CHESTER ISMAY

GETTING USED TO R, RSTUDIO, AND RMARKDOWN

Contents

1
Introduction

This book was written using the **bookdown** R package from Yihui Xie. You can find different formats for the book by clicking on the save icon ⬇ in the top pane of this book website. HTML is the preferred format but PDF and ePub formats are also available.

This resource is designed to provide new users to R, RStudio, and RMarkdown with the introductory steps needed to begin their own reproducible research. A review of many of the common R errors encountered (and what they mean in laymen's terms) is also provided. Many screenshots and GIFs will be included, but if further clarification is needed on these or any other aspect of the book, please create a GitHub issue here or email me with a reference to the error/area where more guidance is necessary.

This book will evolve and be updated as needed based on feedback given in courses that Chester Ismay teaches or facilitates. Check the date at the beginning of this chapter to see when it was last updated. In addition, each individual chapter shows the time of last update.

2
Why R?

If you are brand new to R and programming, you may be scared. You aren't used to having to type commands to tell the computer what to do. You may be more used to using drop-down menus and other graphical user interfaces that allow you to pick what you'd like to do. So why are so many companies, colleges/universities, and individuals of all disciplinary backgrounds shifting towards using R?

There are lots of answers to this question, but some of the most important for us now are:

1. R is free. RStudio is free.

   One of the biggest perks about working with R and RStudio is that they are both provided free of charge to use. R is an open-source programming language that has grown tremendously in recent years with developers adding more functionality and packages on a daily basis. Where other more proprietary packages are sometimes stuck in the dark ages (the 1990s, for example) of development and can be incredibly expensive to purchase, R continues to be a free alternative that allows users of all levels to contribute.

   RStudio is a graphical user interface that allows one to write R code and view the results of that code in an easy way. It is also free to download and work with.

2. Analyses done using R are reproducible.

   As many scientific fields push towards more reproducible analyses, the point-and-click proprietary systems actually serve as a hindrance to this process. If you need to re-run your analysis using these systems, you'll need to carefully copy-and-paste your analysis and plots into your text editors from potentially beginning to end. Anyone that has done this sort of copy-and-pasting knows that it is prone to errors and incredibly tedious.

   If you use the workflows described in this book, your analyses will be reproducible so you don't need to worry about these copy-and-pasting issues. As you might have guessed by now, it would be much better to be able to update your code/data inputs and re-run all of your analysis than to have to worry about manually moving your results from one program to another. Reproducibility also helps you as a programmer since your greatest collaborator will probably be yourself a few months or years down the road. Instead of having to carefully write down all the steps you took to find the right drop-down menu option, your entire code is stored.

3. Using R makes collaboration easier.

   This also helps you with collaboration since, as you will see later, you can share an RMark-down file containing all of your analysis, documentation, commentary, and the code to others. This reduces the time to needed to work with others and reduces the likelihood of errors being made in following along with point-and-click analyses. The mantra here will be to **Say No to Copy-And-Paste!** both for your sanity and for the sake of science.

4. Finding answers to questions is much simpler.

   If you have ever had an issue with software, you know how difficult it is to find answers to your questions. "How can I describe the process to someone else? Do I need to take screenshots? Do I really need to call IT and wait for hours for someone to respond?" R is a programming language and so it is much easier (after a bit of practice) to use Google or Stack Overflow to find answers to your questions. You'll be amazed at how many other users have encountered the same sorts of errors you will see when you begin.

   I frequently (almost on a daily basis) Google things like "How do I make a side-by-side boxplot in R coloring by a third variable?". You'll become better at working with R by reaching out to others for help and by answering questions that others have. In addition, Chapter 6 describes many common errors and how you can fix them.

5. Struggling through programming helps you learn.

   We all know that learning isn't easy. Do you have trouble remembering how to follow a list of more than 10 steps or so? Do you find yourself going back over and over again because you can't remember what step comes next in the process? This is extremely common especially if you haven't done the procedure in awhile. Learning via following a procedure is easy in the short-term, but can be extremely frustrating to remember in the long-term. Programming (if done well) promotes long-term thinking to short-term fixes.

   One unfortunate thing that we frequently take for granted is that our brain tricks us into picking the easy route. If you truly want to learn how to do something (like programming with R), you'll need to feel frustrated at times. Any time you learn something you've been frustrated. (We tend to forget all the frustration and only think about where we currently are.) R still frustrates me from time to time, but I grow through practice and I look forward to the challenges. Hadley Wickham encapsulated this phenomenon nicely in the Prologue of the book "Hands-On Programming with R" (Grolemund, 2014):

   > As you learn to program, you are going to get frustrated. You are learning a new language, and it will take time to become fluent. But frustration is not just natural, it's actually a positive sign that you should watch for. Frustration is your brain's way of being lazy; it's trying to get you to quit and go do something easy or fun. If you want to get physically fitter, you need to push your body even though it complains. If you want to get better at programming, you'll need to push your brain. Recognize when you get frustrated and see it as a good thing: you're now stretching yourself. Push yourself a little further every day, and you'll soon be a confident programmer.

Last updated:

```
## [1] "Sunday, August 14, 2016 11:22:15 PDT"
```

3
R and RStudio Basics

3.1 What is R?

In Chapter 2, I discussed many of the reasons why you should be doing your analyses (especially those of the data type) using R. If you skipped over that chapter in the hopes of just hopping into learning about R, I request that you to go back to it and carefully read it over. As you begin working with R, it is especially important to review that introductory chapter from time to time.

3.1.1 R beginnings

R was created by a group of statisticians who wanted an open-source alternative to the costly proprietary options. Being created by statisticians (instead of computer scientists) means that R has some quirky aspects to it that take a little bit of time to get used to. We'll see that many packages have been developed to help with this and that you don't need to have advanced degrees in Statistics to be able to work with R now.

Getting back to the development of R... R was created by **R**oss Ihaka and **R**obert Gentleman in New Zealand at the University of Auckland. It is a spin-off of the S programming language and is named partly after the first names of its developers (as you can see in the emphasis above). The beginning ideas for creating R came in 1992 and the first version of R was released in 1994. You can find much more about the background of R and its features as well as its connections to the S language on its Wikipedia page.

3.1.2 R packages

I first learned to use R while a graduate student at Northern Arizona University from Dr. Philip Turk in 2007. At the time, I never thought that R could have exploded in users as we have seen since 2011. I never would have thought that students taking an introductory statistics course would be encouraged to learn to use R.

In 2007, it was still largely esoteric and tricky language used by statisticians to do analyses. Getting used to the syntax for producing plots and working with data was especially tricky for those with little to no programming experience. So what has changed since 2007 about learning R?

I believe one of the biggest developments has been the creation of packages to make R easier to work with for newbies. Packages are created by users of R to increase the functionality of the base R installation. Packages created by Hadley Wickham and others recently have greatly expanded the capabilities of R, while also working to make beginning with R simpler. From the Wikipedia page referenced earlier, as of January 2016, there were around 7800 additional R packages available on common R repositories.[1]

Another great development is the graphical user interface called RStudio and the package developed by the those that work for RStudio, Inc. called `rmarkdown`. We will discuss `rmarkdown` (also referred to as RMarkdown) in a Chapter 4, and will now focus on discussing RStudio.

[1] You'll see how to download these packages via `install.packages("dplyr")` and load them into your current R working environment via `library("dplyr")`, for example, in Chapter 5.

3.2   What is RStudio?

RStudio is a powerful, free, open-source integrated development environment for R. It began development in 2010 and its first beta release came in February 2011. It is available in two editions: RStudio Desktop and RStudio Server. This book will focus mostly on using the RStudio Server, but both versions are nearly identical to work with.

You can find instructions linked below for downloading R and RStudio on Windows and Mac machines. If you are using RStudio Server, your professor and members of your organization's IT department have done these steps for you. On the RStudio Server you log on using a web browser to an account sitting on the cloud. There are many advantages to using the RStudio Server for the beginning user including sharing of R projects to help with feedback and error resolution. Installation of software can also cause its own headaches and this is eliminated by using the RStudio Server.

After you complete a few months of work with the RStudio Server, it is recommended that you download RStudio Desktop to your computer. The instructions to do so are below.

3.2.1   Installing R and RStudio Desktop

A step-by-step guide to installing R and RStudio Desktop with screenshots can be found

- here for the Mac and
- here for a PC.

Unless you plan to create PDF documents (which requires a multiple gigabyte download of LaTeX) you can skip some of the later steps of the installation. It is recommended that you select **HTML** as the Default Output Format for RMarkdown. You'll see more about this in Chapter 4.

3.3   Working in RStudio Server

- RStudio Server
- Screenshots of RStudio frames?

Last updated:

```
## [1] "Sunday, August 14, 2016 11:22:15 PDT"
```

4
R Markdown

- Walk through the components of an R Markdown file
- Resource for Markdown: `https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet`
- RMarkdown chunk options
- Help -> Cheatsheets

5
Introductory R analysis using R Markdown

- A beginning workflow
- "File organization and naming are powerful weapons against chaos." - Jenny Bryan
- Give an introduction into using R with periodic table dataset
- Mean, median, standard deviation, five-number summary, distribution
- Some content to cover:

  - data structures (vectors, lists, data frames, matrices)
  - indexing/subsetting
  - functions (default arguments)
  - Case matters in R!
  - Why do some arguments require quotations and others don't?

- R Markdown templates

6
Deciphering Common R Errors

https://github.com/noamross/zero-dependency-problems/blob/master/misc/stack-overflow-common-r-errors.md
   http://blog.revolutionanalytics.com/2015/03/the-most-common-r-error-messages.html

7
Bibliography

Grolemund, G. (2014). *Hands-On Programming with R.* O'Reilly.