Lab One: An Introduction to R/Rstudio

STA 111: Probability & Statistical Inference

Lab Objective

To become familiar with the software package R.

Lab Procedures

We will be using RStudio to explore the applications of some of the concepts we will learn about in class and also to analyze data. RStudio is a front end for the R programming language. You will need to download both R and RStudio on your personal computers.

Go to https://cran.rstudio.com to download R and then https://www.rstudio.com/products/rstudio/download/#download to download RStudio. In the next lab you will learn about the fundamental building blocks of R and Rstudio, but for now lets just make sure you can log on and run some code.

The panel in the upper right contains your workspace as well as a history of the commands that you've previously entered. Any plots that you generate will show up in the panel in the lower right corner. The panel on the left is where the action happens. It's called the console. Every time you launch RStudio, it will have the same text at the top of the console telling you the version of R that you're running. Below that information is the prompt. As its name suggests, this prompt is really a request, a request for a command. Initially, interacting with R is all about typing commands and interpreting the output. These commands and their syntax have evolved over decades and now provide what many users feel is a fairly natural way to access data and organize, describe, and invoke statistical computations.

Getting Started:

You can use R as a calculator. Type $2+2$ right after the $>$ on the console. You can save the result to an object that you can access later by typing $x = 2+2$. Type x to verify the result was saved. A few commands to get you started:

- Creating a vector: $x = c(1, 2, 3, 4, 5, 6)$

- Creating a matrix: $y = \text{matrix}(c(1, 2, 3, 4, 5, 6), \text{nrow} = 2, \text{ncol} = 3)$. You can create a matrix by specifying the number of rows, columns, or both. Thus, the previous code is equivalent to $y = \text{matrix}(c(1, 2, 3, 4, 5, 6), \text{ncol} = 3)$ and $y = \text{matrix}(c(1, 2, 3, 4, 5, 6), \text{nrow} = 2)$. You can also choose to arrange the vector into the matrix by row or by column. Try $y = \text{matrix}(c(1, 2, 3, 4, 5, 6), \text{nrow} = 2, \text{byrow} = \text{TRUE})$ and check the value of y

- Creating a data frame: There are various ways to create a data frame. For now, you can create a simple one by typing $z = \text{data.frame}(\text{A} = c(1, 2, 3), \text{B} = c(4, 5, 6))$. In this example, each vector would be passed into the data frame as a column with the column names ≡ c(A,B). Verify that by typing: $\text{names}(z)\text{ or colnames}(z)$. You should use data frames if columns (variables) can be expected to be of different types (numeric/character/logical etc.). Matrices are for data of the same type. We will deal with data frames most of the time.

- Creating a histogram: hist(x) or hist(c(1,1,1,3,4,5))

- Creating a table: table(x) or table(c(1,1,1,3,4,5))

- Checking the dimensions of a matrix or data frame: dim(z).

Actual Data:

- Create a new data frame from the data below for the ten most fatal earthquakes on record. Call the new data, "Data".Use " " for the countries to signify that they have string values. For example, c("Haiti","China") would create a string vector with those two countries.

Country	**Deaths**
Haiti	92,000
China	242,769
Iran	150,000
China	235,502
Indonesia	230,210
Syria	230,000
China	820,000
Iran	200,000
Turkey	240,000
Japan	142,800

- After inputting the data, lets try the `which` command.

  Type `which(Data$Country==“Turkey”)`.

  Also try `which(Data$Country==“China” & Data$Death==820000)`.

- How many earthquakes killed 200,000 or more people? With ten cases its straightforward to look at the data and get an accurate count. But with a longer dataset, counting the incidences of each number by hand would be cumbersome. In such settings, you can make life easier by sorting the numbers in increasing order, then counting the incidences. Do that by typing `Data[order(Data$Deaths),]`. Even better, you can find the exact rows that satisfy `Deaths ≥ 200,000` by typing `Data[which(Data$Deaths ≥ 200,000),]`.

- Let’s sort the data first by country and then by death toll (i.e. have all countries in alphabetical order with fatalities listed in increasing order by country).

  Type `Data[order(Data$Country,Data$Deaths),]`.

Creating a Reproducible Lab Report

We will be using a markdown language, R Markdown, to type up the lab report. This allows you to complete your lab entirely in RStudio as well as ensuring reproducibility of your analysis and results. To help get you started, a template is provided for you. Download the template from http://www2.stat.duke.edu/~oma9/STA111_SummerI_2018/Labs/Lab.Rmd, save it locally on your computer to a folder of your choice and open it in Rstudio. We need to install a package to help knit the markdown file. Type `install.packages(”knitr”)` in your console.

All you need to do to complete the lab is to type up your brief answers and the R code (when necessary) in the spaces provided in the document. Earlier in the lab, spaces are provided for you to enter R code chunks. Later in the lab, youll need to figure out whether code is needed to answer a particular question, and if so, a new chunk can be inserted by clicking on the Insert Chunk button (dropdown menu under Chunks on the upper right corner of your markdown document). Before you keep going, **type your name in the markdown file**. Then click on Knit to HTML or Knit to PDF and you’ll see your document in a new pop-up window. You can save the pdf and email it to yourself or simply submit directly on Sakai.

Lab Questions:

Load in the data set Forbes94, which contains the 1994 compensation information for Chief Executive Officers (CEOs) of several large companies. You can load the data by typing:

```
Forbes94=read.table("http://www2.stat.duke.edu/~oma9/STA111_SummerI_2018/Labs/Forbes94.txt",header=T)
```

When you get a data set, the first thing to do is to figure out how many variables and how many units of observation you have to play with. Do you remember how to check the number of variables and individuals? Let's get into some data analyses. Compile your answers in the markdown file. You are permitted and encouraged to talk about questions with your classmates, but write up your lab report with your own words.

1. *R displays missing values with NA.* True or false: There are more than five CEOs whose values of total compensation are missing in the data file. (Hint: You can do this by sorting the data and browsing).
2. What is the salary (not total compensation) of the CEO of Duke Power? (Hint: try the `which` command).
3. Of all CEOs, which has the highest total compensation? Which has the lowest total compensation?
4. Which industry type has the highest average CEO total compensation? Be careful not to read the decimals incorrectly when you answer the question. (Hint: Try the `summary` command).
5. How many of these CEOs got their undergraduate degree from Duke?
6. Highest attained educational degree is in the variable GradDegree. Which degree has the highest total compensation: MBA (business), JD (law), MD (physician), PhD, or no graduate degree? Use highest average total compensation as your criterion, and choose only from these categories.

This ends the lab. Remember to turn in your lab reports on Sakai.