R Module 1

Alex Fout[*]

2020-07-06

Contents

1 Welcome!

Hi, and welcome to the R Module 1 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 Course Preliminaries

This course is presented as a bookdown document, and is divided into chapters and sections Each week, you'll be expected to read through the chapter and complete any associated exercises, quizzes, or assignments.

2.0.1 Special Boxes

Throughout the book, you'll encounter special boxes, each with a special meaning. Here is an example of each type of box:

Reflect This box will prompt you to pause and reflect on your experience and/or learning. No feedback will be given, but this may be graded on completion.

Assessment This box will signify a quiz or assignment which you will turn in for grading, on which the instructor will provide feedback.

Progress Check This box is for checking your understanding, to make sure you are ready for what follows.

Video This box is for displaying/linking to videos in order to help illustrate or communicate concepts.

Caution This box will warn you of possible problems or pitfalls you may encounter!

Bonus This box is to provide material going beyond the main course content, or material which will be revisited later in more depth.

Feedback This box will prompt for your feedback on the organization of the course, so we can improve the material for everyone!
Any of the boxes may include hyperlinks like this: I am a link or code like this `This is code`.

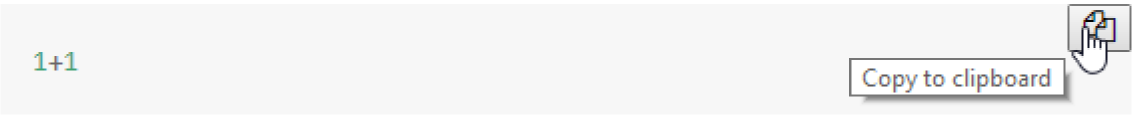
2.0.2 How This Book Displays Code

In addition, you may see R code either as part of a sentence like this: `1+1`, or as a separate block like so:

```r
1+1
```

```
[1] 2
```

Sometimes (as in this example) we will also show the **output** (in yellow), that is, the result of running the R code. In this case the code `1+1` produced the output `2`. If you hover over a code block with your mouse, you will see the option to copy the code to your clipboard, like this:



```
1+1
```

Figure 2: copying code from this book

This will be useful when you are asked to run code on your computer.

2.0.3 Next Steps

When you're ready, go to the next section to learn about the course syllabus and grading policies.

2.1 Course Topics & Syllabus

Broadly speaking, the topics of this course are described by the Chapter Titles. Here's what each entails: - Course Preliminaries: Introduction to R and the world of R - Installing R: Like it sounds, setting up your computer so you can work with R. - R Programming Fundamentals: The basics of programming in R, the building blocks that you need in order to do anything more interesting. - Working with Data: How to do meaningful things with data sets. Probably the most useful Chapter of the book. - Creating R Programs: More programming concepts to increase your R Power!

2.1.1 Syllabus

First, some important details:

- **Instructor**: Alex Fout
- **Office Hours**: TBD on Microsoft Teams
- **Webpages**: Canvas, this textbook
- **Course Credits**: 1. Because this is a summer course, you'll be expected to digest the
- **Textbook**: You're reading it right now. The textbook will be your primary learning resource. You'll be expected to read through the required sections, watch any relevant videos, and complete any reflections, progress checks, and assessments along the way.
- **Prerequisites**: None

- **Progress Checks**: As you work your way through the textbook, you'll encounter purple "Progress Check" boxes. You'll be provided a template RMarkdown Document to fill in as you complete the progress checks. About once per week, you'll compile the document and turn it in for grading via Canvas
- **Homework**: About once per week, you'll complete an assignment using R. Homeworks must be turned in by 11:59pm (Mountain) on the day they are due.
- **Exams**: There will be no exams in this course
- **Quizzes**: Once per week, there will be a 15 minute Canvas quiz. Quizzes must be completed by 11:59pm (Mountain) on the day they are due.
- **Lectures**: Since we aren't having in-person lectures, we will hold short "Virtual Meetings" through MS Teams instead. These will be shorter than a traditional lecture (approximately 10-30 minutes), and the purpose will be to allow some interaction between everyone in the course and to allow the instructor to introduce any relevant topics.
- **Grading**: The grading for the course is apportioned like so:
  - Progress Checks: 30%
  - Homework: 40%
  - Quizzes: 30%

2.1.2 Schedule

Week	Weekday	Date	Reading	Due
1	Monday	July 13	Ch1, Ch2	Progress Checks 1
1	Wednesday	July 15	Ch3	Quiz 1
1	Friday	July 17	Ch4	Assignment 1
2	Monday	July 20	Ch4	Progress Checks 2
2	Wednesday	July 22	Ch4	Quiz 1
2	Friday	July 24	Ch5	Assignment 2
3	Monday	July 27	Ch5	Progress Checks 3
3	Wednesday	July 29	Ch5	Quiz 1
3	Friday	July 31	Ch5	Assignment 3
4	Monday	August 03	Ch6	Progress Checks 4
4	Wednesday	August 05	Ch6	Quiz 4
4	Friday	August 07	Ch6	Assignment 4

2.1.3 Course Policies

- Late Work: Homework and Progress Checks must be turned in on time to receive full credit. You may turn in Homework and Progress Checks up to 2 days late for up to 50% credit.
- Group Work: Students are welcome to discuss the course with each other, but all work you turn in must be your own. This means no sharing solutions to homework, progress checks, or quizzes. You *are* welcome to seek help
- Students with Disabilities: The university is committed to providing support for students with disabilities. If you have an accomodation plan, please provide that to me as soon as possible so we can discuss appropriate arrangements.
- Growth Mindset: This phrase was coined by Carol Dweck to reflect how your learning outcomes can be affected by the way you view the learning process. To quote Dweck: "The view you adopt for yourself profoundly affects the way you lead your life… Believing that your qualities are carved in stone - *the fixed mindset* - creates an urgency to prove yourself over and over. If you have only a certain amount of intelligence, a certain personality, and a certain moral character — well, then you'd better prove that you have a healthy dose of them. It simply wouldn't do to look or feel deficient in these most basic

characteristics… There's another mindset in which these traits are not simply a hand you're dealt and have to live with, always trying to convince yourself and others that you have a royal flush when you're secretly worried it's a pair of tens. In this mindset, the hand you're dealt is just the starting point for development. This growth mindset is based on the belief that your basic qualities are things you can cultivate through your efforts. Although people may differ in every which way — in their initial talents and aptitudes, interests, or temperaments — everyone can change and grow through application and experience." Programming may be a very new, intimidating thing for you. That's okay! View this course as a way to grow and gain new skills which you can use to do incredible and important things!

- Learn by doing: A wise statistics instructor once compared watching someone else solve statistics problems to watching someone else practice shooting basketball free throws. You may learn a little by watching, but at some point you won't get any better until you try it yourself! The same can be said for programming. Reading a textbook and watching videos are a good *start*, but you'll have to actually *program* in order to get any better! This textbook was designed to be *interactive*, and I encourage you to "code along with the book" as you read.

2.1.4 Grading Scale

Grades will be assigned according to the following scale:

Class_Score	Letter_Grade
92%-100%	A
90%-92%	A-
88%-90%	B+
82%-88%	B
80%-82%	B-
78%-80%	C+
70%-78%	C
60%-70%	D
0%-60%	F

2.2 Running your first R Code

Enough of the boring stuff, let's run some R code! Normally you will run R on your computer, but since you may not have R installed yet, let's run some R code using a website first. As you run code, you'll see some of the things R can do. In a browser, navigate to rdrr.io/snippets, where you'll see a box that looks like this:

The box comes with some code entered already, but we want to use our own code instead, so delete all the text, starting with `library(ggplot2)` and ending with `factor(cyl))`. In its place, type `1+1`, then click the big green "Run" button. You should see the `[1] 2` displayed below. So if you give R a math expression, it will evaluate it and give the result. Note: the "correct answer" to $1 + 1$ is 2, but the output also displays `[1]`, which we won't explain until later(TODO), so you can ignore that for now.

Next, delete the code you just wrote and type (or copy/paste) the following, and run it:

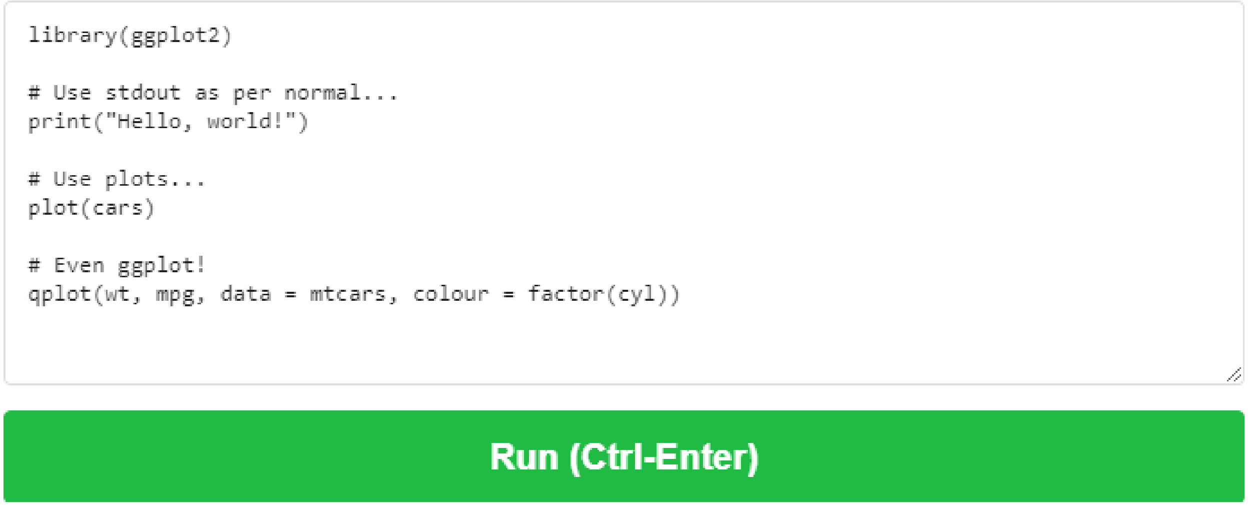
```r
factorial(10)
```

The result should be a very large number, which is equivalent to 10!, that is, $10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$. This is an example of an R *function*, which we will discuss more in Section (TODO: insert ref).

Aside from math, R can produce plots. Try copy/pasting the following code into the website:

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

You should see points in a scatter plot which follow a parabola. Here's a more complicated example, which you should copy/paste into the website and run:



```
library(ggplot2)

# Use stdout as per normal...
print("Hello, world!")

# Use plots...
plot(cars)

# Even ggplot!
qplot(wt, mpg, data = mtcars, colour = factor(cyl))
```

Figure 3: rdrr code entry box

```
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())
```

R can be used to make many types of visualizations, which you will do more of in Section (TODO: insert ref).

> ☀ **Bonus** This may be the first time you've seen R, so it's okay if you don't understand how to read this code. We'll talk more later about what each statement is doing, but for now, here is a brief description of some of the code above:
>
> - `-10:10` This creates a sequence of numbers starting from -10 and ending at 10. That is, $-10, -9, -8, \dots, 8, 9, 10$.
> - `library` This is a function which loads an R *package*. R packages provide extra abilities to R.

2.3 What do you hope to get out of this course?

To close out this chapter, it would be healthy for you to reflect on what you'd like to get from this course. Take some time to think through each question below, and write down your answers. It is fine if your honest answer is *I don't know*. In that case, try to come up with some possible answers that *might* be true.

Reflect

1. Why are you taking this course?
2. If this course is required for your major, how do you think it is supposed to benefit you in your studes?
3. What types of data sets related to your field of study may require data analysis?
4. What skills do you hope to develop in this course, and how might they be applied in your major and career?

TODO: canvas assignment?

Store your answers in a safe place, and refer to them periodically as you progress through the course. You may find that you aren't achieving your goals and that some adjustment to how you are approaching the course may be necessary. Or you may find that your goals have changed, which is fine! Just update your goals so that you have something to refer back to.

2.4 What is R?

What is R? This question can be answered several different ways. Here are a few of them:

2.4.1 R is a Programming Language

A programming language is a way of providing instructions to a computer. Some popular languages (in no particular order) are C, C++, Java, Python, PHP, Visual Basic, and Swift. Much like other types of languages, programming languages combine text and punctuation (syntax) to create statements which provide meaningful instructions (semantics) to be performed by a computer. These instructions are called "code". R code can be used to do many things, but primarily R was designed to easily work with data and produce graphics. The R language can be used to use a computer to do the following: - Read and process a set of data in a file or database - Use data to compute statistics and perform statistical tests - Produce nice looking visualizations of data - Save data for others to use. But this list is just the tip of the iceberg. As you will see, R can be used to do so much more! After the instructions are written, the R code is *run*, that is, the code is provided to the computer, and the computer performs the instructions to produce the desired results.

Bonus Many other programming languages use different syntax for the same purpose.

`#` comments out a line in R and python

`%` comments out a line in matlab

`//` comments out a line in C++ and javascript

Similar to learning a foreign language, learning your first programming language will make it easier to understand other similar ones.

2.4.2 R is software

R can also be thought of as the software program which runs R code. In other words, if R code is the computer language, then the R software is what interprets the language and makes the computer follow the instructions laid out in the code. This is sometimes called "base R".

2.4.3 R is Free

The R software is free, so anyone can download R, write R code, and run the R code in order to produce results on their computer.

2.4.4 R is Open Source

The R software, which runs R code, is also made up of a bunch of code called *source code*. In addition to being free, R is also *open source*, meaning that anyone can look at the source code and understand the "deep-down nuts-and-bolts" of how R works. In addition, anyone is able to *contribute* to R, in order to improve it and add new features to it.

> **Reflect** What are the advantages of open-source software? What are some potential downsides?
> Why do you think the creators of R decided to make it open source?

2.4.5 R is an ecosystem

Another way of thinking about R is to include not only the R language and the R software, but also the community of R users and programmers, and the various "add on" software they have created for R. These add on software are called "packages".

2.4.6 R Packages

An R package is software written to extend the capabilities of base R. R packages are often written in R code, so anyone who knows how to write R code can also create R packages. The importance of packages cannot be understated. One of the reasons for the incredible popularity of R is the fact that members from the community can write new packages which enable R to do more. Sometimes packages are written to help folks in particular disciplines (e.g. psychology, geosciences, microbiology, education) do their jobs better. Other times, packages are written to extend the capability of R so that people from many disciplines can use them. R can be used to make web sites, interactive applications, dynamic reproducible reports, and even textbooks (like this one!).

The inclusion of R packages, combined with the free and open source nature of R software, has led to the development of a active, diverse, and supportive community of R users who can easily share their code, data, and results with one another.

> ☀ **Bonus** skimr provides a frictionless approach to summary statistics which conforms to the principle of least surprise, displaying summary statistics the user can skim quickly to understand their data.

2.4.7 R Interfaces

The R software can be run in many different places, including personal computers, remote servers, and websites (as you have seen!). R works on Windows, MAC OSX, and Linux, and R can be run using a terminal or command line (if you know what those are), or using a graphical user interface (with buttons you can click and such). By far one of the most popular ways of using R is with RStudio, which is *also* open free and open source software. For this course, you'll be using RStudio.

2.5 The R Community

We already mentioned that there is active community of R users around the world, ranging from novice to expert level. Here is a partial list of venues where R users interact (aside from the official websites, none of these links should be considered an official endorsement):

1. R Project: The official website for R
2. R Project Mailing Lists: Various email lists to stay informed on R related activities. The R-announce list is a good starting point, which will keep you updated on the latest releases of the R software

3. Twitter #rstats: Many R Users are active on Twitter and you can find them
4. Tidy Tuesday is a weekly online project that focuses on understanding how to summarize, arrange, and make meaningful charts with open source data. You can see the projects others have done by following #tidytuesday on twitter.
5. R-Ladies is a global group dedicated to promoting gender equality in the R community. They have an elaborate list of resources for learning and host educational and networking events.
6. R-Podcast: A periodic podcast with practical advice for using R, and the latest R news.
7. R-Bloggers: A blog website where authors can post examples of code, data analysis, and visualization.

2.5.1 Places to Get Help (If you're taking this class for credit)

TBD.

2.5.2 Places to Get Help (If you're just looking at this material but aren't taking the class)

If you find yourself stuck, there are many options available to you, here are a few:

1. Stack Overflow is a message board where users can post questions about issues they're having. If you search for your error, there's likely already an answered question about it. If not, you can submit one with a reproducible example that the active community can help you with.
2. R Manuals: With so many R resources available on the internet, sometimes information get's "boiled down" or simplified for ease of communication. If you need the "official answer" to a question, these manuals are the place to go. Check out "An Introduction to R" for a good reference.

3 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.

3.1 Computer Basics

If you're new to computers, this section will be important for you to get set up. If you understand the basics of operating systems, directory structures on your computer, and downloading/installing files, then you can safely skip to the next section.

3.1.1 Operating Systems

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).

3.1.2 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.

TODO: take out?

3.2 Install R & R Studio

You'll actually be installing *two* separate programs, both of which are free to use. The *first* is the R programming language. The *second* is a separate program called R Studio. R Studio will be the primary way in which you interact with R in this class, we will say more about this later.

3.2.1 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.

3.2.1.1 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`.
- any other settings. 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 "continue", "agree", "install" as appropriate, and you don't have to worry about changing any installation settings.

> 🎥 **Video** Create windows R install Video

3.2.1.2 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.

> 🎥 **Video** ?Ben? install from Mac OS X?

> ☀ **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.

3.2.2 Installing R Studio

3.3 Successfull Installation

TODO: add some basic tasks here to make sure R and RStudio work. (screenshots?)

> 🎥 **Video** todo: video orienting to R Studio (Matt video?)