Lab 1 - Introduction to R

1. Installing R and Rstudio

R is a computing environment made for statistics by statisticians. It is free and open source, so you can install it on any computer. To install R on a Windows machine, go to https://cran.r-project.org/bin/windows/base/ and download the latest version. Then open the installer file and follow the instructions to install R.

To install R on a MAC computer, go to https://cran.r-project.org/bin/macosx/ and download the latest version. Then open the installer file and follow the instructions to install R.

The simplest way to work with R is to use Rstudio. After installing R, you should install Rstudio by going to: https://www.rstudio.com/products/rstudio/#Desktop and clicking on "Download Rstudio Desktop". Open the installer file and install Rstudio. You should be able to open Rstudio at this point.

2. R Markdown

This document was created using an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. Markdown is great because it allows you to:

1. Run R code, and save your R code in a reproducible format. For example, to embed an R code chunk in your document, you do this:

```
3+2
```

```
## [1] 5
```

```
3^2
```

```
## [1] 9
```

```
3/2
```

```
## [1] 1.5
```

and when you compile (or "knit") the document, the code will be run, and the output will be printed. As you can see, R can work like a calculator, but it can also do SO much more!

2. Working in Markdown lets you type text around your R code so that you can clearly describe what your code is doing. This will make it easy for you to describe what you are doing in your homework solutions.

3. Markdown files can be easily converted to PDF or HTML documents. In this class we will work exclusively with PDF documents. When you click the **Knit** button in Rstudio, a document will be generated that includes both text content as well as the output of any embedded R code chunks within the document.

2.1 Creating an R Markdown File

For most homework assignments, you will be required to turn in your homework in PDF form. The easiest way to do this is to use an R Markdown File. R Markdown files end with a .Rmd extension. You can create one in Rstudio by the following menu options:

File -> New File -> R Markdown

Or you can use a template file. On Canvas there is a "HwkTemplate.Rmd" which provides a basic template for starting a homework file.

2.2 R Markdown Headings

At the top of an R markdown document, you should put a header that specifies the title and author of the document, and then tells markdown what kind of document you want to produce with it. For your homework files, you should make .pdf documents, so your header should look something like this:

```
---
title: Introduction to R
author: Ephraim M. Hanks
output: pdf_document
---
```

This will tell Rstudio to create a PDF document. You can also create HTML documents and Word documents, but in this class we will focus on PDFs.

2.3 Displaying R code and output using R Markdown

R is a very flexible language for statistics.
You can embed an R code chunk in your file like this:

````
```{r}
x=c(1,2,3,4,5,6)
y=c(8,3,4,2,1,0)
plot(x,y)
```
````

Note the three "ticks" "‘ before and after the code, and note the "{r}" designation that tells R markdown to use R to evaluate the code. Other code languages, like PYTHON, could potentially be used, but we will only use R in this class.

Multiple options can be passed to R code. For example:

````
```{r, eval=FALSE}
x=c(1,2,3,4,5,6)
y=c(8,3,4,2,1,0)
plot(x,y)
```
````

will only display the code, and not show the results of running the code, while

````
```{r}
x=c(1,2,3,4,5,6)
y=c(8,3,4,2,1,0)
plot(x,y)
```
````

or

````
```{r, eval=TRUE}
x=c(1,2,3,4,5,6)
y=c(8,3,4,2,1,0)
plot(x,y)
```
````

will display both the code and the output. If you just want to show the output, but not the R code underneath, you can use:

```{r,echo=FALSE}
x=c(1,2,3,4,5,6)
y=c(8,3,4,2,1,0)
plot(x,y)
```

You should try all these out in your own R markdown file.

3. Important Functions in R

R can do almost any statistical task, and is a very good all-purpose tool for data analysis. In this Section, I list a lot of the functions that we might use in class. Run through them in R, so that you can see their output and understand what they do.

3.1 Basic Operations in R

```
3+2
3-2
3*2
3/2
3^2
exp(3)
sqrt(3)
log(3)
```

3.2 Defining and storing scalars and vectors

```
a=4
b=3
a*b

## Vectors

v=c(1,4,3,2)
v

u=1:4
u

## element-wise operations

v+2
v*2
u^2
u+v
u*v

## Subsetting vectors
```

```
v
v[2]
v[1:3]
v[c(1,4)]

## logical arguments

v
v>2
v==2
v>=2

## subsetting using logical arguments
v
which(v>2)
v[which(v>2)]
```

3.3 Creating Data Frames

The R "data.frame" is a very important data structure. It allows you to store a "matrix" of data that are a mix of numeric columns (like response variables) and character columns (like human-readable names of treatments). Consider the farming example from class, in which 10 plots were assigned either Fertilizer X (plots 1-5) or Fertilizer Y (plots 6-10). We will create a data frame that has this information in it, together with the response (corn yield at each plot).

First we will create a numeric vector of the plot number. I'll show you multiple equivalent ways to do this.

```
plot.ID=1:10
plot.ID=c(1,2,3,4,5,6,7,8,9,10)
```

Next we will create the character vector of treatments. Note that you need to put letters in parentheses, or R will assume that the letter is a name of a vector or scalar already defined in R. See Section 3.2 for examples.

```
treatment=c("X","X","X","X","X","Y","Y","Y","Y","Y")
treatment=c(rep("X",5),rep("Y",5))
```

Now we will create a vector of responses

```
response=c(14,12,11,18,20,21,20,17,23,25)
```

And finally, we will put these all together into a data.frame

```
corn.yield=data.frame(plot.ID,treatment,response)
corn.yield
```

```
##    plot.ID treatment response
## 1        1         X       14
## 2        2         X       12
## 3        3         X       11
## 4        4         X       18
## 5        5         X       20
## 6        6         Y       21
## 7        7         Y       20
## 8        8         Y       17
## 9        9         Y       23
## 10      10         Y       25
```

3.4 Summary statistics in R

The "summary" function can tell you a lot about a data frame, as it computes the minimum, maximum, mean, and median of each numeric column.

```
summary(corn.yield)
```

```
##     plot.ID      treatment    response
##  Min.   : 1.00   X:5       Min.   :11.00
##  1st Qu.: 3.25   Y:5       1st Qu.:14.75
##  Median : 5.50             Median :19.00
##  Mean   : 5.50             Mean   :18.10
##  3rd Qu.: 7.75             3rd Qu.:20.75
##  Max.   :10.00             Max.   :25.00
```

You can also use R to get means, variances, and standard deviations of whole vectors:

```
mean(response)
```

```
## [1] 18.1
```

```
var(response)
```

```
## [1] 21.43333
```

```
sd(response)
```

```
## [1] 4.629615
```

or even subsets of vectors

```
mean(response[treatment=="X"])
```

```
## [1] 15
```

```
mean(response[treatment=="Y"])
```

```
## [1] 21.2
```

```
sd(response[treatment=="X"])
```

```
## [1] 3.872983
```

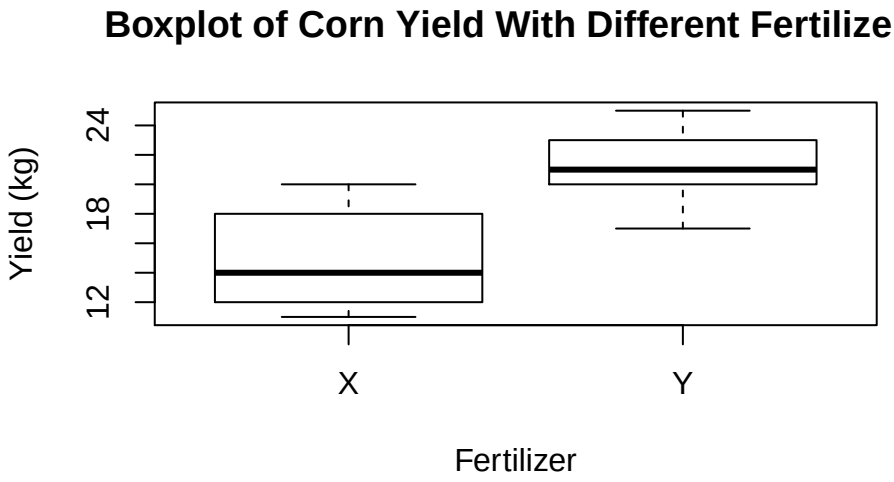
```
sd(response[treatment=="Y"])
```

```
## [1] 3.03315
```

3.5 Plotting in R

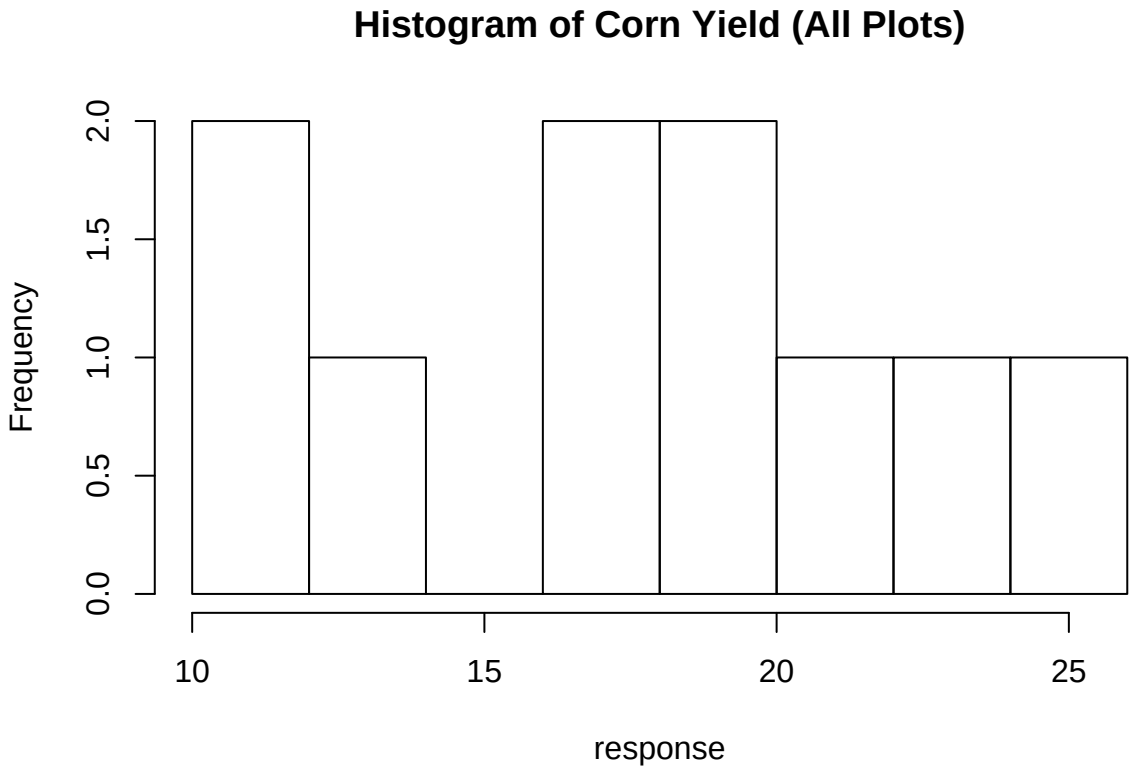
R can be used for plotting data as well. Having the data into a data.frame object can make plotting and data analysis easier. For example, we can plot a boxplot of the corn yield data using the "boxplot" command

```
boxplot(response~treatment,data=corn.yield,
        main="Boxplot of Corn Yield With Different Fertilizers",
        xlab="Fertilizer",ylab="Yield (kg)")
```



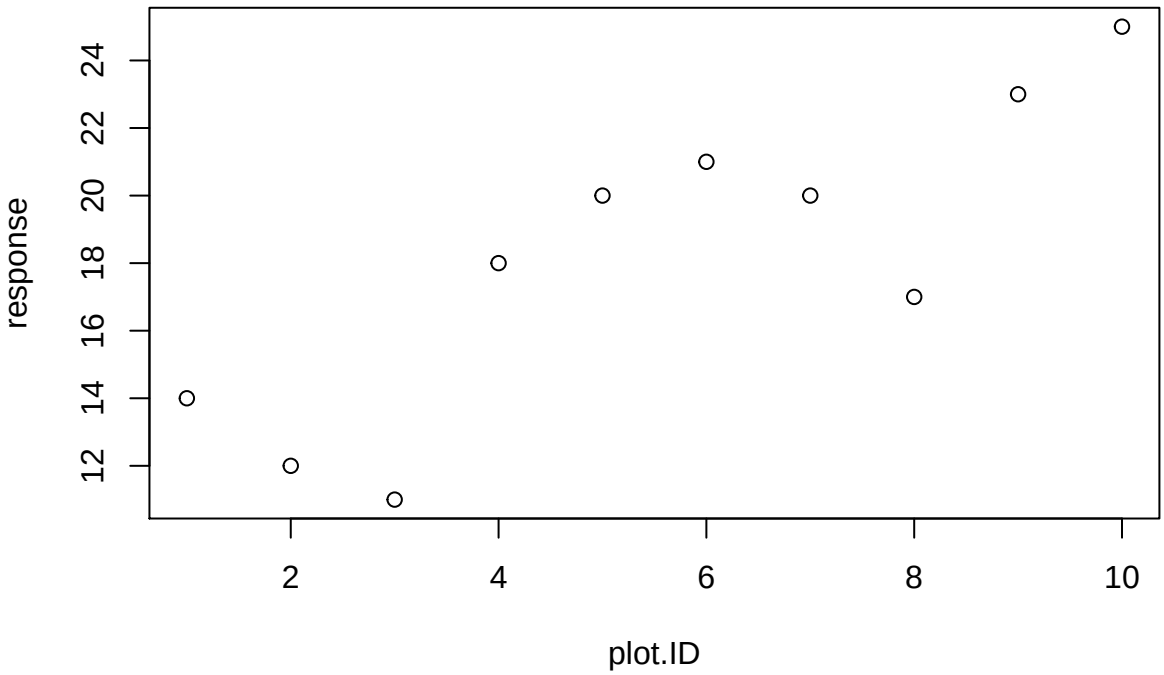
Histograms can be plotted using the "hist" command.

```
hist(response,main="Histogram of Corn Yield (All Plots)")
```



And numeric data can be plotted as a scatterplot using the "plot" command.

```
plot(x=plot.ID,y=response)
```



If you look at the source Rmarkdown code, you can see that you can change the plot size with the "fig.width", "fig.height", and "fig.align" options.

Homework Assignment 1

Consider the following soap experiment, run as a class project by Suyapa Silvia in 1985. The purpose of this experiment is to compare the extent to which three particular types of soap dissolve in water. It is expected that the experiment will answer the following questions: Are there any differences in weight loss due to dissolution among the three soaps when allowed to soak in water for the same length of time? What are these differences?

Four cubes of three different types of soap (Regular, Deodorant, and Moisturizing) were weighed and then placed in baking pans filled with water kept at room temperature. After sitting in the water for 24 hours, the cubes of soap were weighed again. The difference in weights (weight after minus weight before) of the twelve cubes of soap are shown in Table 1.

Table 1: Weight loss in grams of soap cubes.

Cube	**Regular**	**Deodorant**	**Moisturizing**
1	-0.30	2.63	1.86
2	-0.10	2.61	2.03
3	-0.14	2.41	2.26
4	0.40	3.15	1.82

Do the following. Turn in your completed homework as a PDF document created using R markdown. Include all R code used, and clearly label which R code corresponds to which problem. Make your final report clean and clear. Don't just slap your answers and code together. Take the time to write a sentence or two describing how everything comes together.

1. Type the soap experiment data into R. Format the data when you enter it as a data frame with two columns, one for the type of soap and another for the weight lost in the experiment. Each row of your data object should correspond to a single bar of soap (a single experimental unit), so you should end up with 12 rows.
2. Convert the measurements of weight lost to kilograms.
3. Use R to compute the mean and standard deviation of the observed weight lost (in kilograms). Do this first for all 12 data points, and then compute the mean and standard deviation for each of the 3 types of soap.
4. Plot a histogram of all of the observations from the experiment (in kilograms).
5. Plot side-by-side boxplots of the weight lost (in kilograms), with one boxplot for each type of soap.
6. Comment on any observed differences between the soap types that you think would be worth investigating further.