

R Demonstration – Two-Way Factorial ANOVA

Objective: The purpose of this week's session is to demonstrate how to perform a two-way factorial ANOVA in R. We will first perform the analysis assuming that both factors are fixed, and then we will redo the analysis with one or both factors as random effects.

Part I. Two-way factorial ANOVA with both factors fixed

NOTE: This part of the exercise assumes that you have downloaded the dataset that reports the number of eggs produced by limpets (grouped by season and density) and saved it in your PCB6466 folder as a tab-delimited text file named *factorial.txt*. You also need to download the *Factorial_ANOVA.R* script and save it in your PCB6466 folder.

After starting R, change the directory to your PCB6466 folder and open the *Factorial_ANOVA.R* script. The first 2 lines of the script read and attach the dataset:

```
## Read and attach the data
factorial_data <- read.table("factorial.txt", header=T)
attach(factorial_data)
```

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

```
## Use factor() to flag density as a categorical variable
cat_density <- factor(density)
```

Once again, we use the familiar *lm* function to create a linear model, only this time we relate the response variable (*number*) to 2 fixed factors (*season* and *cat_density*) and their interaction term (*season:cat_density*). Then we use the *anova* function to produce summary statistics, including an ANOVA table, for our model:

```
### PART I. Two-way ANOVA: Both Factors Are Fixed ###
model <- lm(number ~ season + cat_density + season:cat_density)
anova(model)
```

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

Analysis of Variance Table

```
Response: number
          Df Sum Sq Mean Sq F value    Pr(>F)
season      1  3.2502   3.2502  17.8419 0.0006453 ***
cat_density  3  5.2841   1.7614   9.6691 0.0007041 ***
season:cat_density 3  0.1647   0.0549   0.3014 0.8239545
Residuals  16  2.9146   0.1822
---
```

Notice that both of the fixed factors (*season* and *cat_density*) are highly significant ($P < 0.001$), but that their interaction term (*season:cat_density*) is not significant ($P > 0.05$).

We can double-check these results by “manually” calculating the F-ratios and their corresponding P-values. First, we will save the output of the *anova* function as a variable named *results*. Then we will use the brackets [] notation to extract the various mean squares (MS) for the ANOVA from the *results* object:

```
## Get mean squares (MS) from the ANOVA
results <- anova(model)
MS_A <- results[1,3]
MS_B <- results[2,3]
MS_AB <- results[3,3]
MS_resid <- results[4,3]
```

In the code above, *MS_A* represents the mean square for Factor A (*season*), *MS_B* represents the mean square for Factor B (*cat_density*), *MS_AB* represents the mean square for the interaction term, and *MS_resid* is the residual mean square. We can use the same approach to extract the degrees of freedom (df) for each of the ANOVA terms:

```
## Get degrees of freedom (df) from the ANOVA
df_A <- results[1,1]
df_B <- results[2,1]
df_AB <- results[3,1]
df_resid <- results[4,1]
```

Now we have all the information we need to calculate the appropriate F-ratios and P-values. Remember, there are three null hypotheses we can test using a two-way factorial ANOVA: 1) that there are no differences among the levels of Factor A, 2) that there are no differences among the levels of Factor B, and 3) that there are no interactions between Factor A and Factor B.

For a two-way ANOVA in which both of the factors are fixed, we simply divide the appropriate MS value by the residual mean square to calculate the F-ratio for each of these null hypotheses. For example, to calculate the F-ratio to test the first null hypothesis (no differences among the levels of Factor A), we divide *MS_A* by *MS_resid*. We then call the *pf* function on this F-ratio, along with the appropriate numerator and denominator degrees of freedom, to generate the corresponding P-value:

```
## Calculate F-ratio & P-value for fixed Factor A
F_ratio_A <- MS_A/MS_resid
p_A <- pf(F_ratio_A, df_A, df_resid, lower.tail=F)
```

In this example, *F_ratio_A* equals 17.8419 and *p_A* equals 0.0006453. These values exactly match those listed in the ANOVA table output from R (see the previous page).

Next, we will compute the F-ratios and P-values for the other 2 hypothesis tests:

```
## Calculate F-ratio & P-value for fixed Factor B
F_ratio_B <- MS_B/MS_resid
p_B <- pf(F_ratio_B, df_B, df_resid, lower.tail=F)

## Calculate F-ratio & P-value for Interaction (AxB)
F_ratio_AB <- MS_AB/MS_resid
p_AB <- pf(F_ratio_AB, df_AB, df_resid, lower.tail=F)
```

Again, we can verify that the manually calculated values for Factor B ($F_{ratio_B} = 9.6691$ and $p_B = 0.0007041$) and the interaction term ($F_{ratio_AB} = 0.3014$ and $p_{AB} = 0.8239545$) are identical to those contained in the R output of the ANOVA table.

Part II. Two-way factorial ANOVA with random effects

In the previous section, we performed a two-way factorial ANOVA in which we considered both of the factors to be fixed effects. When we use the *lm* and *anova* functions for a two-way model in R, the software automatically assumes that both of our factors are fixed. Furthermore, we verified this by comparing the “manually” calculated F-ratios and P-values for a fixed-effects model to those generated in the R output.

There are 2 other possible models for a two-way factorial ANOVA: the mixed effects model, and the random effects model. In the mixed effects model, Factor A is considered to be fixed while Factor B is considered to be random. In our sample dataset, for example, we could consider season (Factor A) to be fixed and the density of adult limpets (Factor B) to be a random effect. To test our null hypothesis regarding Factor A in this model, we would calculate a new F-ratio by dividing MS_A by MS_{AB} (instead of by MS_{resid} as we did in the fixed effects model above):

```
### PART II. Two-way ANOVA: Factor A is fixed, Factor B is random ###

## Calculate F-ratio & P-value for Factor A as MS_A/MS_AB
F_ratio_A <- MS_A/MS_AB
p_A <- pf(F_ratio_A, df_A, df_AB, lower.tail=F)
```

While our new F-ratio (59.20594) and its corresponding P-value (0.00456) would not change our decision to reject the null hypothesis of no difference among levels of Factor A, it is important to note that this will not always be the case! It is also important to note that the F-ratios (and their P-values) for Factor B and the Interaction are calculated in the same manner in the mixed model as in the fixed effects model in Part I.

Finally, we will consider the case of a two-way ANOVA in which both factors are random. In this model, the F-ratio for Factor A is calculated as it was for the mixed model above (MS_A/MS_{AB}). And, once again, the F-ratio for the Interaction AxB is calculated as MS_{AB} divided by MS_{resid} . The only difference between the random

effects model and the mixed model is that the F-ratio for Factor B is now calculated as MS_B divided by MS_AB:

```
### PART III. Two-way ANOVA: Factor A and Factor B are random ###  
  
## NOTE: here, the F-ratio for Factor A is the same as in Part II.  
  
## Calculate F-ratio & P-value for Factor B as MS_B/MS_AB  
F_ratio_B <- MS_B/MS_AB  
p_B <- pf(F_ratio_B, df_B, df_AB, lower.tail=F)
```

The new F-ratio calculated for Factor B (32.08569) and its corresponding P-value (0.00884) lead us to reject the null hypothesis of no differences among the levels of Factor B, the same conclusion we reached in the other models. It should be emphasized again, however, that this will not always be the case. As discussed in class and in the Gotelli & Ellison text, we must take great care to apply the proper model to the analysis of two-way factorial designs. Relying on the F-ratios and P-values calculated for the default fixed effects model in R can cause us to report erroneous results in our two-way ANOVA tables.

As usual, we end our R session by detaching the data used in this example:

```
## Detach the data  
detach(factorial_data)
```