

# **Case Studies using the R Programming Language**

Katherine Bennett, Miguel de los Reyes, Raymond Gao,  
Sophia Hurr, Hannah Gahagan, Nikhil Miland, Mridu Nanda,  
Christa Parrish, Ishaan Rao, Grayson York

**RESEARCH IN COMPUTATIONAL SCIENCE**

**NORTH CAROLINA SCHOOL OF SCIENCE AND MATHEMATICS, DURHAM, NC**

This case studies manual was prepared under the direction of Mr. Robert Gotwals, faculty mentor,  
The North Carolina School of Science and Mathematics, Durham NC.

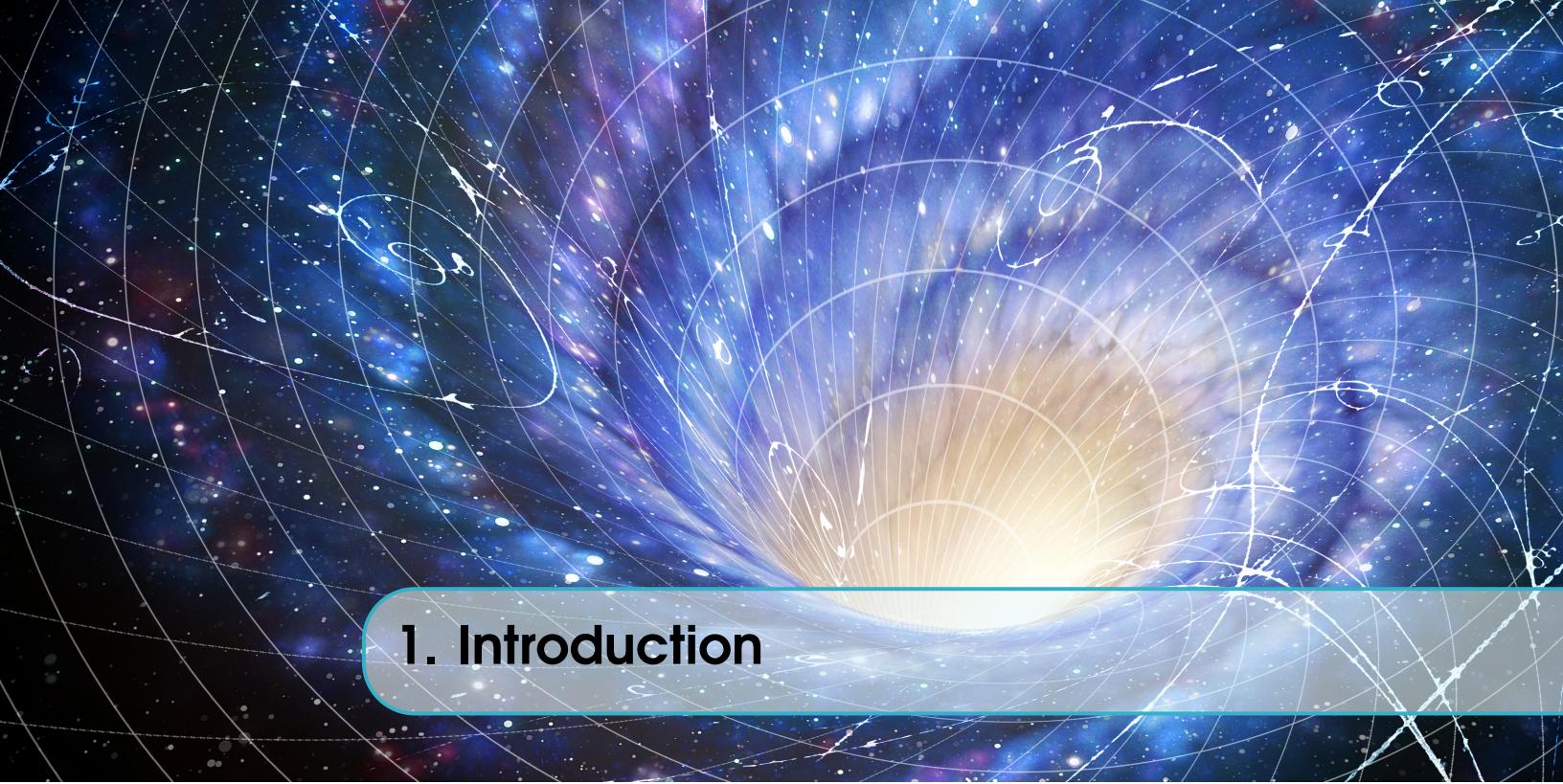
*First release, November 2016*



# Contents

<b>1</b>	<b>Introduction</b>	5
<b>2</b>	<b>ANOVA</b>	7
2.1	<b>Introductory Reading</b>	7
2.2	<b>Objectives</b>	8
2.2.1	ANOVA	8
2.2.2	Visualization	8
2.3	<b>Building the Model</b>	8
2.3.1	Programming Hints	10
2.4	<b>Deliverable</b>	10
2.5	<b>Teaching Code</b>	10
2.6	<b>Example Student Code</b>	11
2.7	<b>Further Readings</b>	12





## 1. Introduction

Gotwals will write an Intro.





## 2. ANOVA

### 2.1 Introductory Reading

The most common type of linear models are analysis of variance (ANOVA) and linear regression models. R is built to handle linear models, and as such it is easy to work with ANOVA. ANOVA is a statistical method used to compare two or more means. These values can be used to determine whether a significant correlation exists between variables. We will be using one-way ANOVA, which compares the means between the groups of interest, and determines whether those means are significantly different from each other. If one-way ANOVA returns a significant result, there are at least two group means that are significantly different from each other. It is important to note that ANOVA is an omnibus test statistic, meaning that although ANOVA tells the user if there is a significant difference in means, it is unable to show which means in particular differ. ANOVA can only be used with certain types of data configurations. To perform ANOVA, data must have a continuous response variable and at least one categorical factor with two or more levels, in lay terms, ANOVA can only be used with numeric values that can be ordered sequentially, with a certain number of possible responses. (i.e. data comparing object's weights, and each new weight is a level) ANOVA is easier to use if data is from approximately normally distributed populations with equal variances between factor levels, however, as R is built to work with statistics, ANOVA procedures will generally work without incident unless one or more of the distributions or variances are highly skewed. A basic understanding of statistics how to use R is recommended before performing ANOVA. The ANOVA user will create and order factors, make box plots, and combine and stack data. [cran-r] The following functions will be useful in using ANOVA.

1. `aov`
2. `as.factor`
3. `ls.str`
4. `data.frame`
5. `stack`
6. `TukeyHSD`
7. `levels`

## 2.2 Objectives

In this assignment, we will use ANOVA to analyze data from ELISA HIV Optical Density Readings. ANOVA identifies the causes of variation, and sorts out the corresponding components of variation with associated degrees of freedom. In this case, we are interested in seeing how HIV Optical Density Readings are related to their lot. The type of ANOVA we are using is one-way between groups, as we are comparing one grouping (the optical density readings) to define the groups (lots).

By the end of this lesson, the reader should be able to:

1. Understand the logic behind one-way analysis of variance.
2. Perform one-way analysis of variance in R for any data.
3. Appropriately interpret results of analysis of variance tests.

### 2.2.1 ANOVA

We will look at variances in data using R's ANOVA functions.

### 2.2.2 Visualization

We will also use our fitted models and our ANOVA data to create box plots and line plots. Much of the information gleaned from ANOVA is presented via numeric such as the sum of square, degrees of freedom, and the mean. ANOVA presents the null hypothesis that there is no difference in means of the treatments, and once this hypothesis is proven incorrect, the question arises of how the treatments differ. The post-hoc test also allows the user to find the differences in means, and specifically categorizes the lower and upper means of the data.

Figure 2.1 shows a box plot of the data before running ANOVA or TukeyHSD. Constructing a box plot before analyzing the data may prove helpful in deciding what kind of analysis would be preferable.

## 2.3 Building the Model

Looking at the data set of interest for ANOVA, note that that data may be in numeric form. Use the function `is.numeric` to review the data. If this function prints TRUE, use the function `as.factor` to change data from numeric to factor. In order to make it easier to manipulate data later on, it is recommended that the factored data is renamed.

---

```
1 lot = as.factor(elisa$Lot)
2 summary(lot)
3 levels(as.factor(elisaOptical))
4 optical = (as.factor(elisa$Optical))
5 levels(as.factor(elisa$Run))
6 run = (as.factor(elisa$Run))
```

---

Now your factor data type is in non-numerical variables. Each different variable of the factor is called a level. For example, our factor type data for the lots of various ELISA HIV Optical Density Readings has five levels, 1, 2, 3, 4 and 5. Moving on, we can make a box plot of our data, so that we

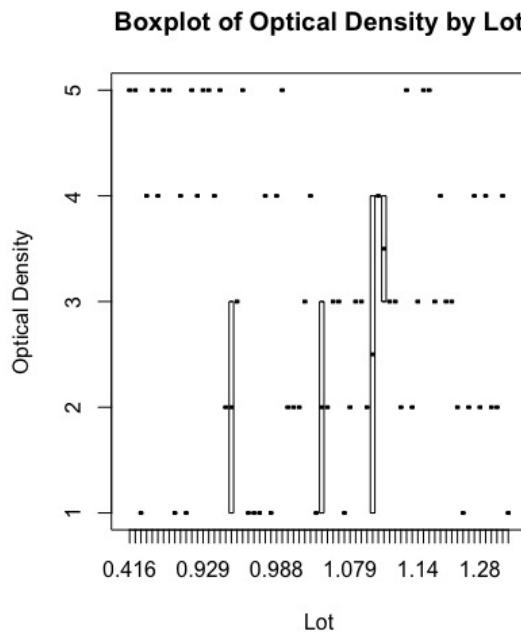


Figure 2.1: Boxplot

can visually compare data. Making a summary of this data allows us to look at the residuals and standard error of the plot. As mentioned before, this type of comparison will work better with data that has a relatively normal distribution. Factors can also be created within factors. We used our factor Optical to categorize and label by the density. For each level within the main factor, create labels. These labels will designate unordered factors. To make this data easier to read, unordered factors should be ordered. Choose ways to order the data that is relative to the averages in the data. For example, some levels were Low density(below 1) and some were High density,(above 1.5) so we ordered the data starting at 0, then continuing with 1, 1.5, 2. Choose labels that reflect the data, noting that the first label will relate to the lower numbers. Next, classify this newly ordered factor, and now label the data in lay terms.

---

```

1 elisa$Optical.type<-ordered(cut(elisa$Optical, c(0,1,1.5,2), labels = c("Low","M
2 class(elisa$Optical.type)
3 elisa$Optical.type

```

---

Note, that if you wish to remove one group of subjects (i.e. Lot 1) R will keep this removed group as a level, possibly skewing your data. Use the function drop.levels to remove a selected group as a level. Rename this reconfigured data as to not overwrite your previous data. In order to use ANOVA, the data must be in a specific format. To properly configure data, combine the data from the two factors intended to be compared with the function data.frame. Next, stack these combined groups. Finally use this stacked data and the function aov to perform ANOVA. Make a summary of these results.

---

```

1 finalelisa <- droplevels(newelisa)

```

---

```

2 summary(finalelisa$Lot)
3 Combined_Groups <- data.frame(cbind(lot, optical))
4 Combined_Groups
5 Stacked_Groups <- stack(Combined_Groups)
6 Stacked_Groups
7 Anova_Results <- aov(values ~ ind, data = Stacked_Groups)

```

---

This method can me used for a simple one-way ANOVA. However some data can compare multiple sets of data against each other. To do this, we can use a post-hoc test. Post-hoc tests compare outcome measurements between multiple groups. With post-hoc analysis the reader can examine differences between pairs of groups after global analysis. Use the function TukeyHSD to perform a pairwise post-hoc analysis on the ANOVA results.

---

```

1 summary(Anova_Results)
2 TukeyHSD(Anova_Results)
3 summary(TukeyHSD(Anova_Results))

```

---

### 2.3.1 Programming Hints

It is recommended to look at data before analysis in order to better understand the levels and attributes of data. In addition, we recommend the use of `summary()` and the console to view the attributes of objects.

## 2.4 Deliverable

Using ANOVA, students should be able to take data with sequentially ordered number data, make a box plot, factor the data as unordered and ordered factors, configure the data such that the function `aov` can be performed, and perform TukeyHSD. Using the box plot, students should be able to identify residuals, coefficients of the linear regression, and residual standard error. Using `aov`, students should be able to identify degrees of freedom, the sum of the squares, the mean of the squares, and the F ratio (the ratio of two mean square values) for their data.

## 2.5 Teaching Code

Begin ANOVA by formatting the data in a `.csv` file, then uploading it to RStudio. This script is set up to analyze publicly available data about HIV Optical density readings, but can be adapted for any properly set up data. As the reader works his or her way through the script, be sure to liberally comment the purpose of each command.

---

```

1 # Your name here
2 # Date
3 # ANOVA - ELISA HIV Optical density readings data
4
5 # Clean up and read in data
6 rm(list=ls())
7 setwd("C:/Your/Directory")
8 elisa = read.csv("ELISAHIV.csv")
9

```

```
10 # Create levels for lot, optical, and run
11 # Make sure to view/check your data
12 levels(as.factor(elisa$Lot))
13 lot=as.factor(elisa$Lot)
14 # etc.
15
16
17 # Create a boxplot showing optical density by lot
18 # Make sure to label your axes!
19 boxplot(..., ...)
20
21
22 # Perform a linear regression of optical density by lot
23 summary(lm(..., data=elisa))
24
25 # Create an ordered factor using Optical.type and verify its type
26 elisa$Optical.type<-ordered(...)
27 class(elisa$Optical.type)
28
29 # Remove 1 as a possible level
30 # Note: even if you eliminate a group of subjects (ex. 1),
31 # because it's a factor, R keeps 1 as a possible level for lot
32
33 # Use droplevels() to remove 1
34
35 # Create stacked results to run aov()
36
37 # Run TukeyHSD on the ANOVA results
```

---

## 2.6 Example Student Code

---

```
1 # KEY
2 # ANOVA – ELISA HIV Optical density readings data
3
4 # Clean up and read in data
5 rm(list=ls())
6 setwd("C:/Example/Code")
7 elisa = read.csv("ELISAHIV.csv")
8
9 # Look at variables and create levels
10 ls.str(elisa)
11 levels(as.factor(elisa$Lot))
12 lot=as.factor(elisa$Lot)
13 summary(lot)
14 levels(as.factor(elisa$Optical))
15 optical = (as.factor(elisa$Optical))
16 levels(as.factor(elisa$Run))
17 run = (as.factor(elisa$Run))
18
19 # Create a boxplot
20 boxplot(elisa$Lot~elisa$Optical, xlab="Lot", ylab="Optical Density",
21         main="Boxplot of Optical Density by Lot")
22 summary(lm(elisa$Optical~elisa$Lot, data=elisa))
```

```
23  
24 # Perform regression  
25 summary(optical)  
26 elisa$Optical.type<-ordered(cut(elisa$Optical, c(0,1,1.5,2),  
27   labels=c("Low", "Medium", "High")))  
28 class(elisa$Optical.type)  
29 elisa$Optical.type  
30  
31 # Remove 1 as a level  
32 # We've verified that optical is an ordered factor  
33 # Note: even if you eliminate a group of subjects (ex. 1),  
34 # because it's a factor, R keeps 1 as a possible level for lot  
35 newelisa <- elisa[1:2,]  
36 summary(newelisa$Lot)  
37 # We remove 1 as a possible level using droplevels()  
38 finalelisa <- droplevels(newelisa)  
39  
40 # We create stacked results to run aov()  
41 summary(finalelisa$Lot)  
42 Combined_Groups <- data.frame(cbind(lot, optical))  
43 Combined_Groups  
44 Stacked_Groups <- stack(Combined_Groups)  
45 Stacked_Groups  
46 Anova_Results <- aov(values ~ ind, data = Stacked_Groups)  
47 summary(Anova_Results)  
48  
49 # We also run TukeyHSD on the ANOVA results  
50 TukeyHSD(Anova_Results)  
51 summary(TukeyHSD(Anova_Results))
```

---

## 2.7 Further Readings

1. Seefeld, Kim and Linder, Ernst. *Statistics Using R with Biological Examples*, University of New Hampshire, Durham, NH Department of Mathematics and Statistics(2007)
2. Julian J. Faraway. *Practical Regression and Anova using R*, University of Michigan, (2002)