R Module 1

Alex Fout*

2020-07-08

Contents

1 Welcome!

Hi, and welcome to the R Module 1 (AKA STAT 158) course at Colorado State University!

This course is the first of three 1 credit courses intended to introduce the R programming language to those with little or no programming experience.

Through these Modules (courses), we'll explore how R can be used to do the following:

1. Perform basic computations and logic, just like any other programming language
2. Load, clean, analyze, and visualise data
3. Run scripts
4. Create reproducible reports so you can explain your work in a narrative form

In addition, you'll also be exposed to some aspects of the broader R community, including:

1. R as free, open source software
2. The RStudio free software
3. Publicly available packages which extend the capability of R
4. Events and community groups which advocate for the use of R and the support of R users

More detail will be provided in the Course Topics laid out in the next chapter.

1.0.1 How To Navigate This Book

To move quickly to different portions of the book, click on the appropriate chapter or section in the the table of contents on the left. The buttons at the top of the page allow you to show/hide the table of contents, search the book, change font settings, download a pdf or ebook copy of this book, or get hints on various sections of the book. The faint left and right arrows at the sides of each page (or bottom of the page if it's narrow enough) allow you to step to the next/previous section. Here's what they look like:

*Department of Statistics, Colorado State University, fout@colostate.edu

Figure 1: Left and right navigation arrows

1.1 Associated CSU Course

This bookdown book is intended to accompany the associated course at Colorado State University, but the curriculum is free for anyone to access and use. If you're reading the PDF or EPUB version of this book, you can find the "live" version at https://csu-r.github.io/Module1/, and all of the source files for this book can be found at https://github.com/CSU-R/Module1.

If you're not taking the CSU course, you will periodically encounter instructions and references which are not relevant to you. For example, we will make reference to the Canvas website, which only CSU students enrolled in the course have access to.

2 Installing R

In the previous chapter, you ran R code on a website. The purpose of this chapter is to install R on your own computer, so that you can run R without needing access to the internet.

> **Feedback** Any feedback for this section? Click here

2.1 Computer Basics

If you're new to computers, this section will be important for you to get set up. We'll briefly introduce some computer concepts and discuss how they're relevant to R. If you understand the basics of operating systems, directory structures on your computer, and downloading/installing files, then you can probably skim this section, but be sure to pay attention to the R-specific information.

2.1.1 Operating Systems

An operating system is a set of programs that allow you to interact with the computer, and the most popular operating systems are Windows, Mac OS X, and Linux. R works on Windows, Mac OS X, and several Linux-based operating systems, so if you have one of these operating systems, you'll be able install and use R. At least, this is mostly true:

> **Caution** Some versions of Windows that run on ARM processors cannot install R, and installing R on a Chromebook will likely be more complicated (see here). If you're in this situation, contact the instructor immediately.

R isn't designed to work on tablets or phones which run mobile/tablet operating systems (like iOS, iPadOS, Android, ChromeOS), so these are not an option for R.

2.1.2 Files & Directory Structures

A file is a collection of data stored on your computer's hard drive. Examples of files include:

- A music file
- A video
- A slide presentation

- A text document

Different types of files are often treated differently by your computer. For example, a music file is played with a music player program, a video can be viewed with a video player, and a slide presentation might be viewed with Powerpoint. Most operating systems know the type of a file by looking at the *extension*, which is at the very end of the file's name. Examples include ".mp3", ".doc", ".txt", and ".ppt". When using R, we can write scripts which contain R code, and *RMarkdown* documents, which include human readable text and code. R scripts usually have either a ".R" or ".r" extension, and we'll also be using *RMarkdown*, which use either a ".Rmd" or ".rmd" extension.

A *directory*, or *folder*, is a collection of files, and computers use directories to logically organize sets of files. When working with R, you may have to organize several different types of files, including R code, data files, and images. It will be important to stay organized when using R, and we will address this more later in the chapter.

With the increasing prevalence of the internet in everyday life, it's becoming less common for files to exist on your computer. When writing R code, you'll be working with files on your computer, not accessing them over the internet.

2.1.3 Downloads and Installations

To install R, you'll have to download a file from the internet which performs the installation. After you install R, you shouldn't have to download anything to run R. The specific steps to install R will be different depending on your operating system, and this will be addressed in the next section.

Feedback Any feedback for this section? Click here

2.2 Install R & R Studio

Here's where you install R on your personal computer, but you'll actually be installing *two* separate programs. The *first* is the R programming language. The *second* is a separate program called R Studio, which will be the primary way in which you interact with R in this class, we will say more about this later.

2.2.1 Installing R

Installation will look slightly different depending on the operating system, but the major steps are the same.

- First, navigate to the CRAN Mirrors Site, which lists several locations from which R can be downloaded.
- Find a location near you (or not, this isn't critical) and click on the link to be brought to the mirror site.

From this point, this will change depending on your operating system.

2.2.1.1 Windows

- Click "Download R for Windows", then click "base".
- Finally, Click "Download R X.Y.Z for Windows", where X, Y, and Z will be numbers. These numbers indicate which version of R you'll be installing. As of the publishing of this book, R is on version `r_version`.
- Your computer might prompt for the location on your computer that you would like to save the file. Select a location (reasonable options are your `Downloads` folder or the `Desktop`) and select "save".
- When the download completes, find the downloaded file in the File Explorer and double click to run it. This will start the installation process.
- Follow the on screen prompts. For the most part you can click "next" and "install" as appropriate, and you don't have to worry about changing any installation settings.
- Click "Finish" to complete the installation!

Video This video shows the installation process for Windows

2.2.1.2 Mac OS X

- Click “Download R for (Mac) OS X”
- Click “R-X.Y.Z.pkg”, where X, Y, and Z will be numbers. These numbers indicate which version of R you’ll be installing. As of the publishing of this book, R is on version `r_version`.
- Your computer might prompt for the location on your computer that you would like to save the file. Select a location and select “save”.
- When the download completes, find the downloaded file in the Finder and double click to run it. This will start the installation process.
- Follow the on screen prompts. For the most part you can click “continue”, “agree”, “install”, as appropriate, and you don’t have to worry about changing any installation settings.
- Click “Close” to complete the installation!

2.2.1.3 Linux We will not provide details on installing R for Linux, because the process varies depending on your distribution, and because if you’re using Linux, chances are you’re more computer proficient than the average user. Suffice it to say, The first step is:

- Click “Download R for Linux”

And you can probably figure things out from there.

2.2.1.4 Conclusion You should now have R installed! Technically speaking, nothing further is required to work with R. You can open the RGui, and start coding immediately. However, for this course we will be using RStudio, which is a very popular program with an incredibly rich set of features, which will enhance your R programming experience.

2.2.2 Installing RStudio

- Navigate to the RStudio Download Page, and find the download file that matches your operating system.
- Click the link to download the installer, which starts with “RStudio-” or “rstudio-”.
- Your computer might prompt for the location on your computer that you would like to save the file. Select a location (reasonable options are your `Downloads` folder or the `Desktop`) and select “save”.
- When the download completes, find the downloaded file and double click to run it. This will start the installation process.

From this point, this will change depending on your operating system.

2.2.2.1 Windows

- Follow the on screen prompts. For the most part you can click “next” and “install” as appropriate, and you don’t have to worry about changing any installation settings.
- You should now be able to open the start menu, open the RStudio folder, and click on the RStudio icon to open RStudio

Video This video shows the installation process for Windows

2.2.2.2 Mac OS X

- In the window which opens, drag the RStudio icon into the “Applications” folder. You may need to enter your password (click the “Authenticate” button) in order to do so.

- You should now be able to navigate to the Applications folder in Finder, and click on the RStudio icon to open RStudio.

2.2.2.3 Conclusion

Bonus Rstudio also offers a cloud service that allows you to work with R in your browser. We'll use the desktop version but you can check out the interactive primers on the cloud site.

Feedback Any feedback for this section? Click here

2.3 Successfull Installation

When you successfully install R and RStudio, you should now be able to program in R! Before moving further, you should become acquainted with the different parts of RStudio. To do so, watch the video below:

Video This video gives an introduction to some of the main pieces of RStudio

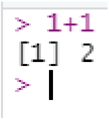
Feedback Any feedback for this section? Click here

2.4 Running Code in RStudio

Now that you're somewhat familiar with RStudio, let's run the same code as we ran on the website, but this time let's run it in R.

2.4.1 The R Console:

In the *R console*, type `1+1` and press enter. The output in the console should look like the following:



```
> 1+1
[1] 2
> 
```

Figure 2: code in the console

Notice that the output `2` is displayed, and the cursor is on a blank line, waiting for more input. This is how coding in the console works.

2.4.2 R scripts

Now let's run the same code, but in an R script. If you haven't already, create a new R script by clicking on the `New File` icon, then selecting `R Script` like so:

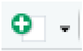


Figure 3: code in the console

In the script window which opens, type `1+1` and press enter. Notice how now, the code did *not* run? In a script, you are free to write R code on several lines before you run it. You can even save the script and load it later in order to run the code it contains. There are multiple ways to run R code in a script. To run a single line of code, do one of the following:

- Place the cursor on the desired line, hold the key, and press enter. On Mac OS X, hold key and press enter instead
- Place the curse on the desired line and click the `Run` button that looks like this:

Figure 4: code in the console

To run multiple lines of code, do one of the following: - Highlight all the code you'd like to run, hold the key, and press enter. On Mac OS X, hold the key and press enter instead. - Highlight all the code you'd like to run, and click the `Run` button.

Run the `1+1` code using one of the methods above, and observe the output. Notice how the output is *still* in the console window, even you ran the code in a script!

> **Caution** Even though running R code from the console and an R script are done differently, they should produce the same results. Both are running R!

Now that you've run some code in the console and from an R script, let's try some of the other code we wrote previously.

2.4.3 Same Examples, On Your Computer!

In the *console*, type the command `factorial(10)`. Did you get the same result as you got on the website? Now type the following two lines in an R script and run them:

```
x <- -10:10
plot(x, x^2)
```

This code produces a plot, which should show up in the lower right corner in the "Plots" window. Finally, *copy* the following code, paste it into your script, and run it:

```
library(ggplot2)
theme_set(theme_bw())
ggplot(mtcars, aes(y=mpg, fill=as.factor(cyl))) +
  geom_boxplot() +
  labs(title="Engine Fuel Efficiency vs. Number of Cylinders", y="MPG", fill="Cylinders") +
  theme(legend.position="bottom",
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank())
```

You're now running R code on your computer!

> **Reflect** Imagine you are practicing programming in R and your classmate tells you they heard about an interesting new R command which they'd like you to try out. Would you run the command in an R script, or the R console? How might your answer change if you wanted to keep a record of all the interesting R command you found?

2.4.4 R Markdown

You've seen how to run R code in the R console, and from an R script, but there's one more way to run R that we need to talk about: R Markdown

R scripts are convenient because they can store multiple R commands in one file. R Markdown takes this idea further and stores code alongside human readable text. There is much that could be said about R Markdown, but for now, we'll just stick with the basics.

To start, watch this video:

Video This Video gives a basic introduction to RMarkdown.

As the video stated, there are three types of sections to an RMarkdown document: - Header - Human readable text - Code Chunks

There's only one header, but there can be many blocks of human readable text and many code chunks.

Bonus See here for more things you can do with RMarkdown.

Caution As part of this class, you'll be filling in an R Markdown document as you complete the progress checks in the book (except for the first progress check box, which you completed already) On Canvas, download the `progress_check_2.rmd` file and follow the instructions. Here is the first code chunk you will include in the document!

Video TODO: Add video link. This video should help get you started with the Progress Check Assignments!

Feedback Any feedback for this section? Click here

2.5 Workspace setup

Whenever you are programming in R, and especially for this class, it's important to stay organized. This section will give you some instructions and tips for how to organize material for this R course

2.5.1 Recommended Settings

First of all, let's set some settings in RStudio. At the top of the R window, click `Tools`, then `Global Options`, and do the following:

1. On the left side of the window that pops up, and make sure it's on the "General" tab
2. Find the "Workspace" section on the right, make the following changes: – *uncheck* "Restore .RData into workspace on startup" – Change the "Save workspace to .RData on exit" option to *never*
3. On the left side, select the "Appearance" tab.
4. (Optional) Change the Zoom setting to increase or decrease the size, to fit your screen best.
5. (Optional) Change the "Editor theme:" setting to find a color scheme that looks good to you.
6. Click "Apply", then "OK" at the bottom of the window.

Step 2 ensures that each time you open RStudio, there's no "memory" of anything you may have been doing in R previously. This is a good option for R beginners to avoid confusion and mistakes. Step 4 can also be done using the shortcuts `<control> <shift> +` (to increase size) and `<control> -` (to decrease size). On Mac OS X, the commands are `<command> <shift> +` and `<command> -`.

2.5.2 Setting working directory

Every time R runs, it has a *working directory*, which is the folder where R “looks” when loading and saving files. In RStudio, the `Files` window contains the “More” menu, which has options to *set as working directory* or *go to working directory*. This will become more relevant when you start loading data and saving results later in the course. For this course, you’ll be using an RStudio project, which automatically sets the working directory.

> ☀ **Bonus** See here for more information about working directories.

2.5.3 Create RStudio Project and directories for class

RStudio also has a feature called *projects*, which

Note: This will automatically set your working directory

Folder structure:‘

- raw_data
- assignments
-

> **Feedback** Any feedback for this section? Click here

2.5.4 Best Practices

When you’re finished coding in R,

How to clear environment How to clear console

2.6 Reflection

> **Reflect** Before moving on to the next section, take a note of all you’ve done so far.
>
> 1. Did your R installation go smoothly? If not, could you troubleshoot the errors or find help online?
> 2. Does using R remind you of other programs you have experience with?
> 3. What could be some reasons that using R code written by someone else might not work on your computer?