# One-Way Analysis of Variance using R[*]

Jeric C. Briones

*Department of Mathematics*

*Ateneo de Manila University*

# 1 Preliminaries

## 1.1 Loading the Dataset

We begin by importing the dataset from the file `calcium.csv`. This dataset will be used for the analysis of variance (ANOVA). Codes used are in `2ANOVA.R`.

The `read.csv()` function imports the specified `.csv` file into a data frame. By default, it is assumed that the file has column headers. If there are no column headers, the parameter `header=F` should be added to `read.csv()`. Moreover, it is also assumed that the imported dataset is a two-column data matrix, where one column contains the values while the other contains the labels. In the example, the file `calcium.csv` is imported and stored in the variable `calcium`. This variable has two columns: `calcium$value`, which has the observation values, and `calcium$method`, which has the "treatment" labels.

```
## loading the dataset
calcium = read.csv("calcium.csv")
```

It should also be noted that the file is assumed to be in the same directory as the working director. Otherwise, either the working directory should be adjusted, or the full file path name must be used as input for the `read.csv()` function.

## 1.2 Exploratory Analysis and Preprocessing

After importing the dataset, we begin by looking at descriptive statistics of the dataset. This can be done using the function `boxplot()`, and the function `desribeBy()` from the `psych` library.

```
## exploratory analysis
library(psych)
boxplot(value~method, data=calcium) #data grouped by 'method'
describeBy(calcium$value, calcium$method, mat=T) # summary statistics
```

For the function `boxplot()`, the parameter `value~method` is used to indicate that the values under the column `value` are grouped based on their values under the column `method`. On the other hand, the parameter `data=calcium` is used to specify that the values are stored in the variable `calcium`.

[*] Supplementary notes for MATH 62.2 Time Series and Forecasting. Last updated Second Semester, SY 2024-2025.

Similarly, for the function describeBy(), the first parameter calcium$value indicates the observation value, the second parameter calcium$method indicates how the values are grouped, while the third parameter mat=T specifies that the results be displayed as a matrix.

After looking at descriptive statistics, we then check if column containing the “treatment” labels are of factor type. This is important since some ANOVA-related functions will not work if the “treatment” is not a factor type. The check can be done using class(). On the other hand, the conversion to factor type can be done using the function factor().

```
## preprocessing
class(calcium$method)
calcium$method = factor(calcium$method) # convert 'method' to factor
```

# 2 Analysis of Variance

Analysis of variance is primarily done using the aov() function. ANOVA results are then retrieved using the summary() function. To perform one-way ANOVA, the one-way balanced ANOVA model must first be specified. For example, we are given the observation data matrix $\boldsymbol{y}$, where $\boldsymbol{y}$ is $nk \times 1$. If the “treatment” labels are stored in $\boldsymbol{x}$, we will specify the ANOVA model as y ∼ x.

## 2.1 One-Way ANOVA

Recall that the column value has the observations while the column method has the “treatment” labels. As such, the model will be specified as value∼method. This will then be the first parameter in the aov() function.

```
## one-way ANOVA
calcium.aov = aov(value~method, data=calcium)
summary(calcium.aov)
```

Here, the parameter data=calcium is used to specify that the values are stored in the variable calcium. If the variables used in the model specification already exist in the current workspace, the parameter data can be omitted. On the other hand, the function summary() is used to display the resulting ANOVA table.

Note that the first row of the displayed table corresponds to the *treatment* row the ANOVA table, while the second row is for the *error* row. On the other hand, the columns are for the degrees of freedom *df*, sum of squares **SS**, mean square **MS**, $F$-statistic, and the corresponding $p$-value, respectively. For example, the value under the Mean Sq column and method row corresponds to $\dfrac{\text{SSH}}{k-1}$, while the value under the Mean Sq column and Residuals row corresponds to $\dfrac{\text{SSE}}{k(n-1)}$.

Inspecting the table, it can be seen that $F = 20.06$, with the corresponding $p$-value of 0.000149. Since the $p$-value is small, the null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3$ (or equivalently, $H_0 : \alpha_1 = \alpha_2 = \alpha_3$) is rejected.

## 2.2 Testing Contrasts

Suppose we are now interested with testing linear contrasts. This can be done using the `summary.lm()` function. Note that by default, the first "treatment" label is considered as the *base* group when producing contrasts. That is, the null hypotheses being tested are $H_{012} : \alpha_1 - \alpha_2 = 0$ and $H_{013} : \alpha_1 - \alpha_3 = 0$. The corresponding results can be seen in the `method2` and `method3` rows, respectively.

```r
## testing contrasts (method 1 as base)
summary.lm(calcium.aov)
```

Inspecting the ANOVA table, it can be seen that the $p$-value for $H_{012} : \alpha_1 - \alpha_2 = 0$ (0.000666) is small, which would lead to $H_{012}$ being rejected. That is, the values obtained using the first two methods are significantly different from each other. On the other hand, the null hypothesis $H_{013} : \alpha_1 - \alpha_3 = 0$ is not rejected since the $p$-value (0.149220) is not small. Thus, the values obtained using the first and third methods are not significantly different.

Suppose we are interested with changing the *base* group to Method 2. This can be done by rearranging the factor levels using `relevel()`. Here, the parameter `ref=2` is used to specify that factor level `2` will be the first factor level. ANOVA is then recarried out for the *modified* data `calcium`.

```r
## testing contrasts (method 2 as base)
calcium$method = relevel(calcium$method, ref=2) #rearrange factors
calcium.aov2 = aov(value~method, data=calcium)
summary(calcium.aov2)
summary.lm(calcium.aov2) # testing contrasts
```

Inspecting the ANOVA table produced by `summary()`, it can be seen that rearranging the factor levels does not affect the one-way ANOVA results. However, looking at the ANOVA table produced by `summary.lm()`, it can be observed that the *base* group has indeed changed.

Here, the null hypotheses being tested are $H_{021} : \alpha_2 - \alpha_1 = 0$ and $H_{023} : \alpha_2 - \alpha_3 = 0$. The corresponding results can be seen in the `method1` and `method3` rows, respectively. Inspecting the new ANOVA table, it can be seen that the $p$-value for $H_{021} : \alpha_2 - \alpha_1 = 0$ (0.000666) is the same as the $p$-value for $H_{012}$. What differed is the sign of their $F$-statistic ($-4.550$ and $4.550$, respectively). This is expected, since the two contrasts only differ by sign. On the other hand, $H_{023} : \alpha_2 - \alpha_3 = 0$ is also rejected since the $p$-value ($5.41 \times 10^{-5}$) is small. That is, the values obtained using the second and third methods are significantly different.

Aside from these basic contrasts, user-defined contrasts can also be used. For more information about this, you may refer to this link: `https://www.uvm.edu/~statdhtx/StatPages/R/AnovaOneway.html`.

## 2.3 Multiple-Comparison Procedures

While testing contrasts can be used to identify which pairs are significantly different from each other, there is a concern that too many false positives might be detected when multiple tests are carried out to identify all groups that are different from the rest. To remedy this, multiple-comparison procedures[1] can be used.

For example, the Bonferroni approach performs pairwise $t$-tests but with the *corrected* significance level $\alpha^* = \dfrac{\alpha}{c}$. Here, $c = \dbinom{k}{2}$ represents the number of pairwise tests expected to be performed. Alternatively, the $p$-value can be adjusted instead so as not to change the significance level $\alpha$. This pairwise test can be carried out using the `pairwise.t.test()` function, with the parameter `p.adj=‘b’` used to indicate the use of Bonferroni correction. Note that the input here are the values (stored in `calcium$value`) and their groups (stored in `calcium$method`).

```
## Comparison Method: Bonferroni approach
pairwise.t.test(calcium$value, calcium$method, p.adj='b')
```

Inspecting the results, the adjusted $p$-values for the $H_{012} : \alpha_1 = \alpha_2$, $H_{013} : \alpha_1 = \alpha_3$, and $H_{023} : \alpha_2 = \alpha_3$ tests are 0.0013, 0.1492, and 0.0002, respectively. Since only $H_{013}$ had $p$-value more than $\alpha = 0.05$, then only the first and third methods are statistically similar. That is, the second method is different from the rest.

Another method is Tukey's honestly significant difference (HSD) test. Unlike the Bonferroni approach, Tukey's method only works for the balanced case. Moreover, the testing is done for the mean difference, with the threshold computed using the studentized range distribution $q$. This method can be implemented using the `TukeyHSD()`.

```
## Comparison Method: Tukey Method
TukeyHSD(calcium.aov)
```

Here, the confidence intervals and (adjusted) $p$-values are provided as outputs. Inspecting the results, only the pair `method1-method3` had their confidence interval include 0. Furthermore, their (adjusted) $p$-value is also more than $\alpha = 0.05$. Thus, consistent with earlier results, only the first and third methods are statistically similar.

[1] While the implementation of some of these methods would be discussed, details of their mathematical formulation and justification would not be covered.