Lab One: An Introduction to R/Rstudio

STA 111 (Summer Session I)

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 can use RStudio by signing on to https://vm-manage.oit.duke.edu/containers. Click on the link that says "Click here to log in to your RStudio environment" , and log on using your NetID and password. 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.

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 equiv-

alent 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 $\equiv$ c(A,B). Verify that by typing: names(z) 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).

We will learn many more commands as we go on.