matches the correct value of approximately 8.21!

To finish this session, we detach both of the datasets we used in this exercise:

```
## detach the data
detach(block_data)
detach(nested_data)
```