Stat 50 - Lab 1

Intro to R and Descriptive Statistics

Dr. Jorge Basilio

Jan 13, 2020

PART 1

Section 0 - Intro to R

What is R?

R is a powerful, comprehensive, open-source software framework for doing statistics. It is possible to download and install the software on computers, or to use it through a website-interface without downloading anything.

We will use R through a website-interface in the form of a Jupyter notebook or R Markdown document. In fact, what you are reading here is an R-Markdown document that will guide you through the first steps of getting familiar with R.

Why use R?

- It's free!
- Using it on web is free (hassle-free, no messy downloading needed)
- Dowloading it is free (if you want to use it without internet)
- It's open source!
- No hidden algorithms (you can look at source code if you wish–and look "under the hood" so to speak)
- Lots of resources to get help
- It will prepare you for the future!
  - You will likely need to learn some basic programming no matter what you study in college
  - In Pyschology, for example, it's common to learn Python together with R when learning statistics
  - Data Analyst: hot new field where there's lots of jobs!
  - It truly answers the question student's always ask: "When will I ever use this???"

Getting Started

- Make an account at the CoCalc website page using the invitation email I sent. Then **login** to the free Sage server. No nosy questions, just make up a username and set a password. Just be sure to use a modern web browser (Google Chrome, Mozilla Firefox, etc).
- Navigate to the Labs folder and find the Lab_1 folder. Inside you will find a Juptyper Notebook which can run and execute R code titled **"Stat50-Lab_1-YourLastName_YourFirstName-W20"**. Re-name the file with the obvious modifications.

- At the beginning of your Jupyter Notebook, double-click on the "Lab1" text. Replace the text "FirstName LastName" with your actual first and last name. Click "run cell" (looks like a play button) or hit `shift+enter`.
  - By looking at this document, you are encouraged to copy and paste lines of code and modify them :-)
- Have some fun and make a few calculations

Basics of R

Simple Calculations

You can perform all sorts of basic arithmetic using R:

```
2+2
```

```
## [1] 4
```

Of course we know 2+2 is 4. In the above, you'll see the gray box shows the **code**, `2+2` and in the box blow that you'll see the **output** `4`.

We can input much more complicated expressions as well. Just use parentheses liberally!

If we want to evaluate the expression $\sqrt{\frac{2+5\cdot 7}{10}}$ we need to know what the square root, multiplication, and division syntax used.

```
sqrt( (2+5*7) / 10)
```

```
## [1] 1.923538
```

You can add spaces so that it's easier to understand/read the code.

Assignments and Variables

Let's look at one more useful feature of R: assignment.

If you type the symbols less-than and hyphen, you type something that looks like an arrow pointing to the left: `<-`. You can use this arrow to assign values to variables.

For example, you can create a variable `x` and assign it the value `3` by typing this into the console:

```
x <- 3
```

Notice there's no output. This is because we've only defined the variable. We need to "call" the variable.

```
x <- 3
x
```

```
## [1] 3
```

Notice by typing `x` in the next line we DO get an output of `3`

You can name variable almost anything.

If you want to assign a variable a description instead of a number we use quotes:

```
fav_color <- "red"
fav_color
```

```
## [1] "red"
```

A Comment about Comments

Comments are very useful in programming and coding. They are parts of code that are NOT read or implemented by the program but are there to be helpful for humans–that is, for you or me to read :-)

In R, any text after a hashtag is not read by the program for that line.

For example,

```
banana <- 11 # define the variable 'banana'
banana
```

```
## [1] 11
```

Writing regular text paragraphs

If you need to write regular text, then you need to create a different structure that understands plain text called "markdown."

To do this, click on the "Cell" menu, then under "Cell Type", select "Change to Markdown M". Double-click on this cell and you can now start writing.

1. Create a variable named "my_feelings" and assign to it your feelings about statistics. Then have R output your feelings. Make sure that each cell begins with a comment that indicates the problem you are working on.

Entering Data

Let's assume we have a simple data set of: 1,2,2,3,3,3,4,4,4,4,5,5,5,5,5,10.

When entering data "by hand" we use the following code:

```
my_cool_data <- c(1,2,2,3,3,3,4,4,4,4,5,5,5,5,5,10) # define my.cool.data
```

If we want to see our data, we have to type the name

```
my_cool_data
```

```
##  [1]  1  2  2  3  3  3  4  4  4  4  5  5  5  5  5 10
```

Now `my_cool_data` is stored and ready to be studied.

Mean, Median, Standard Deviation, and Variance We calculate the mean of the data set `my_cool_data` as follows:

```
mean(my_cool_data) # mean
```

```
## [1] 4.0625
```

Notice the parentheses after the mean that sandwich the data name.

Similarly, we can calculate the median, standard deviation, and variance as follows:

```
median(my_cool_data) # median
```

```
## [1] 4
```

```
sd(my_cool_data) # standard deviation
```

```
## [1] 2.015564
```

```
var(my_cool_data) # variance
```

```
## [1] 4.0625
```

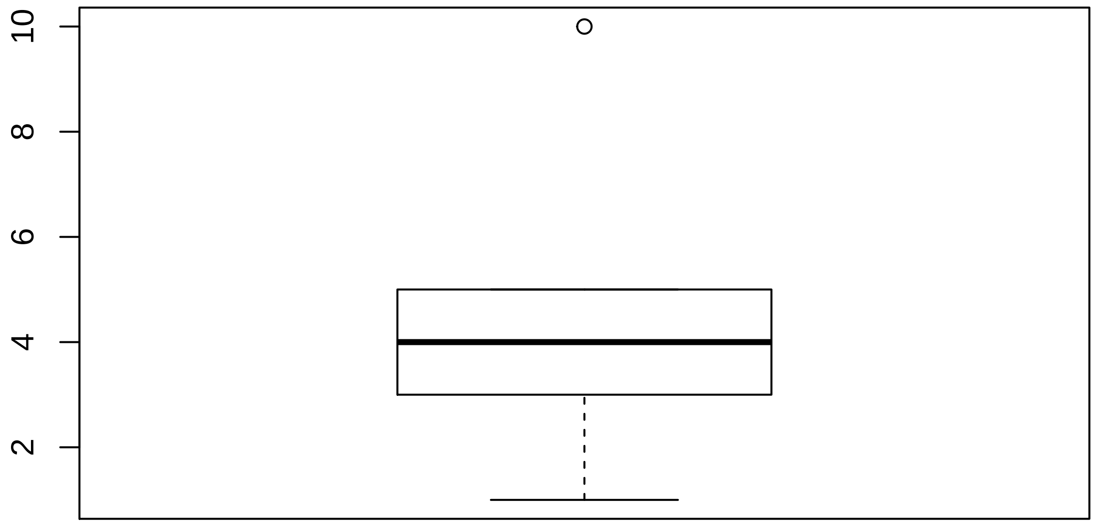
Five Number Summary and Box Plots We can also have the *5 number summary*

```
summary(my_cool_data)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   4.062   5.000  10.000
```

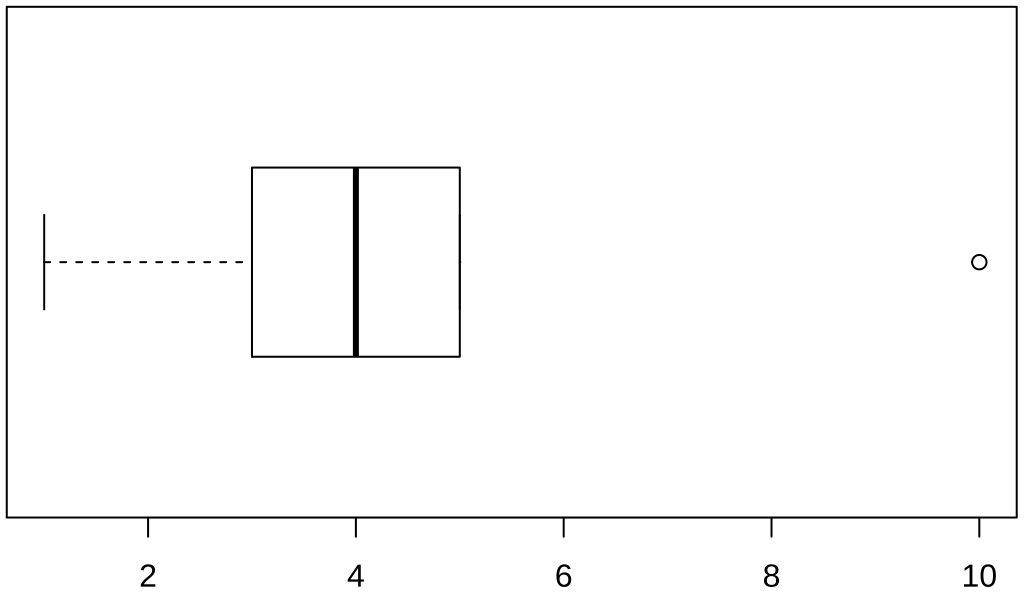
If we want to see the box plot, we use:

```
boxplot(my_cool_data)
```



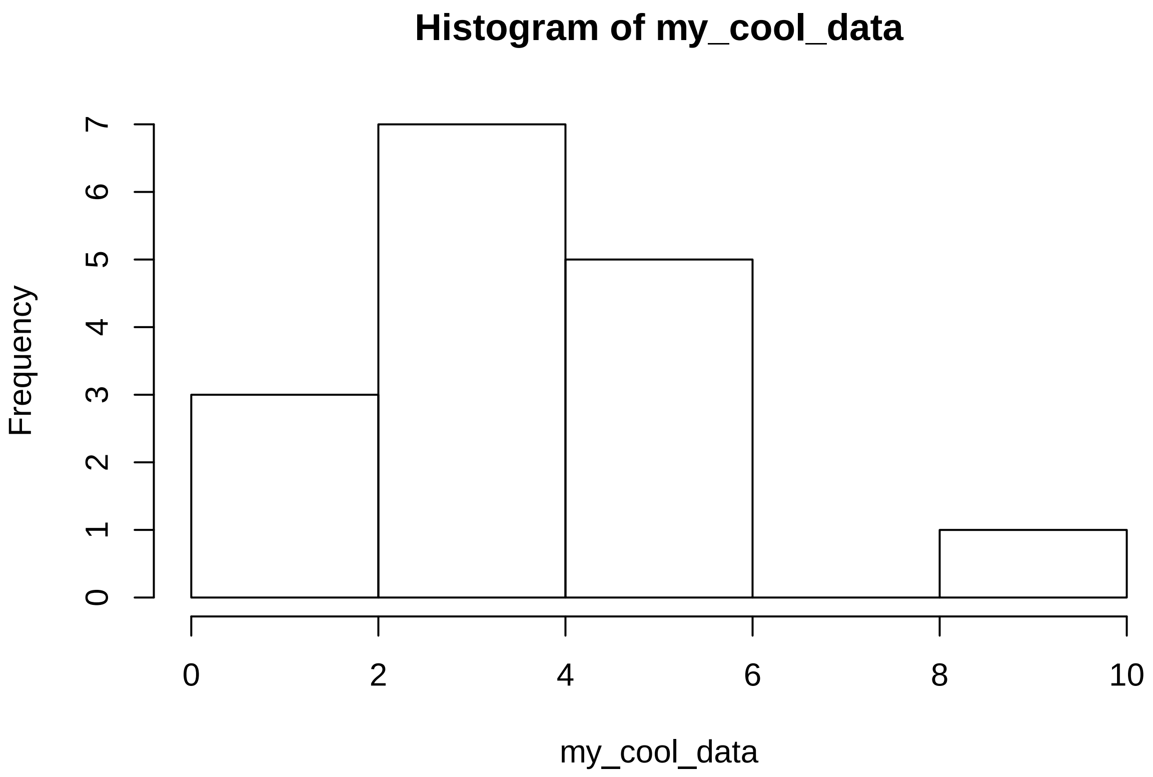
Notice that by default, the box-plot is vertical. We like it to look horizontal so we add the following code `horizontal=TRUE` and use a comma after the data name.

```
boxplot(my_cool_data, horizontal=TRUE)
```



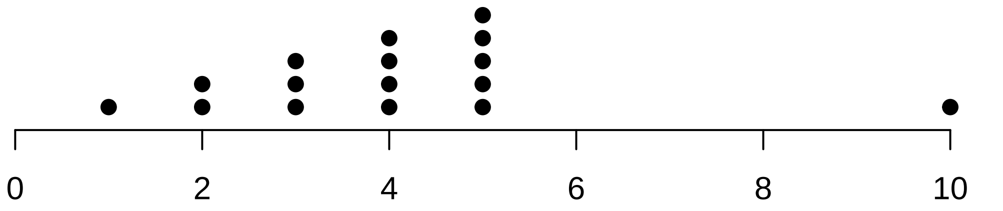
Histograms, Dotplots, Stemplots, and Piecharts A histogram is easy to produce using:

```r
hist(my_cool_data) # histogram
```



A dot plot is easy to produce using:

```r
library(plotrix) # need to use a special package
dotplot.mtb(my_cool_data) # dot plot
```



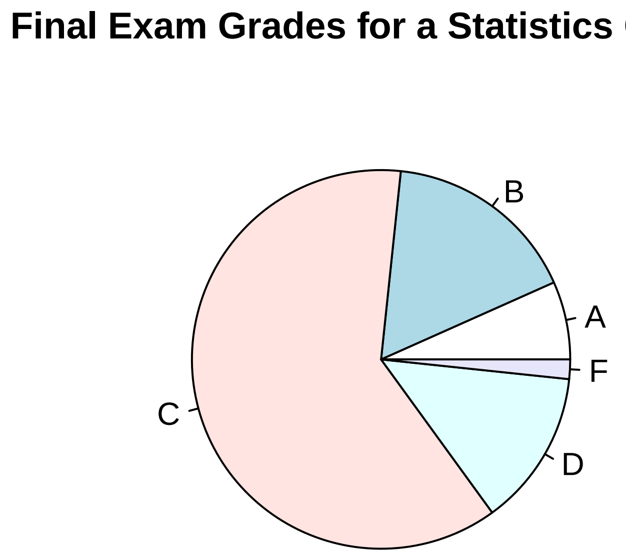
A stem-leaf plot is easy as well. To make it more interesting, we'll introduce a new data set called "ruth".

```r
ruth <- c(22, 25, 34, 35, 41, 41, 46, 46, 46, 47, 49, 54, 54, 59, 60)
stem(ruth) # stem-leaf plot
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   2 | 25
##   3 | 45
##   4 | 1166679
##   5 | 449
##   6 | 0
```

Finally, pie charts require a bit more to set-up.

```r
grades <- c(4, 10 , 37, 8, 1) # the data for each "slice" of the pie
lbls <- c("A", "B", "C", "D", "F") # the labels to apply to each slice (in order)
pie(grades, labels = lbls, main="Final Exam Grades for a Statistics Course")
```



```r
  # Notes:
  # "labels= " will create labels that we defined above
  # "main=" will create a title for the pie chart
```

2. Now, consider the following data set on the "Freshman 15" for September:
   72, 97, 74, 93, 68, 59, 64, 56, 70, 58, 50, 71, 67, 56, 70, 61, 53, 92, 57, 67, 58, 49, 68, 69, 87, 81, 60, 52, 70, 63, 56, 68, 68, 54, 80, 64, 57, 63, 54, 56, 54, 73, 77, 63, 51, 59, 65, 53, 62, 55, 74, 74, 64, 64, 57, 64, 60, 64, 66, 52, 71, 55, 65, 75, 42, 74, 94
   Find:
   [a] Mean, Median, Standard Deviation, and Variance
   [b] Five Number Summary and Box-plot
   [c] Histogram, Dot Plot, and Stem-Leaf Plot.
   [d] Explain the similarities and differences between the Histogram and Dot Plot. Note: Create a "markdown" cell so you can type your answer in regular text.

PART 2

Section 2 - Data Analysis

Reading "big" data from a file

We need a special package that can read "Excel" files (files that end with `.xls` or `.xlsx`) called a "library". To load a library we need to know the name of it. The library we want to load is called "readxl".

```r
#install.packages("readxl")
library(readxl)
```

Type that into a new cell in your lab notebook.

Next, we want R to read the data in the file "06-Freshman15.xlsx". To do this, type the following code in a new line:

```r
library(readxl)
fresh_15_full <- read_excel("06-Freshman15.xlsx")
```

To see what is in the file "06-Freshman15.xlsx", we use the function `str()`

```r
str(fresh_15_full) # print out table of data listing first few values
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    67 obs. of  5 variables:
##  $ SEX      : chr  "M" "M" "M" "M" ...
##  $ WT SEPT  : num  72 97 74 93 68 59 64 56 70 58 ...
##  $ WT APRIL : num  59 86 69 88 64 55 60 53 68 56 ...
##  $ BMI SEPT : num  22 19.7 24.1 27 21.5 ...
##  $ BMI APRIL: num  18.1 17.4 22.4 25.6 20.1 ...
```

```r
names(fresh_15_full) # print out names of variables
```

```
## [1] "SEX"       "WT SEPT"   "WT APRIL"  "BMI SEPT"  "BMI APRIL"
```

```r
dim(fresh_15_full) # print out dimension
```

```
## [1] 67  5
```

The 66 observations tells us there are 66 pieces of data for each variable. Note that the first value is the variable names so we subtract 1. There are also 5 variables.

We see that the variables are `SEX` in first row, `WT SEPT` in second row, etc.

2. Explain what each of the five variables are. Be sure to write your answer using a markdown cell.

Next, let's start some computations.

Work in Progress…