R Tutorial #1 - Introduction to R and RStudio [COM520]

Benjamin Mako Hill

January 2, 2021

Contents

Very brief introduction to R (and R Studio and R Markdown)

If this is your first foray into R (or just your latest attempt), welcome! I hope this introduction helps you get started.

This file accompanies a screencast. The idea is that you can watch the screencast with RStudio and the file open on your own machine. You can read the compiled file as an HTML or PDF document and interact with it more directly by opening the accompanying .Rmd (RMarkdown) file in R Studio.

We'll begin with a few basics. What is R? What is R Studio? What is the R Console? What is R Markdown? After that (in the second screencast), we'll move on to performing some basic operations that you'll need to understand to actually use R to perform statistical programming.

What is R?

R is a free software environment and programming language for statistical computing. At it's core, it's a very flexible system that you can use to conduct any kind of statistical computing you (or anyone else) can imagine. People may say "R" to refer to the language and the software environment interchangeably. You can read more about R on the R project home page.

What is R Studio?

R Studio is an “integrated development environment” (IDE) that you can use with R. In other words, it’s an application built to make it relatively easy to conduct statistical analysis, manage datasets, generate plots, and generally interact with R in a whole variety of ways.

R Studio has a bunch of options that you can use to adjust the look, feel, and organization of the interface. You can find them under the ‘Tools’ menu and ‘Global Options’.

R Studio also has a number of very, very helpful keyboard shortcuts. Personally, I love keyboard shortcuts and I find that they vastly improve my experience using R Studio. If you want to learn the keyboard shortcuts, print out a copy of a cheatsheet like this and make sure it’s handy any time you even think about using R Studio. You’ll improve quickly.

The most important keyboard shortcut when you’re using R Studio is probably ‘CTRL-Enter.’ It lets you send a command from the scripting window to the R console.

What is the R Console?

If you’re have opened RStudio, you should see a window labeled “Console.” This window allows you to enter direct commands to R. You can type these commands after the little sideways angle-bracket symbol (‘>’) or send them to the console from a script or notebook file. I will demonstrate how I do both of these things in the screencast.

The important thing to know about the R Console is that when you type anything (a ‘statement’) after the sideways angle-bracket and press ‘Enter’ R will try to evaluate the statement and “do” whatever it says. If the console cannot evaluate the statement successfully, it will generate an error.

What is RMarkdown?

RMarkdown is a markup language that you can use to create documents that combine snippets of text and (evaluated) R code. This file was created as an RMarkdown document. You can use RMarkdown to compile documents in many formats, including HTML, PDF, or MS Word documents. For more details on using RMarkdown see http://rmarkdown.rstudio.com where you can also access a series of dedicated tutorials.

You can think of RMarkdown files as “notebooks” where you can write, execute, and compile a combination of text and R code that can then be “knitted” together. Indeed, this is exactly what RStudio does when you click the **Knit** button above an open RMarkdown (.Rmd) file. Clicking **Knit** (or executing the corresponding keyboard shortcut) will produce a document that includes both the text content as well as the output of any embedded R code “chunks” within the document. An embedded R code chunk looks like this:

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

That chunk calls a built-in function ‘summary’ to provide information about a built-in dataset called ‘cars’. R has many built-in functions and a few built-in datasets. We’ll come back to them later. For now, the point is that you can see how RMarkdown integrates a snippet of code and the results of executing that code right into the flow of the text.

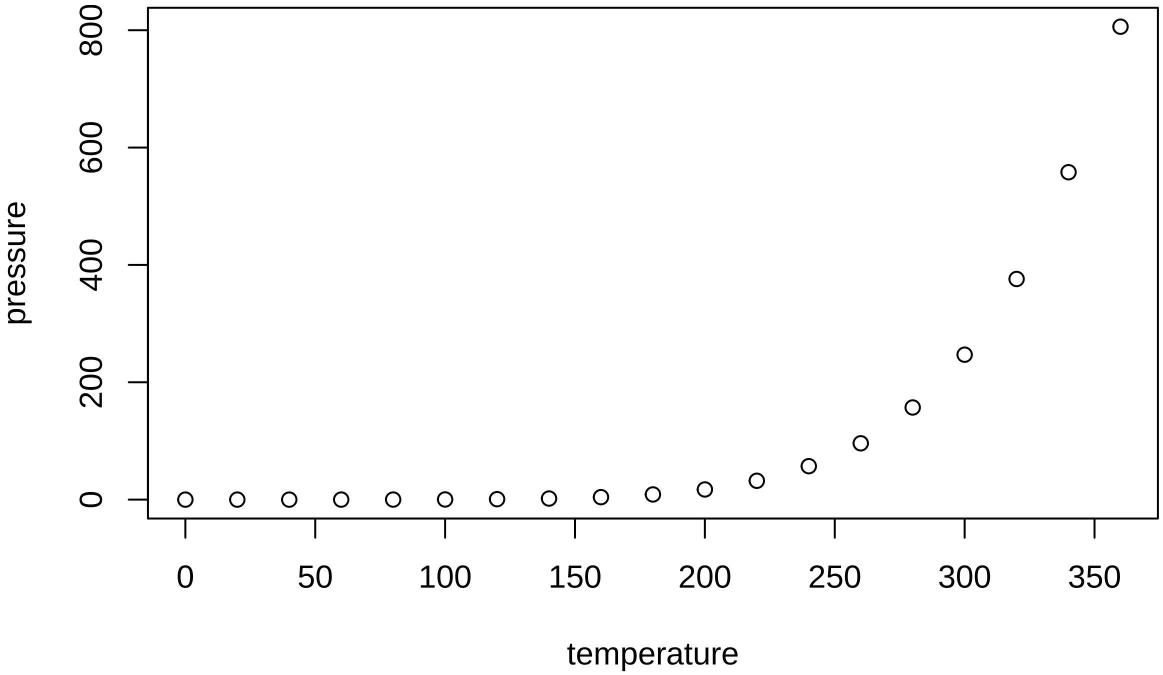
Embedding a new code chunk is easy. There is an 'Insert chunk' option in the 'Code' menu as well as an 'Insert' dropdown at the top of the .Rmd window. You can also use the keyboard shortcut (recommended! It's CTRL-ALT-i for Windows/Linux/Chrome users).

RMarkdown also lets you format text in a variety of ways including *italics* and **bold**.[1]

RMarkdown is not required for this course, but I strongly recommend that you invest the effort to learn to work in something like RMarkdown (other alternatives include Jupyter Notebooks or LaTeX) to complete your problem sets. The results will be clean, easy-to-read files that can integrate R code, analysis output, and graphics.

Including Plots in R Markdown

You can also embed plots, for example here's a plot of another one of R's built-in datasets containing data on the relation between temperature in degrees Celsius and vapor pressure of mercury in millimeters (of mercury).:



Note that the R code that generated the plot didn't show up this time. That's because I included an `echo = FALSE` parameter at the beginning of the code chunk (open up the .Rmd file to see what this looks like and try changing it the `echo = TRUE`). It is sometimes helpful to run code chunks without printing them.

Some other basics of R Studio

R Studio is being very actively developed and has many features that I don't know much/anything about. You can learn a **lot** more on the R Studio site and online (more on that in a moment). For now, I want to make sure you know how to do a few other things that will make it possible to complete your assignments for my class.

Setting preferences and options

The appearance and some features of R studio can be customized "globally" (across all projects) through the 'Global options' item in the 'Tools' menu. For example, some people prefer a darker editor theme.

[1] It even does footnotes!

Working with projects

An R Studio ‘project’ is a bundle of data, code, figures, output, and more that you want to keep bundled together. A project might contain multiple data files or notebooks. It also might contain other material such as a README file (documentation), supplementary materials, or even a finished paper. For the purposes of class, you may want to treat each problem set or assignment as an R Studio project.

R Studio projects are saved as ‘.Rproj’ files accompanied by whatever else the project may entail. You can open them with R Studio and/or create new ones from the ‘File’ menu (select ‘New Project’). Note that R studio can have multiple scripts and/or R Markdown files open, but seems to only be able to have one project open at a time.

Creating and saving a new R Markdown script

Creating new R Markdown scripts is also very straightforward in R Studio. From the ‘File’ menu, select ‘New File’ and ‘R Markdown’. This will let you define some key attributes of the new file and automatically populate the .Rmd with some basic information.

Getting help

There are many, many ways to get help figuring out how to do things in R, R Studio, and R Markdown. I’ll talk more about getting help with R functions in the second Screencast, but for now you should make sure you also have some idea of where/how to look things up when you have questions about R Studio or R Markdown. For example, try out the ‘Help’ menu items and identify some of the cheatsheets (like the one I mentioned earlier) that you think you might want to have around while you’re learning to use these tools. The R Studio website links to several other resources and tutorials that you might find useful. StackOverflow also has extensive Q&A activity for questions about R, R Studio, Markdown, and related topics.