Applied Economic Forecasting
Tutorial on R Studio I

Spring 2020

VIRGINIA TECH.

Section 1

R Introduction

Roadmap

1. What is `R`?
2. Why `R`? Three useful features
3. Why R studio?
4. A Short Tour of R Studio
5. Script file

What is `R`?

`R` is a **programming language** used for statistical analysis and graphics. It is based on S-plus, which itself was based on S, a programming language originally developed by AT&T.

Why `R`?

- Open source, cross-platform, and **free**
- Great for reproducibility
- Interdisciplinary and extensible
- Tons of learning resources
- Works on data of all shapes and sizes
- Produces high-quality graphics
- Large and welcoming community

`R`: Object-Oriented Programming

Unlike many other statistical softwares such as SAS and SPSS, `R` will not spit out a mountain of output on the screen.

Instead, `R` returns an **object** containing all the results. You, as an user, have the flexibility to choose which result to be extracted or reported.

This feature allows us to write faster yet more compact code. For example, a common theme in `R` programming is **avoidance of explicit iteration**. Unlike many other statistical softwares, explicit loops are discouraged.

Instead, `R` provides some functions that could allow us to express iterative behavior implicitly.

R is also *polymorphic*, which means that a single function can be applied to different types of inputs (much more user friendly).

Such a function is called a *generic function* (If you are a C++ programmer, you have seen a similar concept in *virtual functions*).
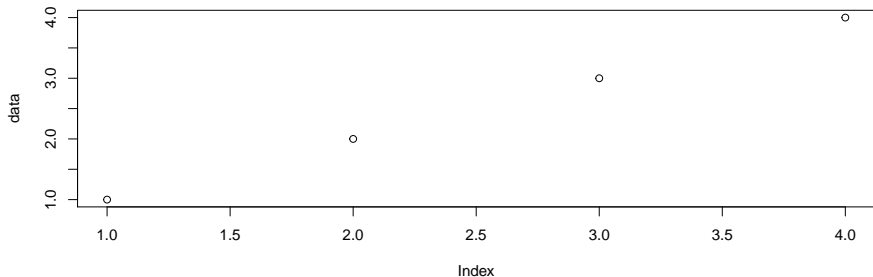
Lets look at one example `plot()`

1. Plot a vector of numbers
2. Plot some model results
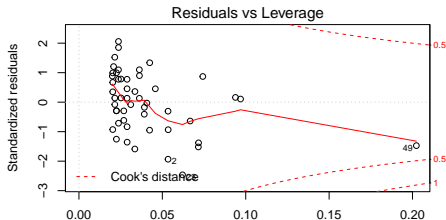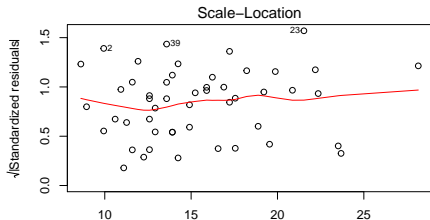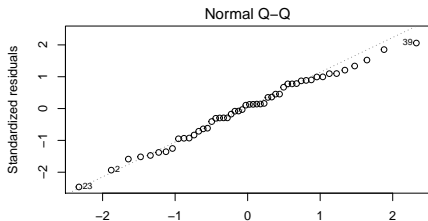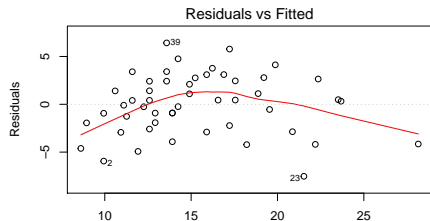
No matter which purpose, we use the same function.

```r
data<-c(1,2,3,4)
plot(data)
```

```
# Regression Analysis
par(mfrow=c(2,2),mar=c(2,4,2,2))
results<-lm(speed ~ dist,data=cars)
plot(results)
```

Why R Studio?

- `R` Interface is **ugly**!

- Many students in this class are much more familiar with Windows operation system and have never been exposed programming before, so we will use R studio, one of the free Graphical User Interfaces (GUIs) that have been developed for `R`.

  - R studio should really be considered as *integrated development environments* (IDEs), since it is aimed more toward programming.

- Easy publishing of reproducible documents such as reports, interactive visualizations, presentations, and websites.
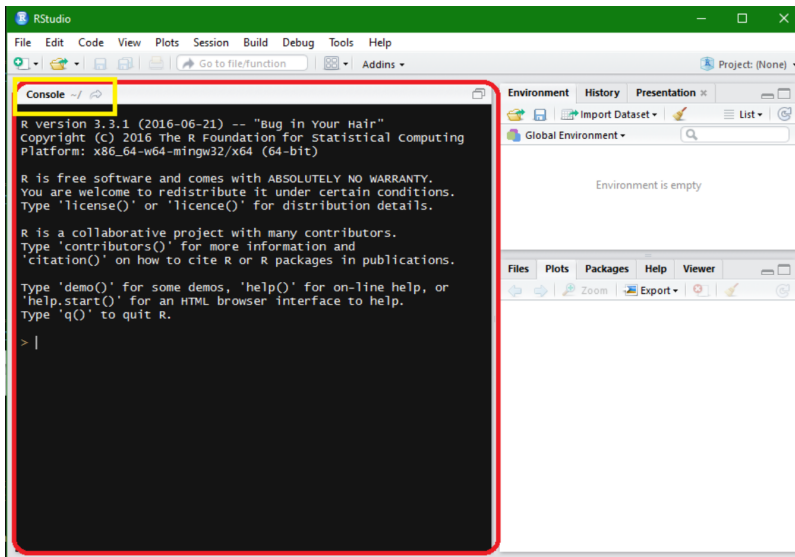
Initial Start

When you first (like very first time) open R studio you will see three panels.
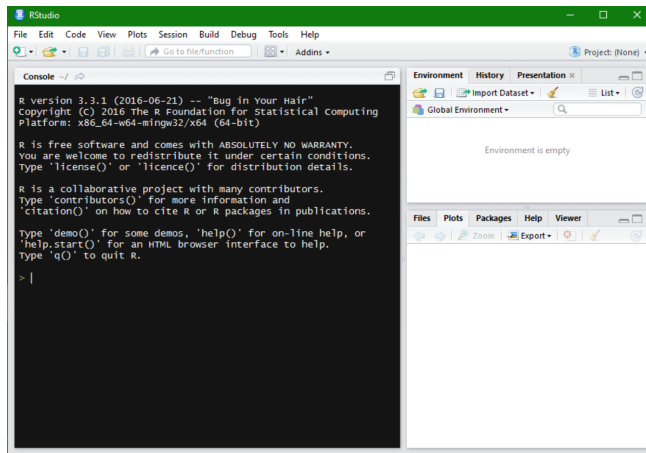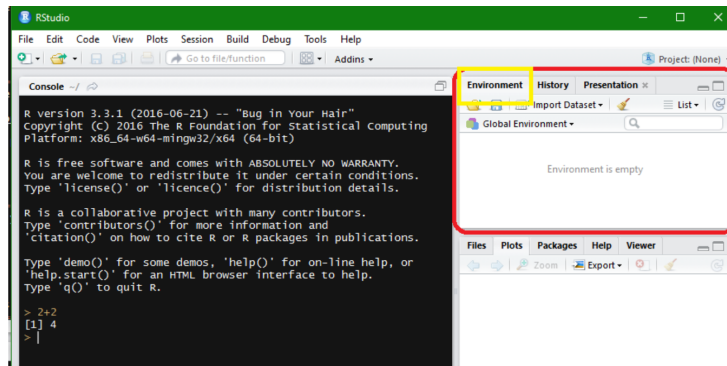
R Studio: A short tour

Console

Console

1. Everytime 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.
2. Below that information is the prompt, $>$. As its name suggests, this prompt is really a request, a request for a command.
3. Initially, interacting with R is all about typing commands and interpreting the output.
4. These commands and their syntax have evolved over decades (literally) and now provide what many users feel is a fairly natural way to access data and organize, describe, and invoke statistical computations.

The console is where you type commands and have them immediately performed.
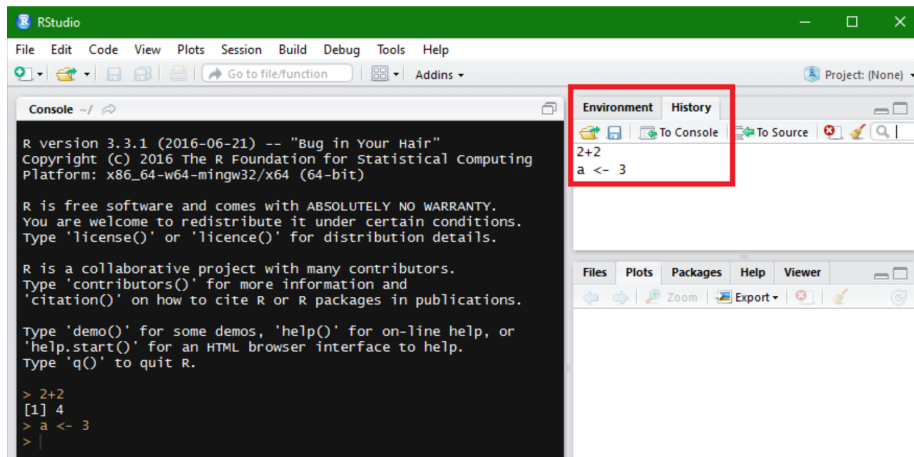
R Studio: A short tour

Environment The panel in the upper right contains your workspace (aka Environment)

❶ This shows you a list of objects/variables that R has saved.
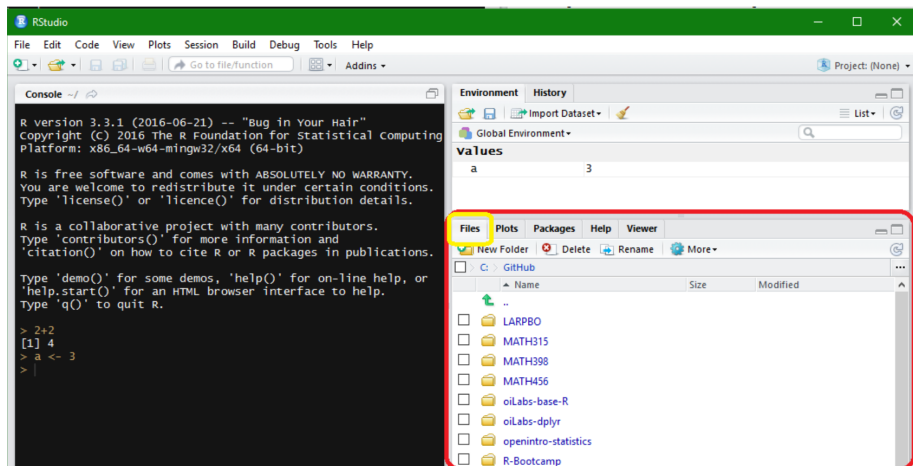❷ For example here a value of 3 has been assigned to the object a.

R Studio: A short tour

History Up here there is an additional tab to see the history of the commands that you've previously entered.
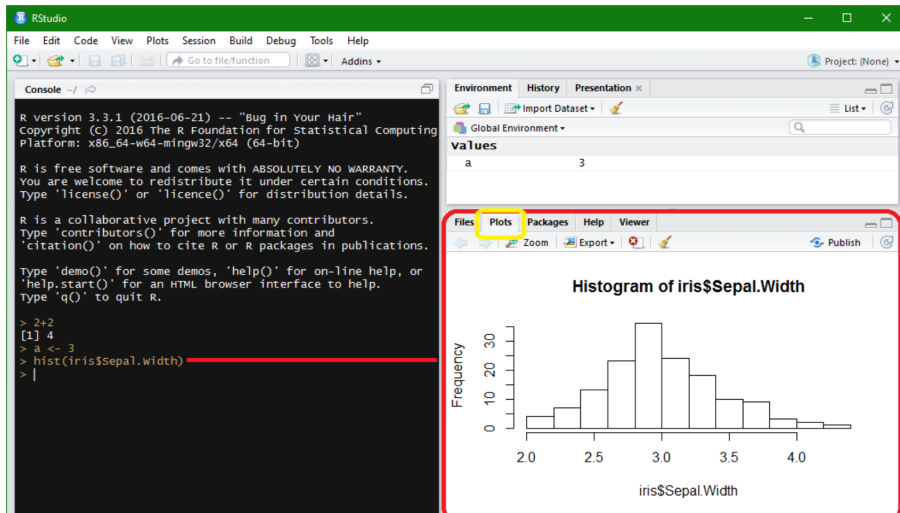
Files The files tab allows you to open code/script files within R studio.

R Studio: A short tour
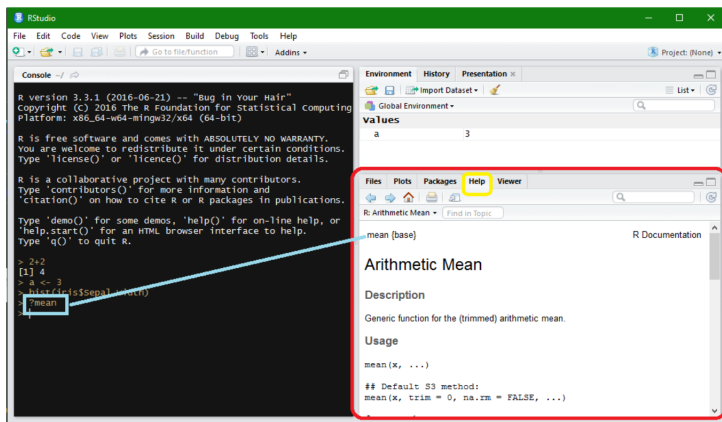
Plots Any plots that you generate will show up in the panel in the lower right corner.

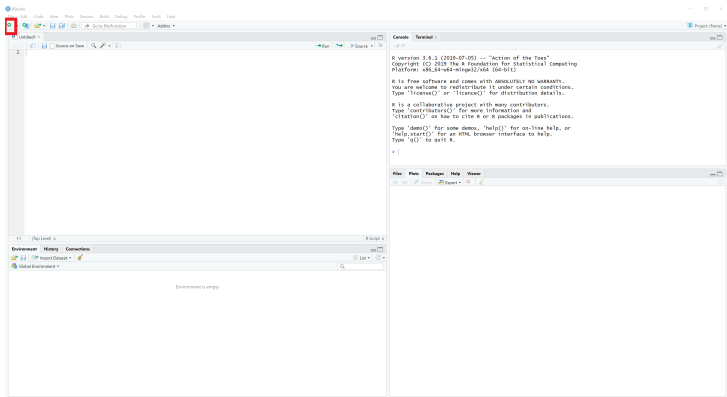**Help** To check the syntax of any function in R, type ? in front of the function name to pull up the help file.

For example here I typed ?mean to get the help file for the mean function. Admittedly these aren't the most helpful of files at times.

Script File The top left is your editor window, where you write code or script, the console is now at the bottom. **I usually change it**

The picture above illustrates my preferred style in R Studio.

R Script

Most of `R` users typically submit commands to `R` by typing either in console or editor panel, rather than clicking a mouse in a Graphical User Interface (GUI).

Script is nothing but **a collection of commands**

There are at least two advantages of doing so:

1. This allows us to run a bunch of results altogether by putting a collection of commands in a file (i.e., "script").
2. It is also a lot more transparent and straightforward to **share** and **replicate** what you have done.

In our class, you will do this!

R Script

Task 1: Create a script file

❶ Open R Studio and go to File > New > R Script. This will open a blank text document.[1]

❷ In the document, type

```r
x = 5   # Assign the variable x a value of 5
x == 5  # Does x = 5? Notice the double ==
```

Highlight both lines of code and click the button marked "Run". If everything is working correctly, the console should display TRUE.

OR, pressing **CTRL + ENTER** or **COMMAND + RETURN** depending on whether you're running Mac OSX, Linux or Windows.

❸ go to "File > Save As", and choose a file name.

---

[1]Two alternative ways are: **CTRL + SHIFT + N** or press the button marked "+", just below File, and select R Script