Getting Started with R*

Whitt Kilburn
Associate Professor, Dept. of Political Science

Faculty Fellow, Data Inquiry Lab
http://www.gvsu.edu/datainquirylab

Grand Valley State University
Fall 2015

comments or suggestions?
kilburnw@mail.gvsu.edu

This brief guide reviews how to get R up and running, either in the computer labs or on your own machine. It provides an overview of the icons and file menu commands in the R windows, how to use a script file, how to save your work, how to troubleshoot errors, and where to find further information.

*This guide is a draft, updated periodically.

Contents

1 Introduction

Welcome! It's normal to find R a bit overwhelming at first. The software's learning curve is a bit steep, but it flattens out later, after you understand the basic steps of opening and saving analysis files and writing script file code. — The script file code is simply the commands, in R syntax, typed into a text file that instruct R what to do. Before you continue with this guide, if you are unsure about what R is, here's a short 90-second video: `http://www.revolutionanalytics.com/what-r`.

After reading this guide:

1. Install your own copy of R, or locate it in a campus computer lab.
2. Open R and identify what the different windows and menus commands do.
3. Set your R working directory.
4. Use the command line in the R console window.
5. Within R, open a script file (a text file containing R code).
6. Run lines of the script file.
7. Write and save an R script file.
8. Save 'your work' which includes saving
   (a) text output from the console window
   (b) graphics,
   (c) your script file
   (d) an R command history file
   (e) your R workspace.

1.1 R vs. R Studio

The core software application is R. It is freely available to anyone and available for you to download or use on campus lab computers.

RStudio is a commercial software product that is built upon R. RStudio provides some additional features, but its main purpose is to support large scale professional data analysis and communication. Most features of RStudio are freely available; to download your own copy, to to `http://www.rstudio.com/products/rstudio/download/`. It is also located in the campus computer labs alongside R.

1.2 Accessing R

There are two routes to accessing R, from within in a campus computer lab or on your own personal computer.

1.2.1 R on campus

On a campus computer lab, look on the desktop. Find the folder that contains lab applications, labelled "Lab Applications". (Or, because it changes names every now and then, look for "Applications"). Open it, and look for a folder for "Statistics". Open this folder, and you will see a folder titled "R and R Studio" or individual software application icons. Click this folder, or otherwise look for a folder titled "CRAN R". In the "CRAN R" folder, look for a small "R" icon. Double click this icon to open R. Once R starts up, what you see is the "R console", with a prompt, >, waiting for you to issue it a command.

1.2.2 R on your own computer

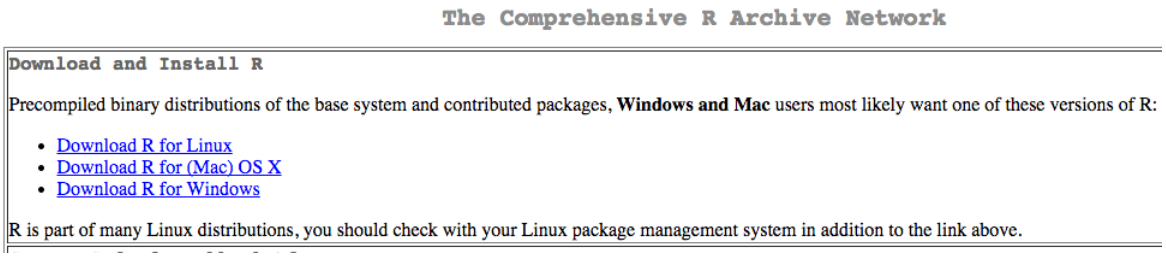

Figure 1: Go to http://cran.r-project.org/ and follow the links to download R for your operating system.

For your own computer, go to the R Archive Network at `http://cran.r-project.org/`. The site contains the files for downloading it to your own computer. From there, you should see the links as they appear in Figure 1. Follow the one for your operating system. For Macs, you'll download the .pkg file for your version of Mac OSX, such as Mavericks or Yosemite. For Windows, click the link to the "base" files, then download the file linked from "Download R 3.X.X for Windows". (The version you see will be 3.2.2 or a more recent version.)

You may be prompted to select a "Mirror" to download R. You can download it from anywhere, but it will probably be fastest to download within the USA or within Michigan. So scroll down to the USA and choose the link `http://cran.mtu.edu`. If R starts downloading automatically, without selecting a Mirror, that's OK, too.

Follow your usual steps to install software, accepting the default installation options unless you can identify a specific reason why you would not want to do so. Double-click to start up R. In a few seconds, you should see a screen similar to Figure 2 on page 5. While Figure 2 displays the Microsoft Windows system, the Mac arrangement of R's windows, drop-down menus, and buttons, is nearly identical.[1]

2 A Brief Tour of R's Console Window and Graphical User Interface

The window in Figure 2 on page 5 is the Console, where you interact with R by either writing commands directly into the console window at the > prompt, or via a script file, described later in this guide. The

[1]From your perspective, R will work almost the exact same way, regardless of whether you are working on a Windows machines or a Mac. You write commands in a script file, or enter the commands directly on the prompt, and R does what it is told to do. Of course, some of the shortcut keys are different. And there are some differences in how you save graphical output, such as scatterplots and histograms. But these differences are easy to accommodate. In this guide, the differences are are pointed out as needed. Unless otherwise noted, proceed through the guide without worrying about whether you are on a Mac or Windows machine. An appendix provides a tour of the Mac version of R.

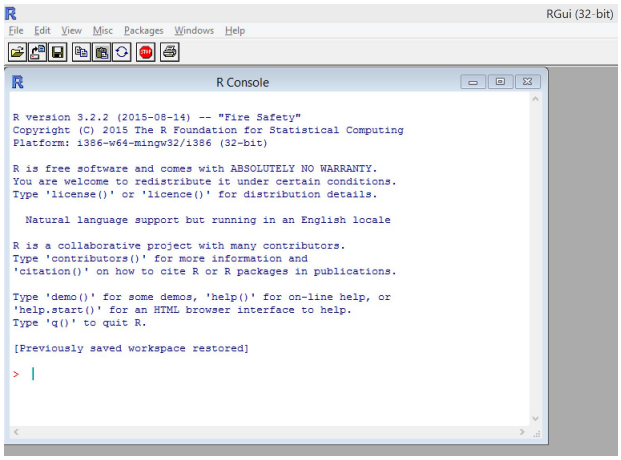

Figure 2: The R console window, where you send commands for analysis. The red > symbol is the command prompt, waiting for you to issue a command. Try typing 2+2 and observe the result.

icons at the top left, starting below the File and Edit drop-down menus provide shortcut buttons to commands for opening files and R workspaces, copy-pasting files, and getting help. Hold your mouse cursor over each one and a tooltip will appear identifying it. The Help menu pulls up some useful help files, but careful — the official user's guide to R is a bit cryptic. An important button for new users of R is the Stop button, highlighted in Figure 3 on page 5 . We will see the usefulness of the stop button in a section further below.

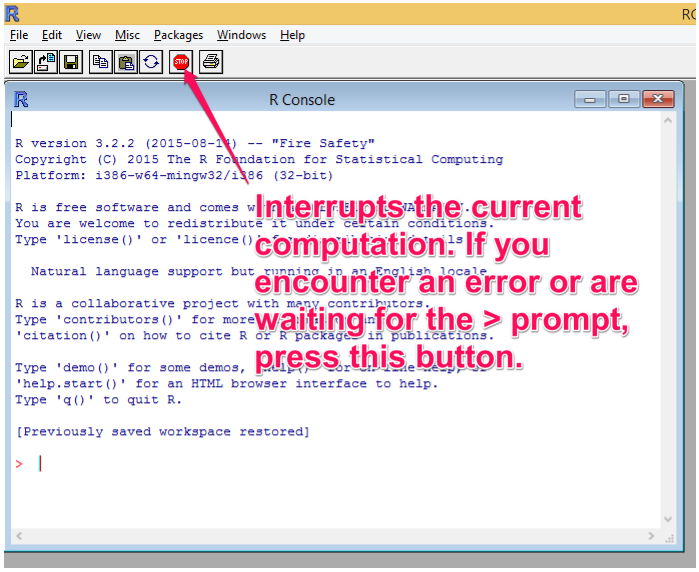

Figure 3: The R console window. Notice the Stop sign icon. The Escape ("Esc") key performs the same function as the Stop button.

Go back to your open version of R. Using the mouse, click on the window that says, 'R console'. Commands can be entered into R via this command line, which always begins with a red > symbol.

This > symbol is an R prompt. When you see it, it is waiting for instructions from you.

The most basic command is to use it as a calculator. Type 2+2 at the prompt and press return. The result appears below. (You can use R as a calculator, with arithmetic operators + (addition), − (subtraction), / (division), or ∗ (multiplication), and `sqrt ( )` (square root). There are many others, but we'll consider those later. For example, `sqrt(4)` yields the result of the square root of 4, 2. Try a few different combinations to see how it works.

3 Getting Started with an Analysis

We will get started with opening a script file, running lines from it to import data, and doing a brief analysis of the data.

3.1 An important preliminary: A working directory to store your R analysis files

R imports and exports files in a working directory. The working directory is a file folder, either located on a local computer hard drive or somewhere on a network drive. Pick a location to store your R files, at least for your first analysis example with R. Will it be on your personal drive space at GVSU? A thumb drive? Pick a safe location, which you can easily remember, and use it throughout the semester. I suggest doing the following. First, create a folder on your GVSU network drive labeled "R analysis files". Then inside the folder, create a subfolder for each assignment or activity, such as "My first R assignment".

3.2 Functions for file directories

The command `sqrt(4)` points toward an important lesson. Commands in R are functions, which means that the name of the function such as 'sqrt' tells R what to do. Usually, R needs an *object* contained in the parentheses `()` to know on what the function should be applied. So the `sqrt ( )` function was applied to the object 4, in the command `sqrt(4)`.

At the prompt, type `getwd()` exactly as it appears and you should then see a line of output that reports the location of your working directory. This `getwd()` command is stands for "get working directory". It includes a `()` parentheses set, indicating it, too, is a *function*.

And again, because R commands are entered as functions, most of the time, you will need to fill in the `()` parentheses set with information for R to interpret. But not for `getwd()`. You can see the contents of your working directory with the `dir()` command, which stands for "list directory", meaning list the contents of the directory. When you type in `dir()`, you should see the names of any files you have stored in this folder. Look at the commands listed in Table 1 on page 7. If you haven't already, try entering `getwd()`, `dir()` and `ls()` on your own at the command prompt. Observe the results in the Console window.

Next, go to the the following page, `http://faculty.gvsu.edu/kilburnw/Rscriptfile.txt`, and download the text file to the folder you have identified as your working directory, to store your R analysis files.

This file, 'Rscriptfile.txt' is a text file containing R commands. Typically R expects script files to have the file extension '.R', but this one is '.txt' to make clear that any text file editor can create or read an R script file.

Table 1: Three basic file folder management commands.

Command	Description
`getwd()`	# lists the working directory
`dir()`	# lists all of the contents of the working directory. Useful to make sure you have all of your homework files in one place.
`setwd('file path' )`	# A function to set the working directory. Replace *file path* with your working directory file path. # A less error prone alternative is to use the drop-down menus to change the working directory.
`ls()`	# Lists the R objects in your workspace. Useful for checking if you have successfully loaded a dataset.

3.2.1 Setting the default working directory

We'll want to tell R where you are storing your R analysis files. We will select this folder as the "working directory", the place where R will check for any datasets to import, and where it will store output.

On a Windows machine From the top menu, go to "File" then "Change dir...", and select your directory where your work will be located. Press "OK".

On a Mac machine From the "Misc" menu, choose the "Change Working Directory..." from the drop-down menu, choose the folder, then select "Open".

When you give R commands to import or export anything, everything imported or exported defaults to this working directory location. Specifying the working directory is important; many frustrating errors can be prevented by remembering to do this step each time you open R.

Verify that the directory R has identified as the working directory is located where you think it is. At the prompt, type `getwd()` and `dir()`.

3.3 Script files

Next, we will open a script file. In the world of professional data analysis, people typically use script or syntax files to record their commands instead of the sole use of drop-down, push-button menus. Why is this? It's because you will need to save your work. Saving your data analysis commands helps you to recover your work later. And usually, you will either need to replicate (reproduce) your work at at later date, improve it, or share it with your colleagues. A script file in R becomes a permanent record of your work.

Script files in R are ordinary text files with commands written in the R language. You can write a script file from within R or you can use a text editor. Script files can have any file name, and while the Windows and Mac operating systems will associate a ".R" file name extension with the R software, using ".R" as a file name extension is not necessary.

3.3.1 Opening a script file.

To open a script file, click on the Console window. You should see the "File" menu at the top left. Open "File" then "open script..." and select the "Rscriptfile.txt" script file you saved to your working

directory. You may have to select the file type of "all files" in the drop-down menu of the Open file window. It should look something like Figure .

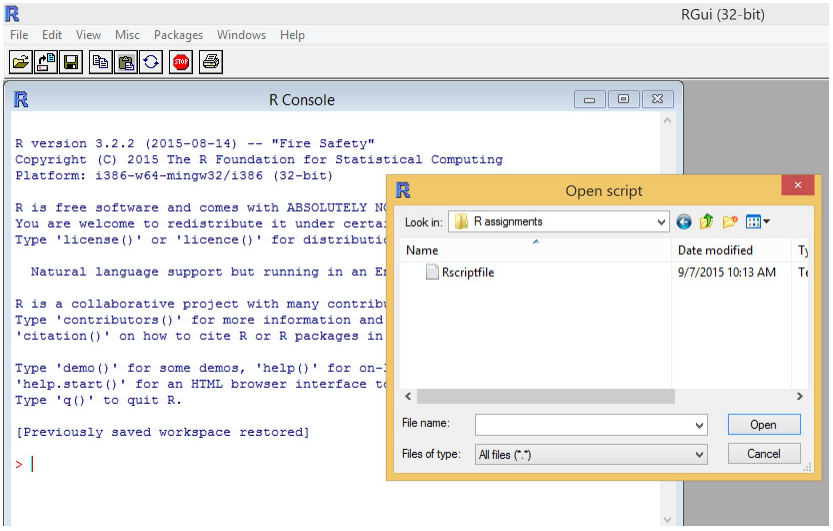

Figure 4: To open a script file, go to File – Open Script.

Once opened, it should appear in your R window. Arrange the Console window and Script file so you can see both, like Figure 5.

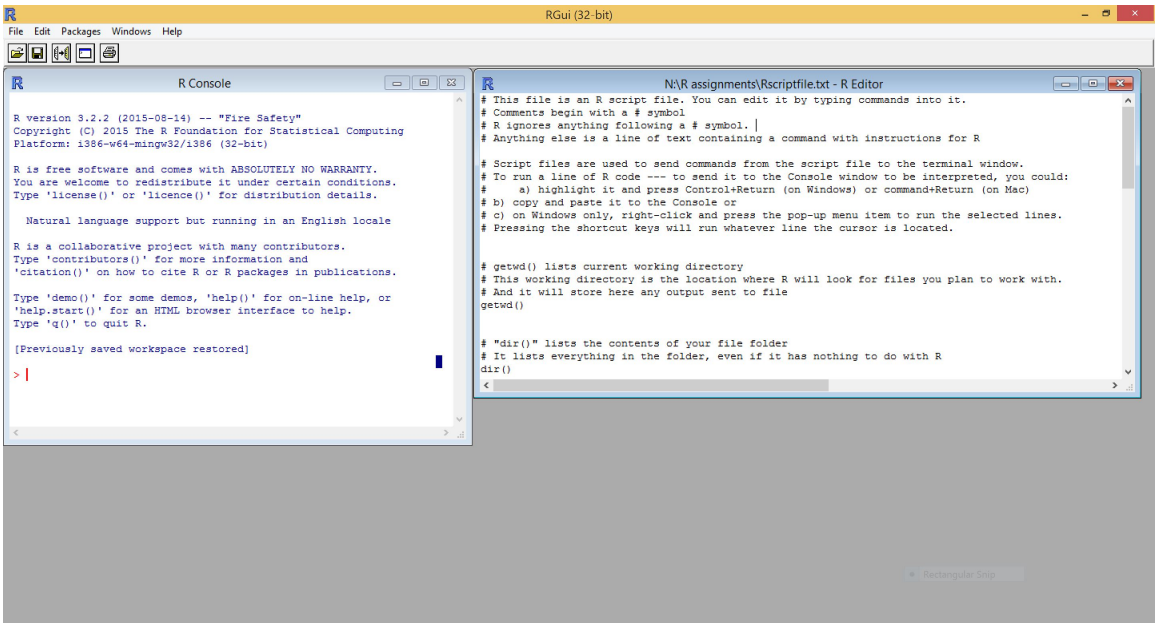

Figure 5: Arrange your Console window and script file so you can see both at the same time.

A standard practice is to write R commands in the script file and then "run" the commands. This is easy to do by highlighting the commands in the script file, and then right clicking with your mouse, and choosing "Run line or selection". –This right click method works on Windows machines. Or you

can highlight lines with the mouse, and use shortcut keys Control + return. On a Mac, highlight the text and then while the text is highlighted, press the Command + return keys at the same time. (The Command key is the one next to the spacebar, with the Apple logo.)

3.3.2 Loading data into R

You will encounter data in at least two different formats — as an external file, which has to be imported into R; or you may encounter data already imported, residing as a dataset within an "R workspace". Until you have imported a dataset into an R workspace, you will not be able to analyze it.

In the introductory R script file, `http://faculty.gvsu.edu/kilburnw/Rscriptfile.txt`, the file provides a dataset that is already imported into an R workspace. Using this line in the script file:

```
load(file=url("http://faculty.gvsu.edu/kilburnw/pollingexcerpt.RData"))
```

3.4 Analyzing data

Any analysis is performed by sending commands to the R Console window, either typing the commands directly into the Console window or via a script file. Run the lines in the script file to observe what happens, and read the comments explaining what each function is intended to do.

Keep in mind these tips:

1. Script file comment lines begin with a hashtag, #. Anything to the right of a hashtag is ignored by the R interpreter.
2. Lines should be run in the order in which they appear in the script file, from top to bottom.
3. Highlight as many or as few lines as you wish to run. You can highlight the whole script file, right click and run.
4. If you encounter an error in which R appears to be waiting for more code from you (a plus sign '+' appears in place of a >), press the Stop button.

After working your way to the end of the script file, you should see something like Figure 6 on page 10.

3.5 Saving your work: data, commands, a history file, and a workspace

3.5.1 Saving data in a workspace

After you have imported data, or if you have made any changes to a dataset, you'll want to save those changes to a workspace. Saving your workspace is how you save your data within R. Click on the Console window, go to the File menu and select "Save Workspace...". In another R session, you open this workspace with the "Load Workspace..." command.

3.5.2 Saving Console window output

To save everything that has scrolled past on the Console window, click on the Console window. Go to the File menu, and then select "Save to File...". The result is a text file.

```
> # reports the mean feeling toward feminists
> tapply(poll$feministsft, poll$gender, mean, na.rm=TRUE)
    Male   Female
52.70722 59.56906
>
> # a histogram
> hist(poll$feministsft)
>
> # with labels and adjusted y scaling
> hist(poll$feministsft, ylim=c(0, 300), xlab="Feelings toward 'Feminists'",
ylab="Response Frequency", main="feelings toward feminists, 2004 ANES")
>
>
```

Quartz 2 [*]

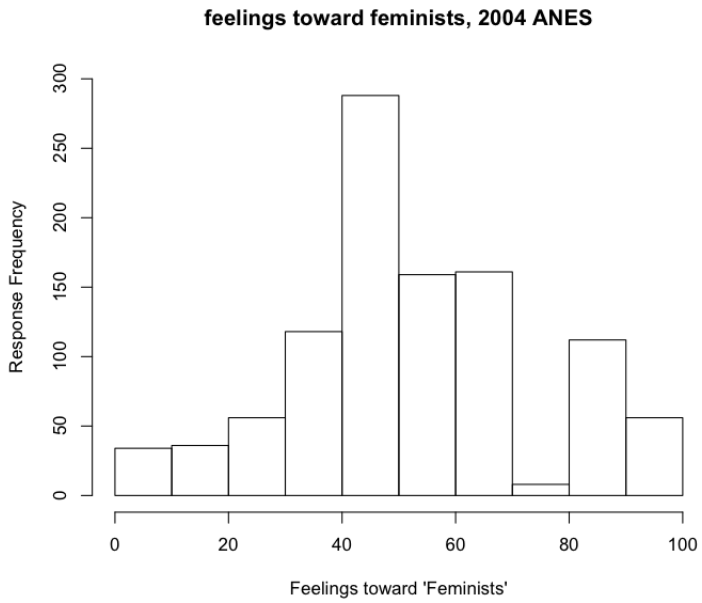

Figure 6: The output result from the last few lines of the script file. On a Mac, the key combination command + shift + 4 was used to capture part of the screen.

3.5.3 Saving a history of commands

A backup, in case you forgot to save your commands in the script file, is to save a list of all the commands you entered into the R console window. It's better to save working commands in the script file, but worst case scenario, after you have completed your work (or need to quit and come back to it), click on the R Console, then choose "File", then "save history..." and you will save a text file containing

3.5.4 Saving graphics

You may find the screen shot function helpful. You could use the "Snipping tool" in Windows or use the command key + Shift + 4 on a Mac to save screenshots. To save a graphics file in an image format, such as .PNG or .JPG, click on a graphic. Go to the File menu, choose "Save as" and save the image in your preferred format.

3.5.5 Remember to always save changes to your script file.

Be sure to save your script file each time you make changes to it. Treat it like a Word document when writing an essay — press save after meaningful changes are made to it.

4 Appendix: notes on working on a Mac

There are a few tweaks to using R on a Mac. Most noticeably, the drop-down menu items from the Console window are different:

1. To open a script file, to go File then "Open Document..."
2. Notice the Workspace menu for saving and opening a workspace.
3. To change working directories, go to the Misc menu and select "Change Working Directory...".
4. While there are methods to send graphics, such as a histogram, to an external file, before you learn how to do that you'll need to use command + shift + 4 to grab part of the screen.
5. use command + return to send script file code to the Console window.

Otherwise, you won't notice much of any difference. There's plenty of Mac centric R advice on the Internet.

5 Where to go for help

The help menu in R links to some helpful documents. There are many useful guides to R floating around the Internet, from videos to PDF guidebooks. The GVSU library holds many e-books about data analysis in R. You should get use to using Google – asking how to do something or googling an error message. Finally, please contact the Data Inquiry Lab, at dil@gvsu.edu. We can help and direct you toward other resources. Check our website, `http://www.gvsu.edu/datainquirylab` for R workshops and handouts.