# ANOVA Exploration

**Motivation:**

I'm writing this document in an attempt to provide a little more information about the ANOVA test we use in this class. While it's easy to pick out the p-value and test statistic from our *R* functions, I think you should have a better understanding than just that.

I wouldn't turn to this document as the end-all/be-all of ANOVA expertise... because I am far from an ANOVA expert. Instead I wanted to connect a couple dots from topics and research questions we have covered this semester.

**Calculations:**

Let's take the data set we used in Lesson 22 that concerned memory recall of details in a story. First we'll load that dataset and use the **aov()** function to get our ANOVA table.

```
library(tidyverse)

recall = read_table2("http://www.isi-stats.com/isi/data/chap9/Recall.txt")

head(recall)
```

```
## # A tibble: 6 x 2
##   Condition Recall
##   <chr>      <dbl>
## 1 After          7
## 2 After          5
## 3 After          5
## 4 After          5
## 5 After          2
## 6 After          8
```

```
recall_anova = aov(recall$Recall ~ recall$Condition)

summary(recall_anova)
```

```
##                  Df Sum Sq Mean Sq F value   Pr(>F)
## recall$Condition  2  80.04   40.02   12.67 3.07e-05 ***
## Residuals        54 170.53    3.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now that we have this table laid out, let's take a step back and consider a broader view of the ANOVA test. This test analyzes the variance in the response variable (*Recall*) that is associated with the treatment or explanatory variable (*Condition*) and associated with the error (residuals). Said another way: how much of the variance in the response is due to changes in the explanatory variable and how much is due to random error?

Let's take a look at how we measure both of these variance quantities but first let's consider the general equation for variance:

$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$

The numerator here is the sum of squares of deviations from the mean and the denominator is the degrees of freedom. We can apply this definition to the two variances we need to calculate.

**Variance due to treatment:**
The term used for numerator for this variance value is the *treatment sum of squares* ($SSTr$) and the equation is as follows:

$SSTr = \sum_{i=1}^{k} n_i \times (\bar{y}_i - \bar{y})^2$ where $k$ is the number of treatments.

Let's calculate this value for our data set.

```
sst_r = recall %>%
  #Group by condition so we can calculate the terms inside
  # the summation for each group
  group_by(Condition) %>%
  #Calculate the term inside the summation
  # (term for each treatment group)
  summarise(terms = n() * (mean(Recall) - mean(recall$Recall))^2) %>%
  #Sum all of these terms
  summarise(sst_r = sum(terms))

sst_r[[1]]
```

```
## [1] 80.03509
```

*Note*: You might see *treatment sum of squares* using the abbreviation $SST$ in some sources. To me this can get easily confused with *total sum of squares* so I've added the extra $r$.

*Note 2*: Notice the difference between **mean(Recall)** and **mean(recall$Recall)** in the code above. Because this **summarise()** command is preceded by **group_by()**, **mean(Recall)** gives us the mean of the *Recall* values for the group it's "working on" and **mean(recall$Recall)** gives us the mean of the entire data set.

This term is a weighted (by sample size) measure of how different the average in a group (or condition) is from the overall average of our response variable. Hopefully you can see how calculating this would give you some idea of much of the variation in our response is due to group.

We scale this measurement by dividing by one less than the number of groups ($k - 1$) to calculate the *treatment mean squares* ($MSTr$).

```
k = recall %>%
  summarise(k = n_distinct(Condition))

mst_r = sst_r[[1]] / (k[[1]] - 1)

mst_r
```

```
## [1] 40.01754
```

**Variance due to error:**
The term used for numerator for this variance value is the *error sum of squares* ($SSE$) and the equation is as follows:

$SSE = \sum_{i=1}^{k} (n_i - 1)s_i^2$ where $k$ is the number of treatments.

Let's calculate this value for our data set.

```
sse = recall %>%
  #Group by condition so we can calculate the terms inside
  # the summation for each group
```

```
  group_by(Condition) %>%
  #Calculate the term inside the summation
  # (term for each treatment group)
  summarise(terms = (n() - 1) * var(Recall)) %>%
  #Sum all of these terms
  summarise(sse = sum(terms))

sse[[1]]
```

```
## [1] 170.5263
```

We scale this measurement by dividing by the sample size less the number of groups $(n - k)$ to calculate the *error mean squares* (*MSE*).

```
n = nrow(recall)

mse = sse[[1]] / (n - k[[1]])

mse
```

```
## [1] 3.157895
```

**F-statistic:**

As I mentioned in the *Lesson 22 Companion*, the F-statistic measures the ratio of "between group variability" and "within group variability." Here our measure "between group variability" is our *MSTr* and the measure of "within group variability" is our *MSE* (more to follow).

```
f_stat = mst_r / mse

f_stat
```

```
## [1] 12.67222
```

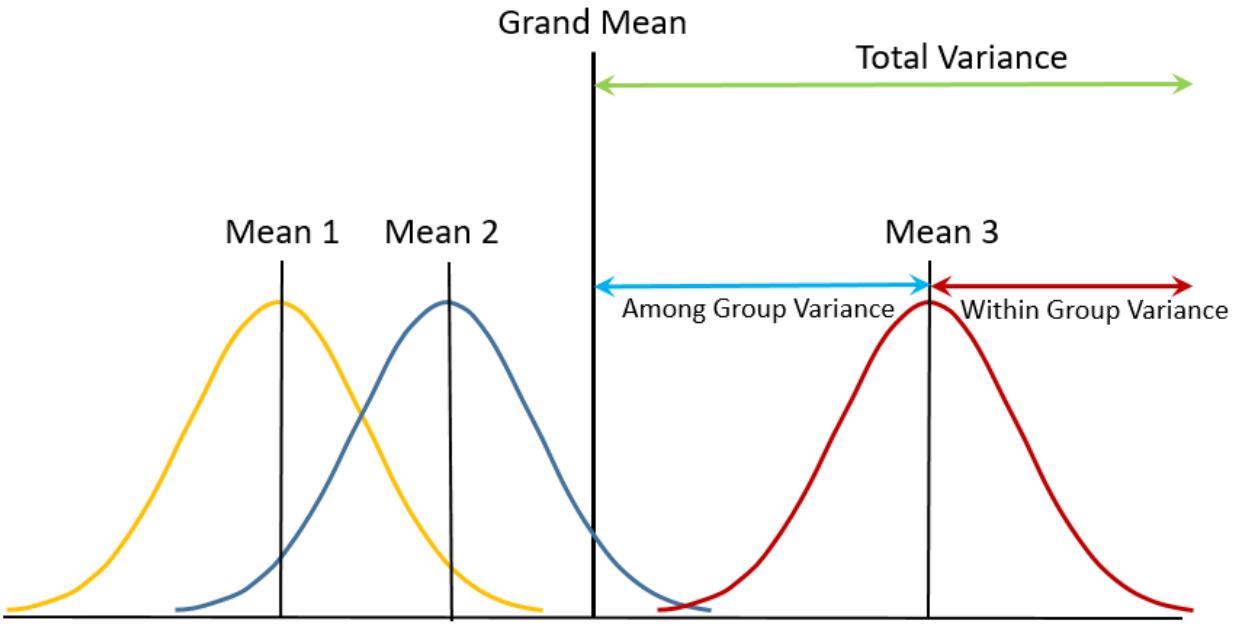
I know this is just a lot of work to verify the value that is already in the chart but hopefully you get an idea of where the other values comes from. At this point let's talk a bit more about why *MST* and *MSE* are good measures to use for our ratio.

**Measures of variability:**

I think the first hurdle to discuss is perhaps we use the *mean* (*MSTr* and *MSE*) values as opposed to the *sum* (*SSTr* and *SSE*) values. Hopefully this isn't a big hurdle when you consider that we are trying to utilize a single statistic that gives us a snapshot of the variance between all groups. It's a similar sentiment for why we so often use mean to describe a quantitative measure in a sample.

Taking a step back, I've already discussed how taking the weighted difference of the average in that group and the overall average provides us with an idea of the total variance that is due to the groups (or explanatory variable). As far as *SSE* and *MSE* providing a measure of within-group variance, if that doesn't make sense, try removing the weighting $(n - 1)$ from the equation. What you're left with is the variance of each group $(s^2)$ as a measure of within-group variance.

Sometimes I know that a picture is worth all of my words so here is one I got from the interwebs to help you out.




https://howecoresearch.blogspot.com/2019/01/using-analysis-of-variance-anova-in.html

They must be British or something because they call it "Among Group Variance" but you get the general idea.

**It's all about the ratio!**

Let's consider why this ratio (and thus the F-statistic) is useful to determine if there is evidence of a difference between groups.

What do you expect the value of $SSTr$ to be if there wasn't a significant difference between the groups? I would expect that the difference between the group averages and the overall average would be pretty small. This would reduce $SSTr$ and $MSTr$ and our numerator in the F-statistic calculation.

If we didn't change the demominator ($MSE$), we would end up with a smaller number. Given the shape of the F-distribution, a smaller standardized statistic value means a higher p-value. Higher p-value means less evidence against the hypothesis that there is no difference between the group means (null hypothesis).

```
library(patchwork)

#Red area
bigger_value = 2.5

#Green area
smaller_value = 0.5

p1 = ggplot(data.frame(x =c(0, 10)), aes(x = x)) +
  stat_function(fun = df,
                args = list(df1 = k[[1]] - 1, df2 = n - k[[1]])) +
  stat_function(fun = df,
                args = list(df1 = k[[1]] - 1, df2 = n - k[[1]]),
                xlim = c(bigger_value, 10),
                geom = "area", fill = "red") +
  labs(x = "", y = "")
```

```r
p2 = ggplot(data.frame(x =c(0, 10)), aes(x = x)) +
  stat_function(fun = df,
                args = list(df1 = k[[1]] - 1, df2 = n - k[[1]])) +
  stat_function(fun = df,
                args = list(df1 = k[[1]] - 1, df2 = n - k[[1]]),
                xlim = c(smaller_value, 10),
                geom = "area", fill = "green") +
  labs(x = "", y = "")

p1 / p2
```