MATH203 - Introduction to R-Studio

Tutorial Notes - James McVittie

1 Introduction

The programming language that will be used in MATH203 this semester is known as R-Studio. With the increase in size of available data from the internet, statisticians and researchers in many fields have become reliant on computer software like the R-studio program to perform their analyses. This document will go step by step in how to download R-studio from the internet, the different windows in the R-studio program, how to open a dataset in R-studio and finally how to perform basic statistical computations (average, variance, etc.) and output graphs (histograms, boxplots, etc.).

2 Downloading R-studio

Open your internet browser (i.e. internet explorer, firefox, google chrome, etc.) and enter in the URL line the following address: https://www.rstudio.com/. Then on the main website page, hover the mouse cursor over "Products" and click on R-studio. In the new window, there will be a choice for "Desktop" or "Server". Click on the "RStudio Desktop" in blue and then click on the blue box "DOWNLOAD RSTUDIO DESKTOP". On the new page there will be many different installation files listed. Using the options from the "Installers for Supported Platforms" list, click on the appropriate file that matches to your computer's operating system (i.e. Windows, Mac, Ubuntu, Fedora). Finally, Save and Run the execute file to install R-studio.

3 The Basics of R-studio

Now that you've downloaded R-studio onto your computer, open the program. In the opening screen, you will see 3 windows: on the left "Console", on the top right "Environment / History" and on the bottom right "Files / Plots / Packages / Help / Viewer". Note this layout is from the Windows OS and other OS layouts may differ slightly.

3.1 Console

In this part of the window, you will be typing your code to produce different outputs. Unlike programs like Microsoft WORD or Microsoft EXCEL, R-studio requires that you

tell it every single step in a procedure. For instance, if you were to fry an egg, you simply cannot tell a computer to "fry an egg" but rather you need to give step by step instructions like open the fridge, remove 1 egg, close the fridge, heat stove, obtain pan, break egg into pan, fry, remove egg. The console is where we give our step by step commands to R-studio.

3.2 Environment / History

As described in the Console section, we will be giving R-studio commands and sometimes we will be creating variables to store data. For instance, in the console line, if you assign the value 5 to the variable x by the following command

$$\mathrm{x} = 5$$

then in the Environment window, you will see the variable x associated with the number 5. By clicking on the History tab, you will see every single command you have ever typed in R-studio. This is very similar to the "History" on your internet browser that saves every website you have visited.

3.3 Files / Plots / Packages / Help / Viewer

In the third section, we have 5 tabs. The first tab called "Files" lists all the directories in your computer and allows you to view where different files (datasets for example) are stored. Once a dataset has been opened, we can create a plot by using different commands (see later section) which can be viewed, and saved in the "Plots" tab. The third section, which will not be used very much is called "Packages". Statisticians all over the world have developed pre-built packages in R-studio to either calculate different values or produce different plots etc. so that other researchers do not need to type all the raw code by hand. Referring back to the "fry an egg" analogy, a package is like a prepackaged pre-cooked egg, all you have to do is warm it up. The fourth section is "Help". If you are ever unsure of how to execute a particular command, there are many extensive manuals and resources available to read from. Finally, the "Viewer" tab which we will not be using in the course.

4 Working with R-studio

4.1 Using it like a Calculator

First and foremost, R-studio can be used like a calculator. In the console window, we can type the following command:

$$5 + 9$$

and R-studio will output 14. R-studio can perform subtraction operations as well with the following command as an example:

$$6 - 2$$

which outputs the number 4. To perform mutiplication we use the asterisk symbol between two numbers $*$. To perform division, we use the backslash symbol as seen in the following example:

$$4/2$$

which outputs the number 2. For squaring a number we use the up arrow and place a 2 next to it as seen in the following command:

$$5 \wedge 2$$

which outputs the number 25. Using order of operations (BEDMAS) and choices of brackets, more complicated calculations can be performed through the console window like the following:

$$((5 + 2) * 4) \wedge 1.5$$

which outputs the number 148.1621.

4.2 Opening a dataset

Using R-studio as a calculator is useful for simple calculations; however, in most problems, a dataset is already given to the researcher and the statistician must perform an analysis. Here, we will show how to import a dataset into R-studio. In MyCourses, a file named "JamesTut1" is listed. Save this file somewhere you will remember. To open a dataset, go to the menu at the top of the page and click on "Tools", then to "Import Dataset" and then "From textfile". Open the "JamesTut1" file. After the file has been imported, a new window will appear displaying the data, in the Environment section, the data will be listed, and in the Console section two lines of code will have appeared: one to open the dataset and the other to view the dataset. R-studio has automatically set the variable name of the dataset to JamesTut1 as this was the name of the file. Note that instead of using the point and click interface, you could have manually entered the line of code that reads in the dataset by using the command: *read.csv*() for csv type files and in the brackets typed the directory of the dataset file.

4.3 Some basic calculations

Now that we have imported our first dataset, we will do some basic calculations. The dataset has 3 columns: "id", "height" and "weight" of 10 individuals. Rather than calculating by hand, suppose we want to obtain the average height of all the individuals. Then in the Console window, we type

mean(JamesTut1$

Then R-studio will automatically give a drop down menu of the three variables listed and so we click on height. (Note that R-studio automatically adds the closed bracket on the end of the mean command). Thus the final command is as follows:

```
mean(JamesTut1$height)
```

which outputs 176.3. This command is literally saying to calculate the mean from the JamesTut1 file with column height. "mean()" is the command for the average, "JamesTut1" is the variable name for the dataset, "$" tells R-studio to access something in the object that comes before the symbol and "height" is the column that we wish to access. Similarly, we can calculate the mean of weight by an almost identical command:

```
mean(JamesTut1$weight)
```

which outputs the value 155.6.

Later in the course, you will learn about a statistic known as the sample variance which describes the spread of the dataset. The command we use to calculate the variance is "var()". For example, if we were to calculate the variance of the height, we would type

```
var(JamesTut1$height)
```

and we obtain the value of 39.78889. There are many commands that we can add on such as taking a square root: sqrt() or squaring a number $()^2$ etc. which are useful for calculating values like the standard deviation.

4.4 Creating Some Plots

In addition to being able to calculate values from our dataset, it is also very useful to view our dataset in a meaningful way. One of the most basic plots we can create is a scatterplot. First, we will create a plot of height against the order in which the datapoint appears. Similar to commands in the previous section, we use the following command:

```
plot(JamesTut1$height)
```

which generates a scatterplot in the "Plots" tab of R-studio. If we want to save this image, we click on "Export" and save it as the desired file type.

As an exercise, create a scatterplot of the weight variable.

Suppose now that we want to plot height (x-axis) versus weight (y-axis). The plot function also alows us to create this plot by adding in a second argument (i.e. the y-axis variable). So we use the following command:

```
plot(JamesTut1$height, JamesTut1$weight)
```

Again, the plot is visible in the "Plots" tab in the bottom right window.

Suppose now that we wish to create a histogram of the height data, then we use the command "hist(JamesTut1$height)" and if we wanted to create a boxplot of the height we can use the command "boxplot(JamesTut1$height)".

5 Final Comments

From the previous two sections, it is clear that the commands in R-studio all follow the same format of "DOTHIS()". Please take the time to experiment with the different commands explained above to become familiar with the layout of the R-studio program. Please note that with programming, R-studio is VERY case sensitive and a minor typo will not allow your code to run. If there are any specific issues with your code not running or not producing the desired output, please attend either Professor Yang's office hours or my own to help fix the bug in your code.