# Reproducible Medical Research with R

Peter D.R. Higgins, MD, PhD, MSc

2020-05-26

# Contents

# Chapter 1

# Preface

Welcome to Reproducible Medical Research with R (RMWR). I hope that this book meets your needs.

## 1.1 Who This Book is For

This is a book for anyone in the medical field interested in analyzing the data available to them to better understand health, disease, or delivery of care. This could include nurses, dieticians, psychologists, and PhDs in related fields, as well as medical students, residents, fellows, or doctors in practice.
I expect that most learners will be using this book in their spare time at night and on weekends, as the medical school curriculum is already packed full, and there is no room to add skills in reproducible research to the standard curriculum. This book is designed for self-teaching, and many hints and solutions will be provided to avoid roadblocks and frustration. Many learners find themselves wanting to develop reproducible research skills after they have finished their training, and after they have become comfortable with their clinical role. This is the time when they identify and want to address problems in their practice with the data they have before them. This book is for you.

## 1.2 Prerequisites

Thank you for giving this e-book a try. This is designed for physicians or others analyzing health data who are interested in pursuing this field using the R language. We will assume that:

- you have access to a computer

- that you have access to the internet
- that you can download the current version of R, and
- that you have downloaded a current version of Rstudio.

## 1.3 The Spiral of Success Structure

This book is structured on the concept of a "spiral of success", with readers learning about topics like data visualization, data wrangling, data modeling, reproducible research, and communication of results in repeated passes. These will initially be at a superficial level, and at each pass of the spiral, will provide increasing depth and complexity. This means that the chapters on data wrangling will not all be together, nor the chapters on data visualization. Our goal is to build skills gradually, and return to (and remind students of) their previously built skills in one area and to add to them. The eventual goal is for learners to be able to produce, document, and communicate reproducible research to their community.

## 1.4 Motivation for this Book

Most medical people who learn R to do their own data analysis do it on their own time. They rarely have time for a semester-long course, and their clinical schedules usually will not allow it. Fortunately, a lot of people learn R on their own, and there is a strong and supportive R Community to help new learners. A 2019 Twitter survey conducted by `@RLadies` found that more than half of respondents were largely self-taught, from books and online resources.

There are a lot of good resources for learning R, so why one more? In part, because the needs of a medical audience are often different. There are distinct needs for protecting health information, generating a descriptive Table One, using secure data tools like REDCap, and creating standard medical journal and meeting output in Word, Powerpoint, and poster formats.

More and more, all science is becoming data science. We are able to track patients, their test results, and even the individual pixels (voxels) of their CT scans electronically, and use those data points to develop new knowledge. While one could argue that health care workers should collect data and bring it to trained statisticians, this does not work nearly as well as you might expect. Most academic statisticians are incentivized to develop new statistical methods, and are not very interested (or incentivized) to do the hand-holding required to wrangle messy clinical data into a manuscript.

There also are simply not enough statisticians to meet the needs of medical science. Having clinicians on the front lines with some data science training makes a big difference, whether in 1854 in London (John Snow) or in 2014

in Flint, Michigan (Mona Hanna-Atisha). Having more clinicians with some training will impact medical care, as they will identify local problems that would have otherwise never reached a statistician, and probably never been addressed with data otherwise.

## 1.5 The Scientific Reproducibility Crisis

Beginning as far back as 1989, with the David Baltimore case, and increasingly publicly through the 2010s, there has been a rising tide of realization that a lot of taxpayer-funded science is done sloppily, and that our standards as scientists need to be higher. The line between carelessly-done science and outright fraud is a thin one, and the case can be made that doing science in a sloppy fashion defrauds the funders, as it leads to results that can not be reproduced nor replicated. Particularly in medicine, where incorrect findings can cause great harm, we should take special care to do scientific research which is well-documented, reproducible, and replicable. This topic as a motivating force for doing careful medical research will be expanded upon in Chapter 1.

## 1.6 What this Book is Not

### 1.6.1 This Book is Not A Statistics Text

This is not an introduction to statistics. I am assuming that you have learned some statistics somewhere in secondary school, undergraduate studies, graduate school, or even medical school. There are lots of statisticians with Ph.D.s who can certainly teach statistics much more effectively than I can. While I have a master's degree in Clinical Research Design and Statistical Analysis (isn't that a mouthful!) from the University of Michigan, I will leave formal teaching of statistics to the pros.

If you need to brush up on your statistics, no worries. There are several excellent (and free!) e-books on that very topic, using R. Some good examples include (go ahead and click through the blue links to explore):

1. Learning Statistics with R (LSR)
2. Modern Dive
3. Teacup Giraffes

We will cover a lot of the same materials as these books, but with a less theoretical and more applied approach. I will focus on specific medical examples, and emphasize issues (like Protected Health Information) that are particularly important for medical data. I am assuming that you are here because you want to analyze your own data in your probably very limited free time.

### 1.6.2 This Book Does Not Provide Comprehensive Coverage of the R Universe

This book is also far from comprehensive in teaching what is available in the R ecosystem. This book should be considered a launch pad. Many of the later chapters will give you a taste of what is available in certain areas, and guide you to resources (and links) that you can explore to learn more and do more beyond the scope of this book.

## 1.7 Some Guideposts

Keep an eye out for Guideposts, which look like this:

**Warnings**

This is a common gotcha. Watch out for this.

**Tips**

This is a helpful tip for debugging.

**Try It Out**

Take what you have learned and try it yourself in the code box below.

**Challenge** - take the next step and try a more challenging example.

Try this more complicated example.

**Explore More** - resources for learning more about a particular topic.

If you want to learn more about Shiny apps, go to https://mastering-shiny.org to see an entire book on the topic.

# Chapter 2

# Getting Started and Installing Your Tools

One of the most intimidating parts of getting started with something new is the actual getting started part. Don't worry, we will walk you through this step-by step.

## 2.1 Goals for this Chapter

- Install R on your Computer
- Install RStudio on your Computer
- Install Git on your Computer
- Get Acquainted with the RStudio IDE

## 2.2 Website links needed for this Chapter

While in many chapters, we will list the R packages you need, in this chapter, you will be downloading and installing new software, so we will list the links here for your reference

- https://www.r-project.org
- https://rstudio.com/products/rstudio/download/
- https://git-scm.com/downloads

## 2.3 Pathway for this Chapter

This Chapter is part of the **TOOLS** pathway. Chapters in this pathway include

- Getting Started and Installing Your Tools
- Updating R, RStudio, and Your Packages
- Advanced Use of the RStudio IDE
- When You Don't Want to Update Packages (Using *renv*)
- Major R Updates (Where Are My Packages?)

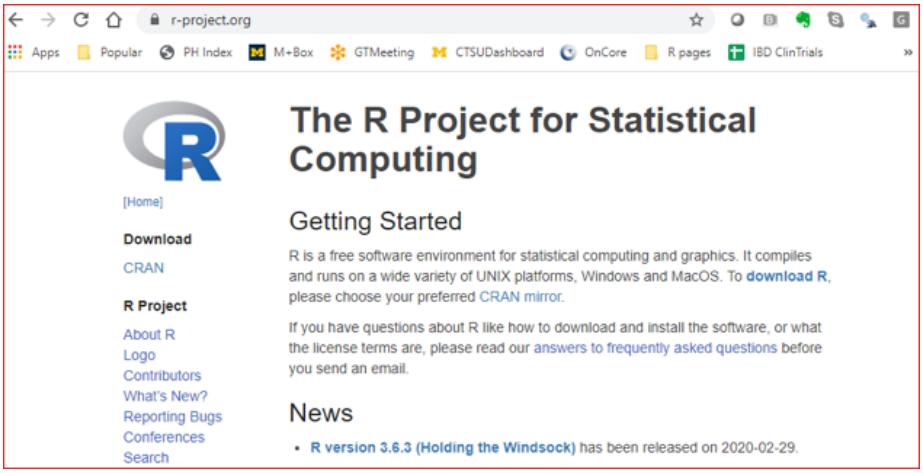
## 2.4 Installing R on your Computer

R is a statistical programming language, designed for non-programmers (statisticians). It is optimized to work with data in tables. It is a very fast and powerful programming engine, but it is not terribly comfortable or convenient. R itself is not terribly user-friendly. It is a lot like a drag racing car, which is basically a person with a steering wheel strapped to an airplane engine.


Very aerodynamic and fast, but not comfortable for the long run (more than about 8 seconds). You will need something more like a production car, with a nice interior and a dashboard, and comfy leather seats.

This equivalent of a comfy coding environment is provided by the RStudio IDE (Integrated Developer Environment). We want you to install both R and RStudio, in that order.
Let's start with installing R.
R is free and available for download on the web. Go to the r-project website to get started.

This screen will look like this

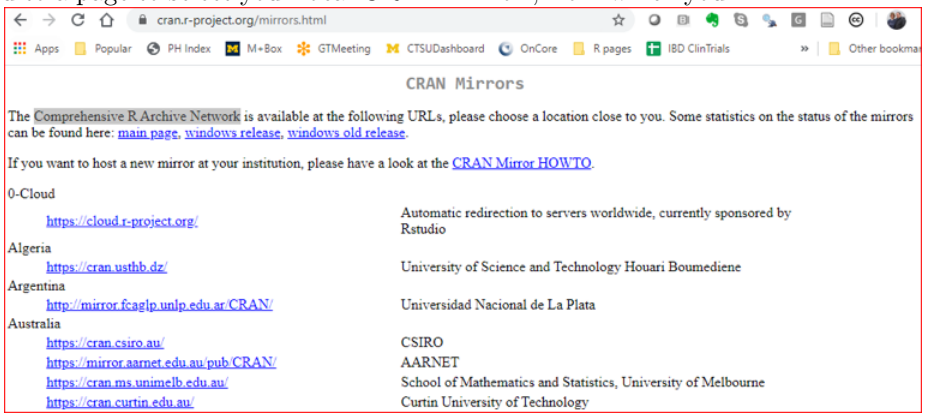



You can see from the blue link (download R) that you can download R, but you will be downloading it faster if you pick a local CRAN mirror.
You might be wondering what CRAN and CRAN Mirrors are. Nothing to do

with cranberries, fortunately. CRAN is the Comprehensive R Archive Network. Each site (mirror) in the network contains an archive of all R versions and packages, and the sites are scattered over the globe. A CRAN Mirror maintains an up to date copy of all of the R versions and packages on CRAN. If you use the nearest CRAN mirror, you will generally get faster downloads.
At this point, you might be wondering what a package is…
A package is a set of functions and/or data that you can download to upgrade and add features to R. It is a lot like a downloadable upgrade to a Tesla that lets you play the video game *Witcher 3* on your console, but more useful.

Another useful analogy for packages is that they are like apps for a smartphone. When you buy your first smartphone, it only comes with the basic apps that allow it to work as a phone, but a notepad and a calculator. If you want to do cool things with your smartphone, you download apps that allow your smartphone to have new capabilities. That is what packages do for your installation of R.


Now let's get started. Click on the blue link that says "download R".
This will take you to a page to select your local CRAN Mirror , from which you will download R.



cran.r-project.org/mirrors.html

Apps Popular PH Index M+Box GTMeeting CTSUDashboard OnCore R pages IBD ClinTrials Other bookmarks

CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page, windows release, windows old release.

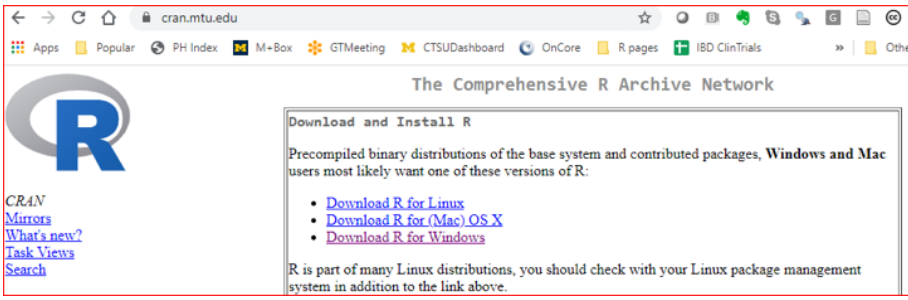
If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

| | |
|---|---|
| 0-Cloud | |
| https://cloud.r-project.org/ | Automatic redirection to servers worldwide, currently sponsored by Rstudio |
| Algeria | |
| https://cran.usthb.dz/ | University of Science and Technology Houari Boumediene |
| Argentina | |
| http://mirror.fcaglp.unlp.edu.ar/CRAN/ | Universidad Nacional de La Plata |
| Australia | |
| https://cran.csiro.au/ | CSIRO |
| https://mirror.aarnet.edu.au/pub/CRAN/ | AARNET |
| https://cran.ms.unimelb.edu.au/ | School of Mathematics and Statistics, University of Melbourne |
| https://cran.curtin.edu.au/ | Curtin University of Technology |

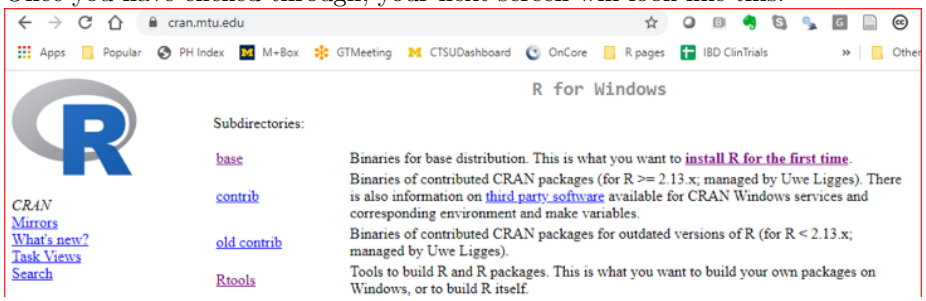
Scroll down to your local country (yes, the USA is at the bottom), and a CRAN mirror near you. This is an example from the state of Michigan, in the USA.

| | |
|---|---|
| USA | |
| https://mirror.las.iastate.edu/CRAN/ | Iowa State University, Ames, IA |
| https://ftp.ussg.iu.edu/CRAN/ | Indiana University |
| https://rweb.crmda.ku.edu/cran/ | University of Kansas, Lawrence, KS |
| https://cran.mtu.edu/ | Michigan Technological University, Houghton, MI |
| https://repo.miserver.it.umich.edu/cran/ | MBNI, University of Michigan, Ann Arbor, MI |
| http://cran.wustl.edu/ | Washington University, St. Louis, MO |
| http://archive.linux.duke.edu/cran/ | Duke University, Durham, NC |
| https://cran.case.edu/ | Case Western Reserve University, Cleveland, OH |
| https://ftp.osuosl.org/pub/cran/ | Oregon State University |
| http://lib.stat.cmu.edu/R/CRAN/ | Statlib, Carnegie Mellon University, Pittsburgh, PA |
| http://cran.mirrors.hoobly.com/ | Hoobly Classifieds, Pittsburgh, PA |
| https://mirrors.nics.utk.edu/cran/ | National Institute for Computational Sciences, Oak Ridge, TN |
| https://cran.revolutionanalytics.com/ | Revolution Analytics, Dallas, TX |

Once you click on a CRAN Mirror site to select the location, you will be taken to

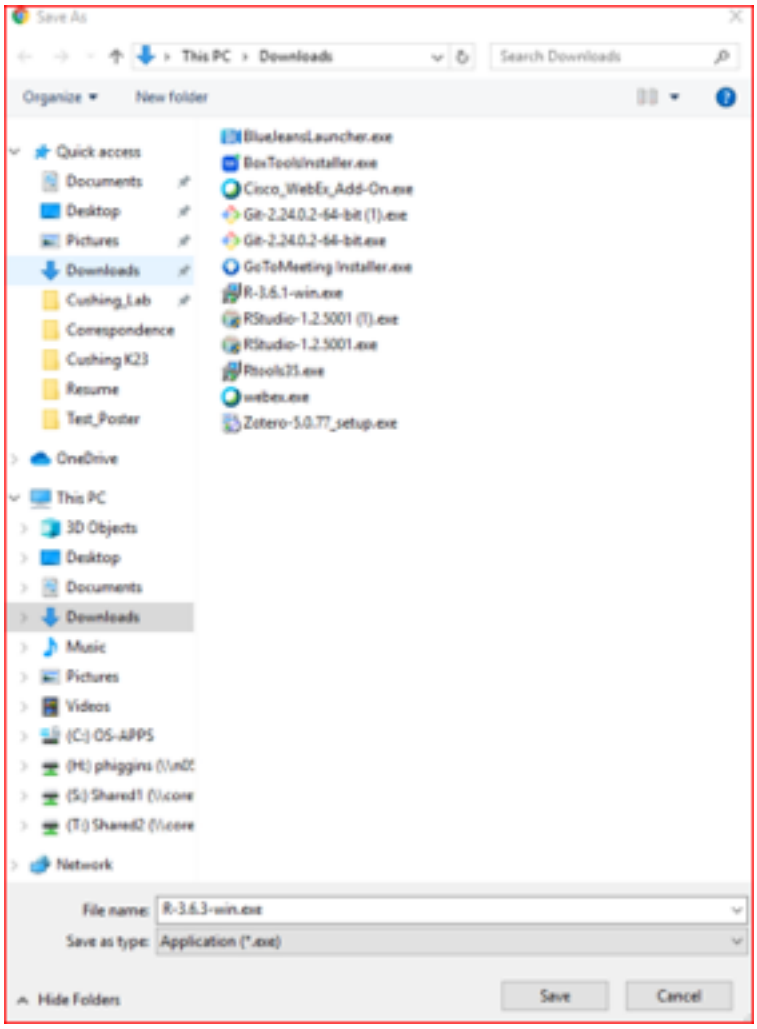



the actual Download site. Select the link for the operating system you want to use. We will walk through this with Windows first, then Mac. If you are using a Mac, skip forward to the Mac install directions. If you are computer-savvy enough to be using Linux, you can clearly figure it out on your own (it will look a lot like these).
Once you have clicked through, your next screen will look like this:



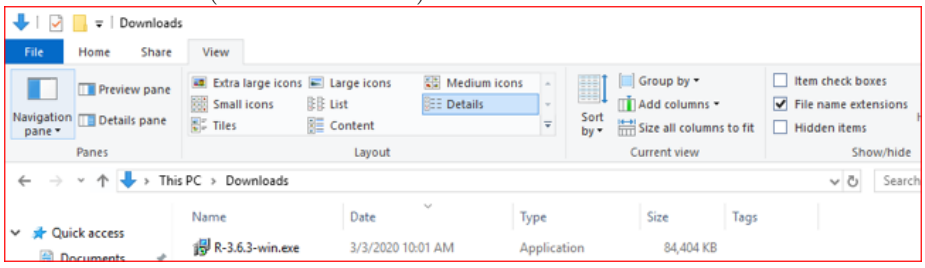

You want to download both base and Rtools (you might need Rtools later). The base link will take you to the latest version, which will look something like this.

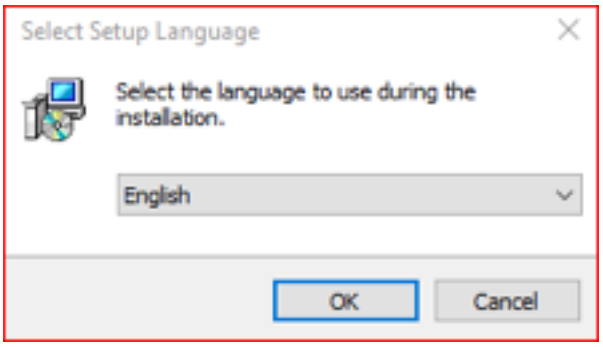

Click on this link, and you will be able to save a file named R-N.N.N-win.exe (Ns depending on version number) to your Downloads folder. Click on the Save but-



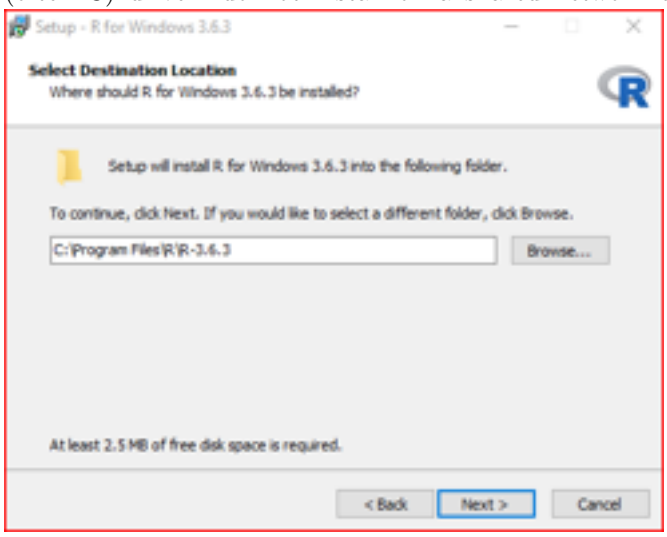

ton to save it.

Now, go to your Downloads folder in Windows, and double click on the R installation file (R-N.N.N-win.exe). Click Yes to allow this to install.

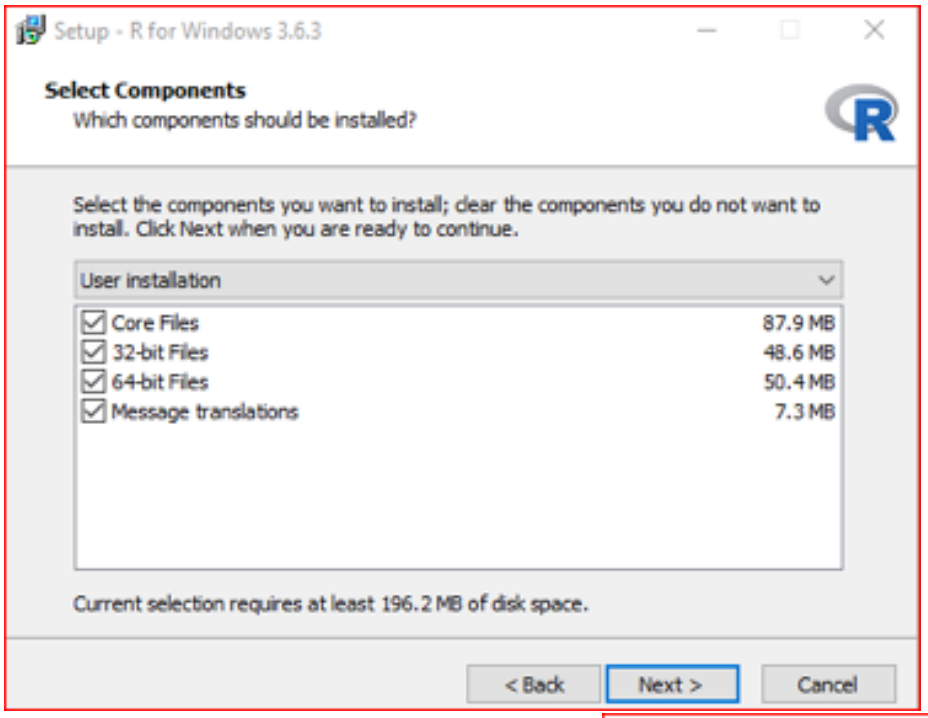




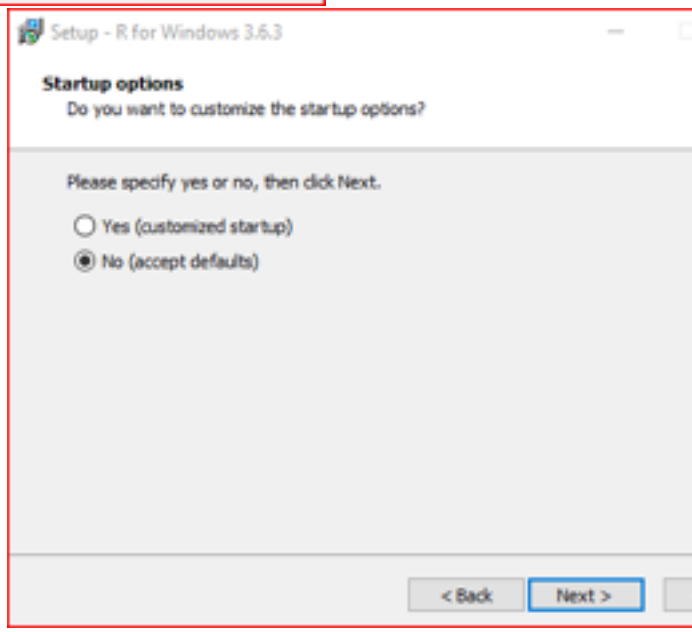

Now select your language option. You will be asked to accept the GNU license - do so. Click Yes to allow this to install. Then select where to install - generally use the default- a local (often C) drive - do not install on a shared network drive or in the cloud.



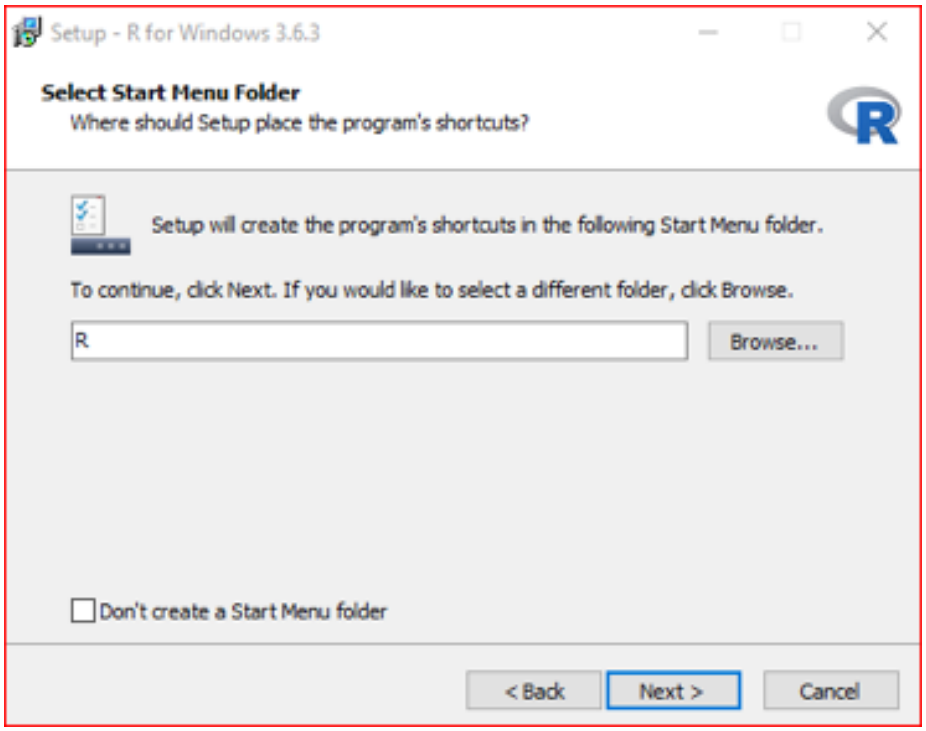

Then select the Components - generally use the defaults, but newer computers can skip the 32 bit ver-



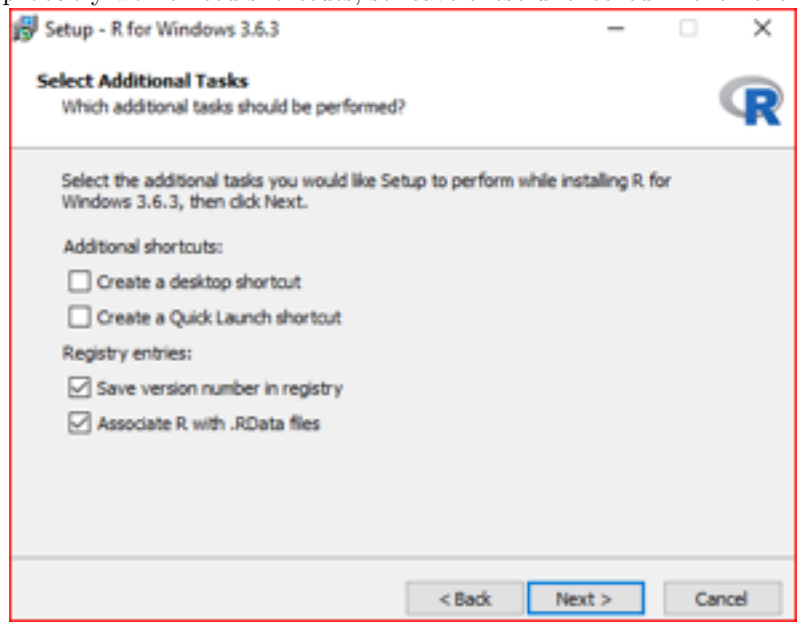

sion.




In the next dialog box, accept the default startup options. You can choose the start menu folder. The default R folder is fine.




You probably won't need shortcuts, so leave these unchecked in the next dialog box.




Then the Setup Wizard will appear - click Finish, and the rest of the installation will occur.

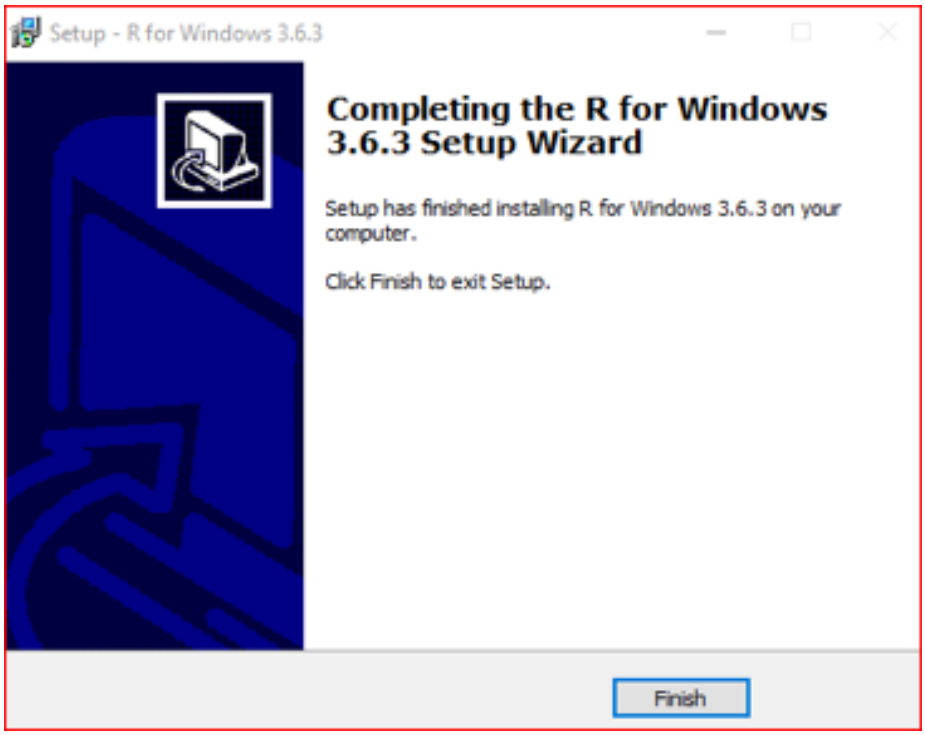
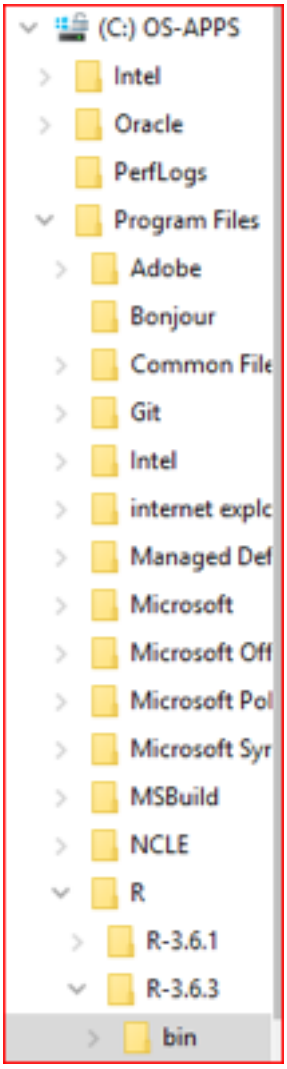
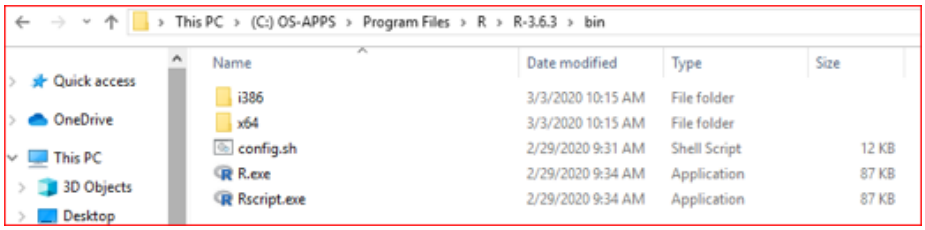



Figure 2.1: install_wizard

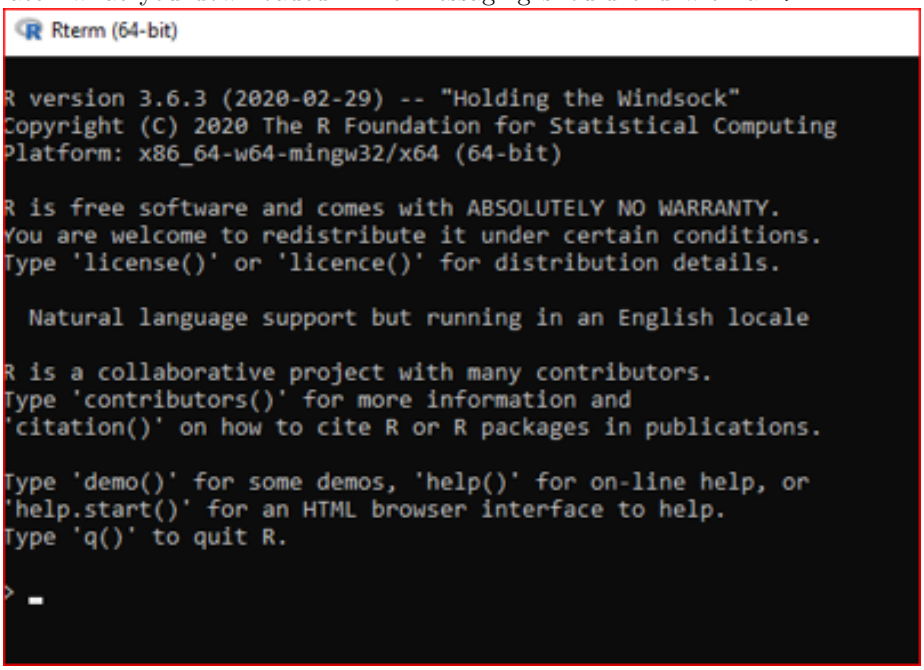
### 2.4.1 Testing

Now you want to test whether your Windows installation was successful. Can you find R and make it work? Hunt for your C folder, then for OS-APPS within that folder. Keep drilling down to the Program Files folder. Then the R folder, and the current version folder within that one (R-N.N.N). Within that folder will be the bin folder, and within that will be your R-N.N.N.exe file. Double click on this to run it. The example paths below can help guide you.



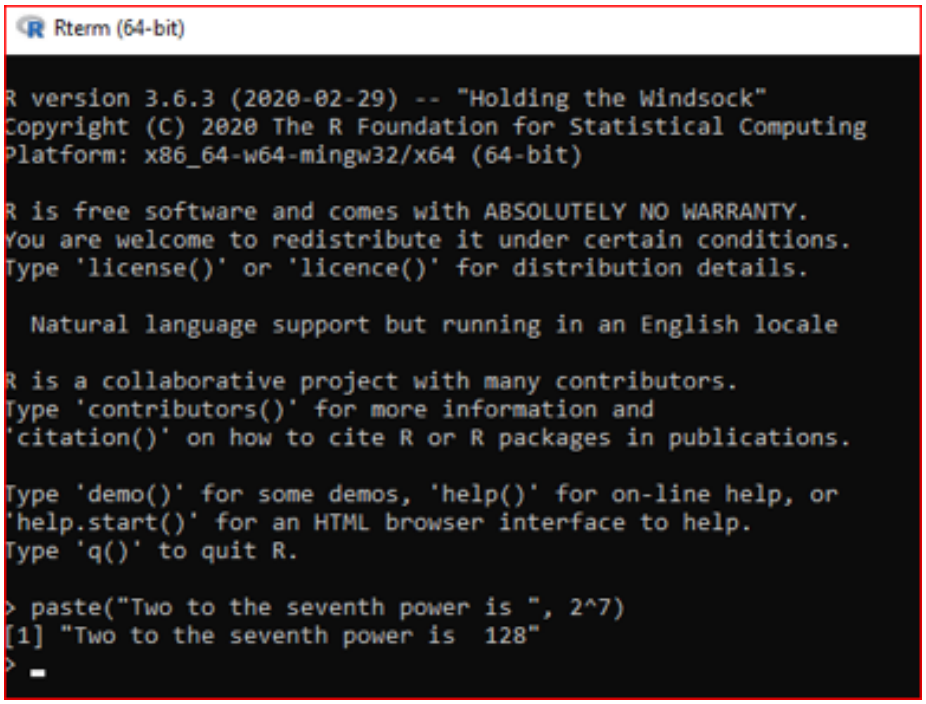



Opening the exe file will produce a classic 2000-era terminal window, called Rterm, with 64 bit if that is what your computer uses. The version number should match what you downloaded. The messaging should end with a ">" prompt.




At this prompt, type in:
paste('Two to the seventh power is', 2^7) (don't leave out the comma) - then press the Enter key.

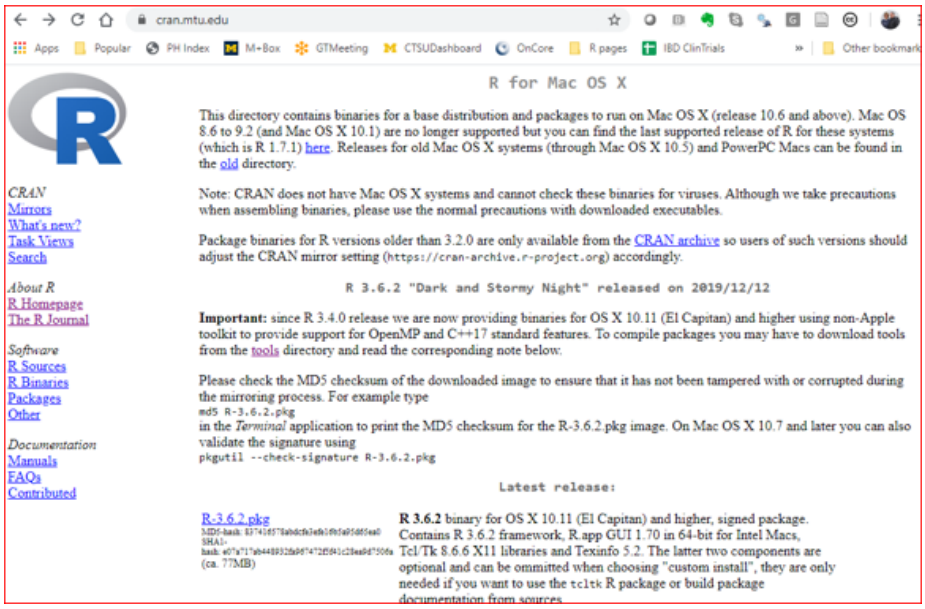
This should produce the following:



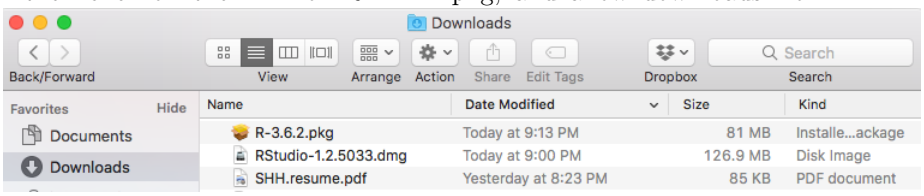

Note that you have explained what is being done and computed the result.

### 2.4.2 Mac Install of R

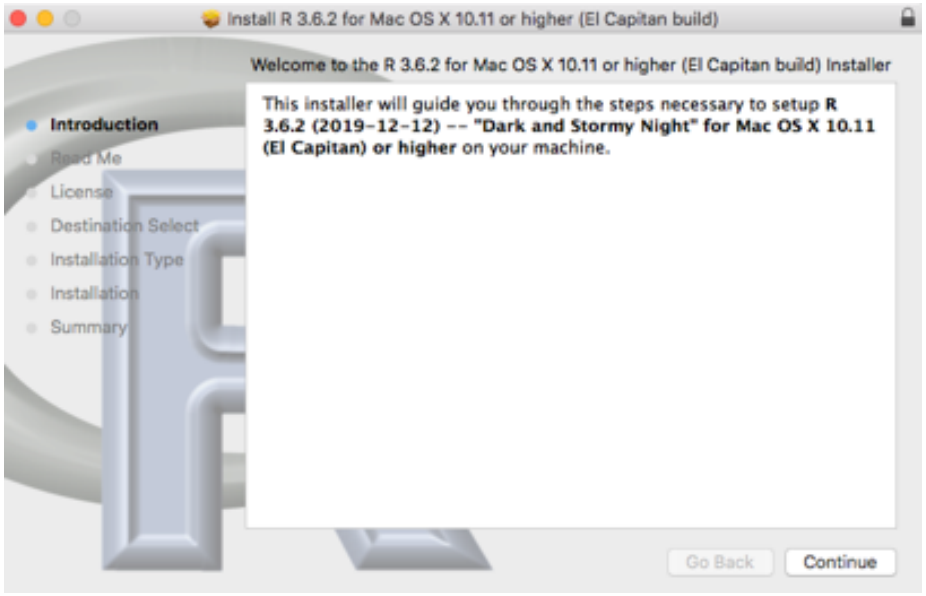
The installation for Mac is very similar, but the windows look a bit different. At the Download Version page, you click on the Mac Download. You will then click on the link for R-N.N.N.pkg, and allow downloads from CRAN.




Then go to Finder, and navigate to the Downloads folder. Click on R-N.N.N.pkg You will then click on the link for R-N.N.N.pkg, and allow downloads from CRAN.




Click on Continue on 2 consecutive screens to download



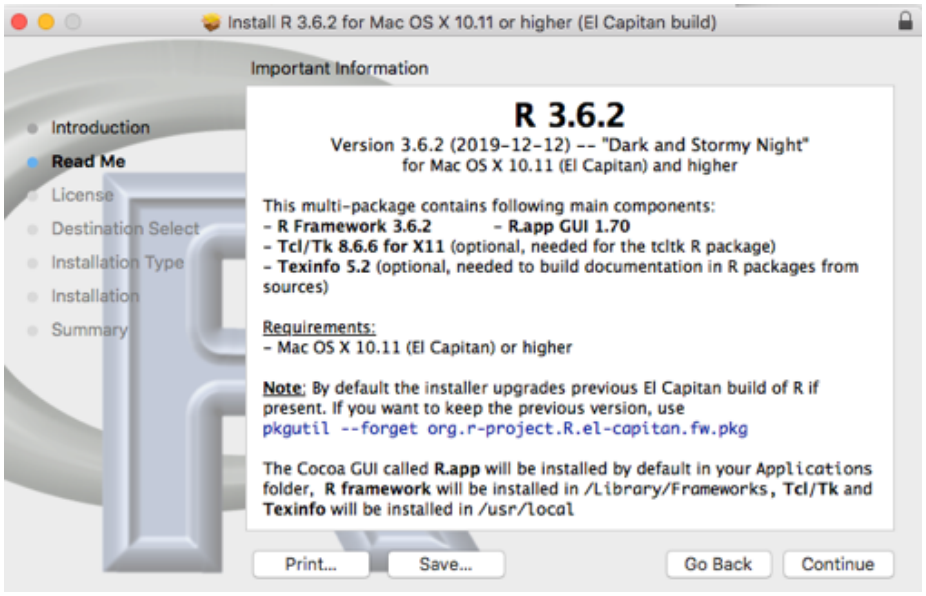
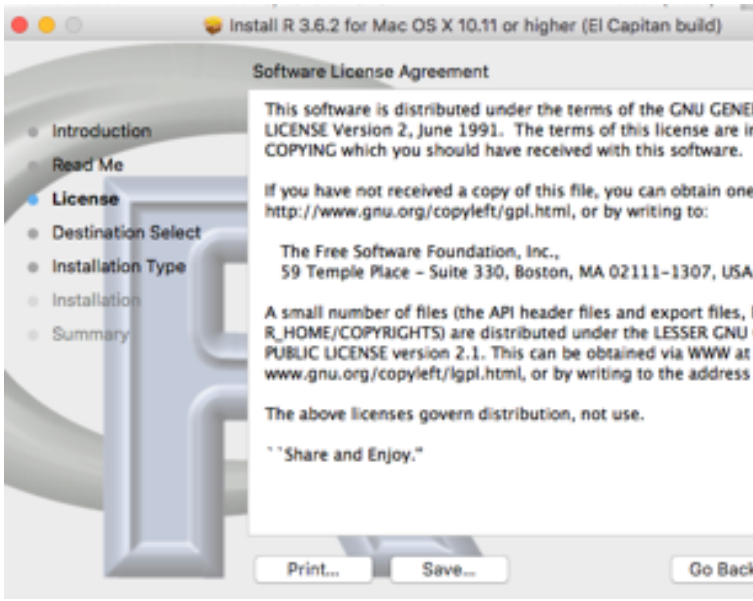
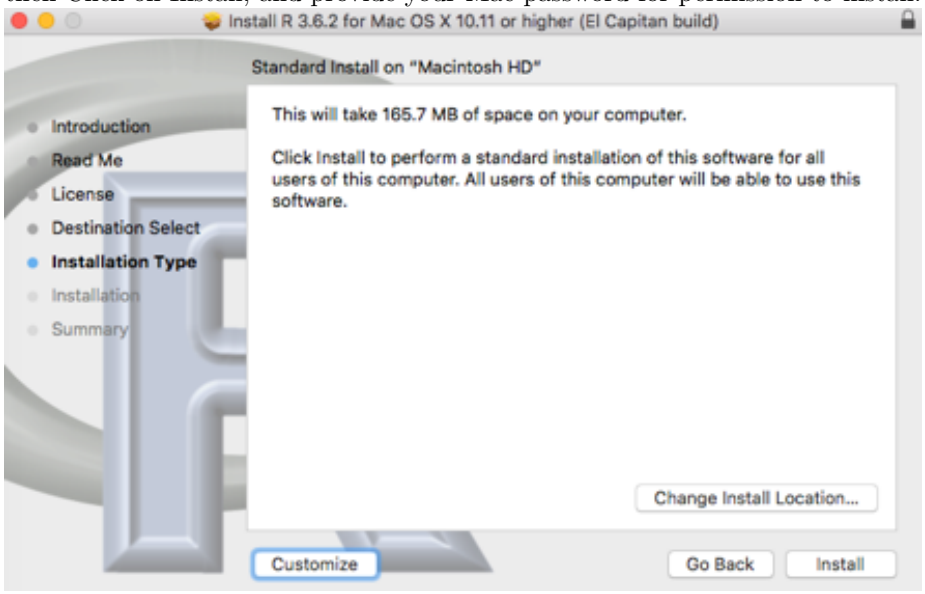




Figure 2.2: cont2_mac



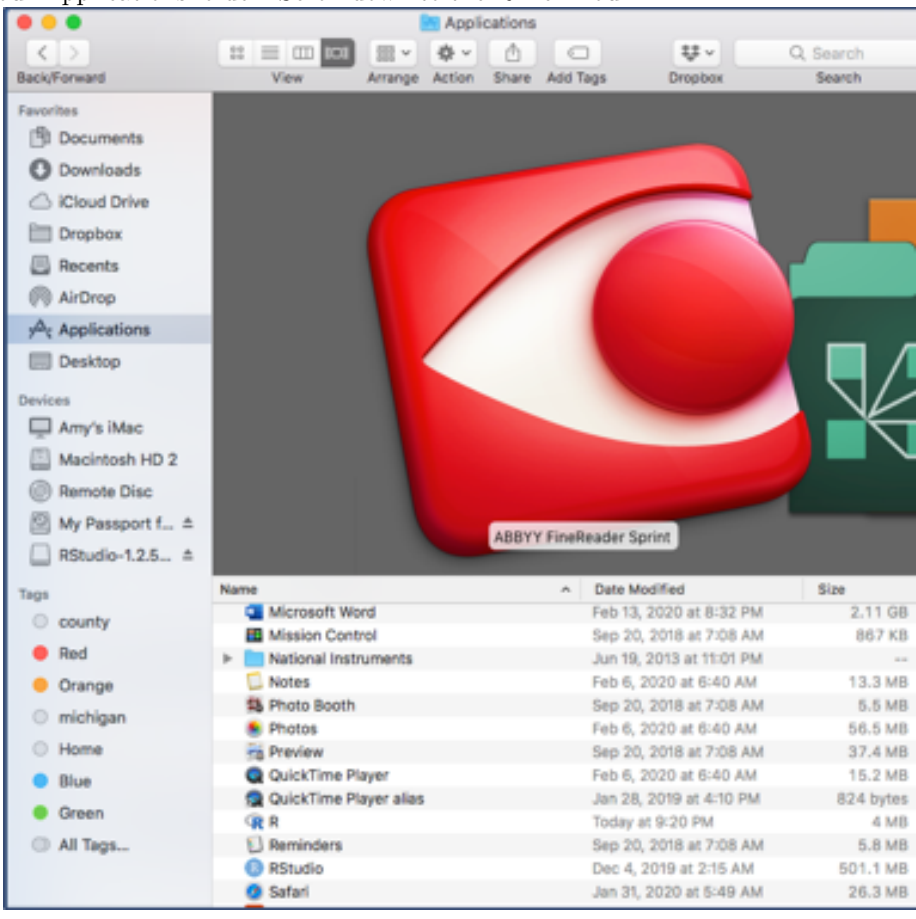

Then you need to agree with the License Agreement, then Click on Install, and provide your Mac password for permission to install.



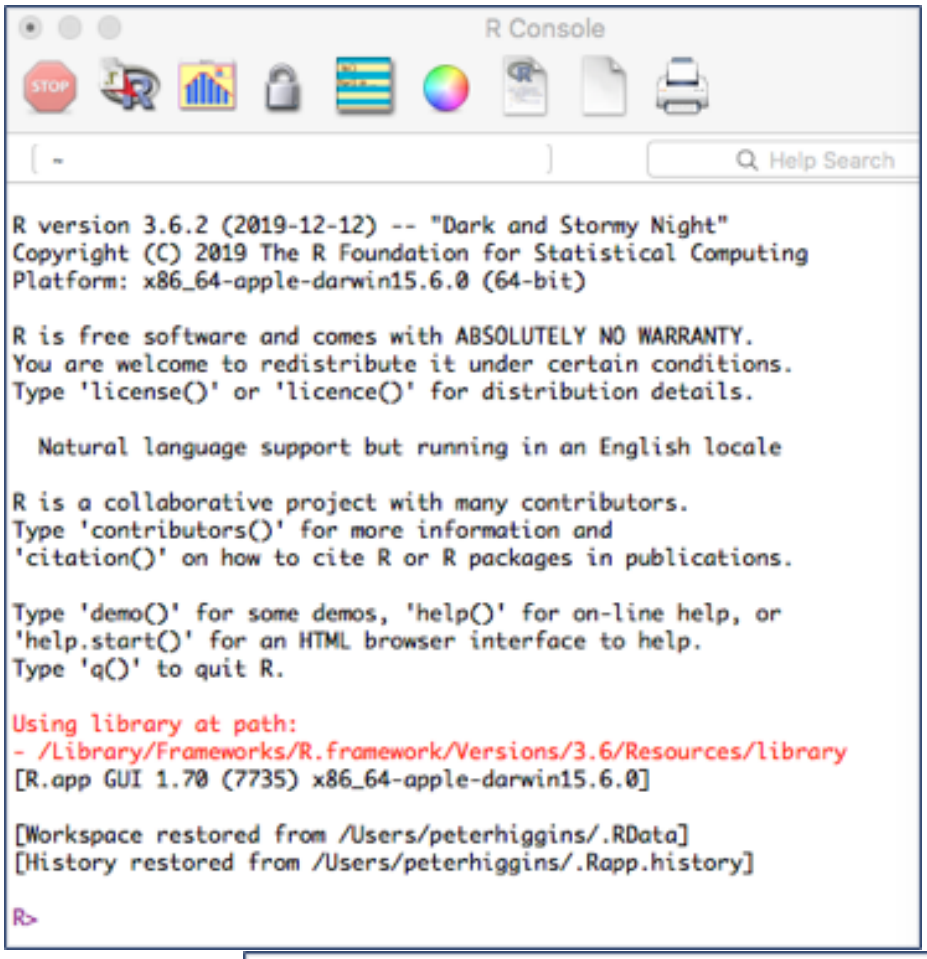

When the installation is complete, click on the Close button. Accept the prompt to move the installer file to the trash.

### 2.4.3 Testing R on the Mac

Go to Finder, and then your Applications folder. Scroll down to the R file. Dou-




ble click on this to run it. You should get this 2000-era terminal window named R Console. The version number should match what you downloaded, and the messaging should end with a ">" prompt. At this prompt, type in paste("Two to the seventh power is", 2^7) (DON'T leave out the comma or the quotes)




This should result in

```
R> paste('Two to the seventh power is', 2^7)
[1] "Two to the seventh power is 128"
R>
```

### 2.4.4 Successful testing!

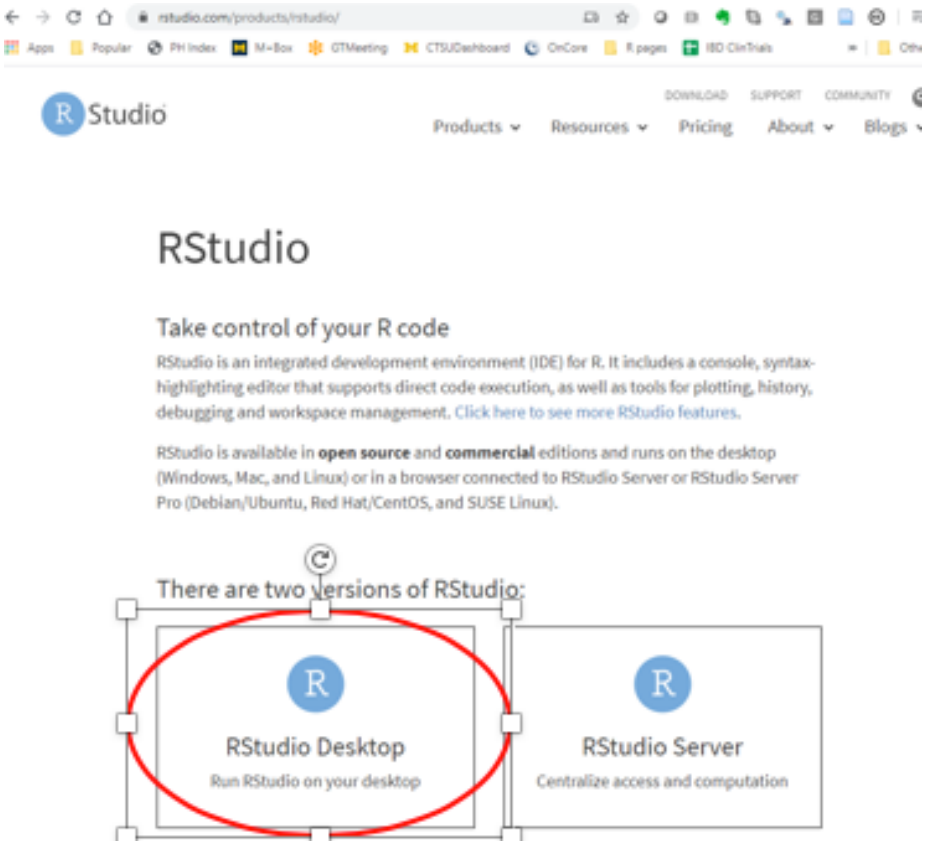
Awesome. You are now Ready to R!

## 2.5 Installing RStudio on your Computer

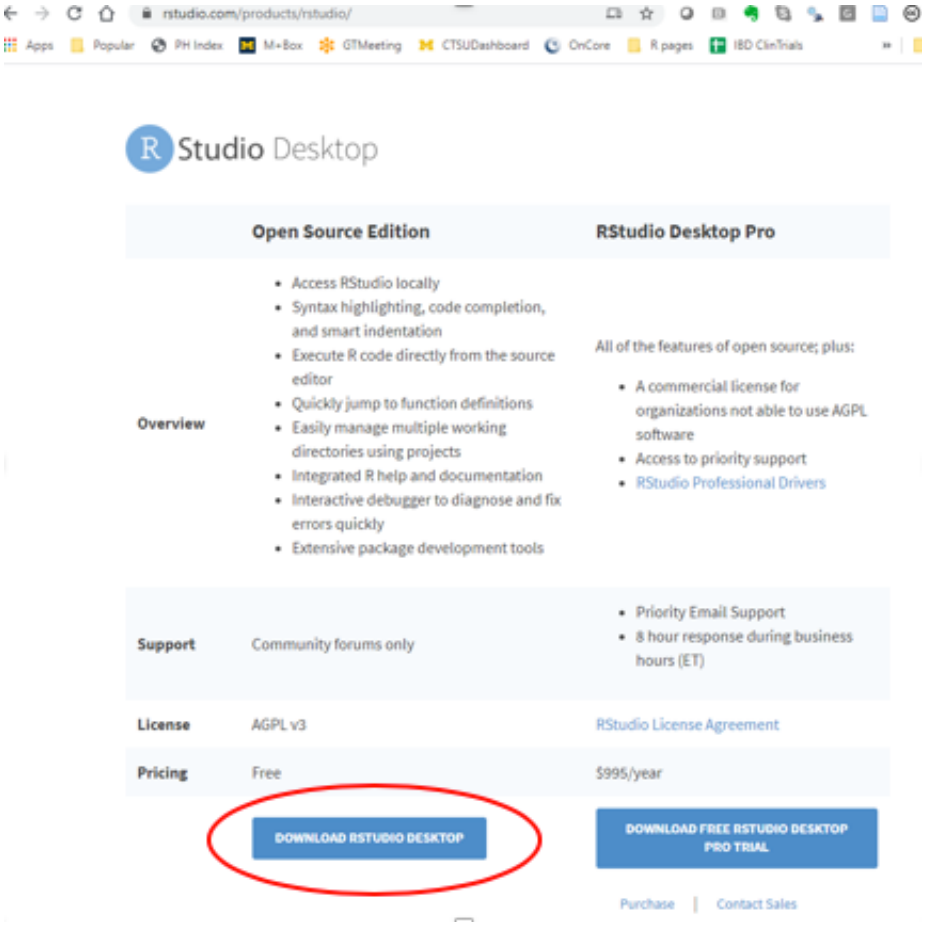
Now that R is working, we will install RStudio. This is an IDE (Integrated Development Environment), with lots of bells and whistles to help you do reproducible medical research. This is a lot like adding a dashboard with polished walnut panels, a large

video screen map, and heated car seats with Corinthian Leather. Not absolutely necessary, but nice to have. The RStudio IDE wraps around the R engine to make your experience more comfortable and efficient.

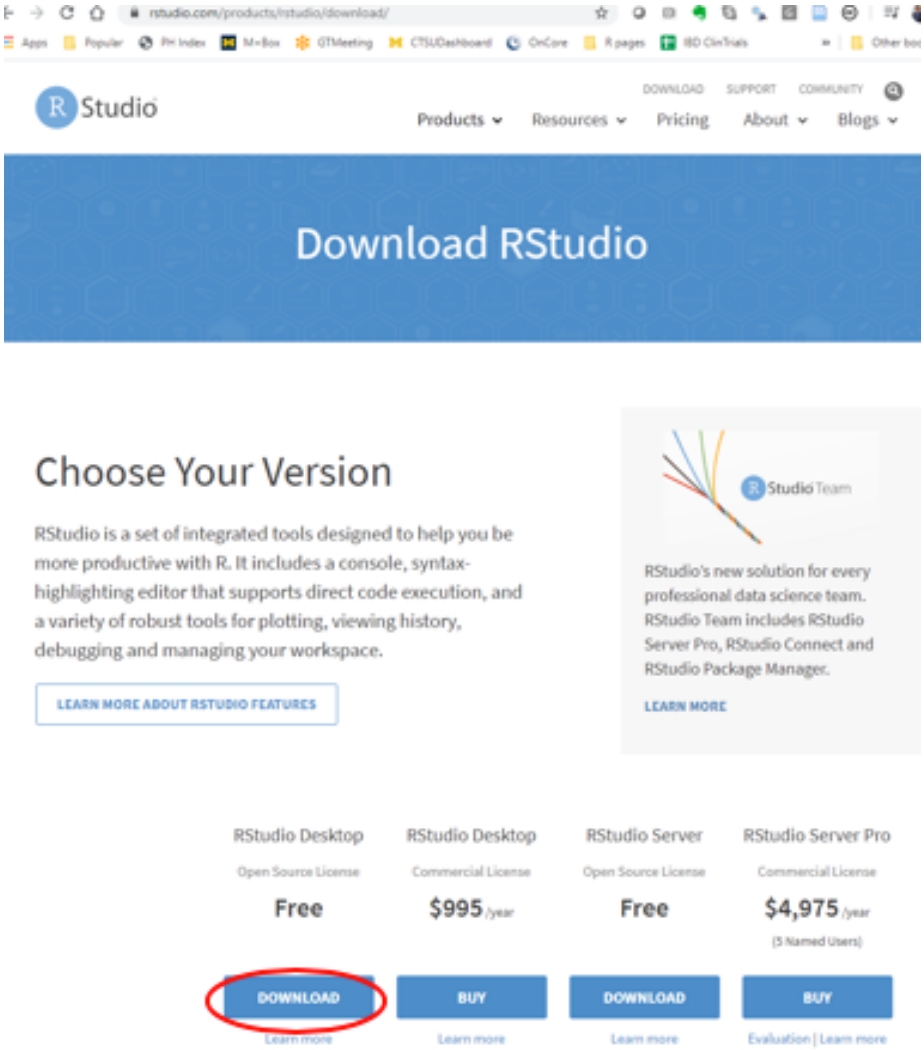
Fortunately, RStudio is a lot cheaper than any of these cars. In fact, it is free and open source. You can download it from the web at: rstudio Click on the RStudio Desktop icon to begin.




This will take you to a new site, where you will select the Open Source Edition of RStudio Desktop

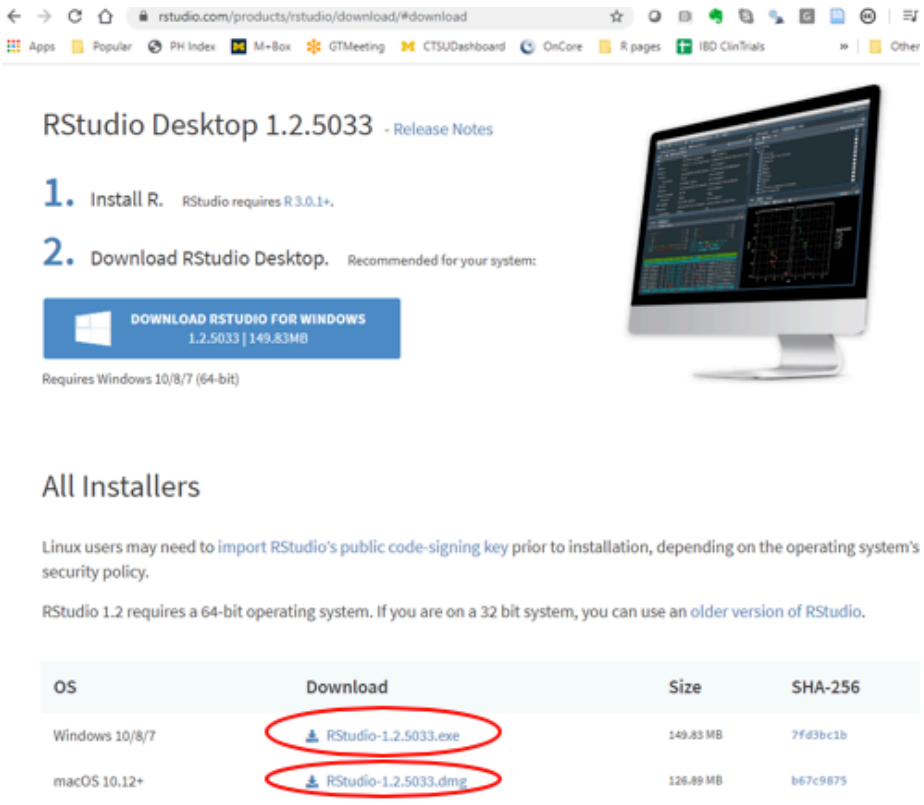



This will take you to a new site, where you will select the Free Version of RStudio



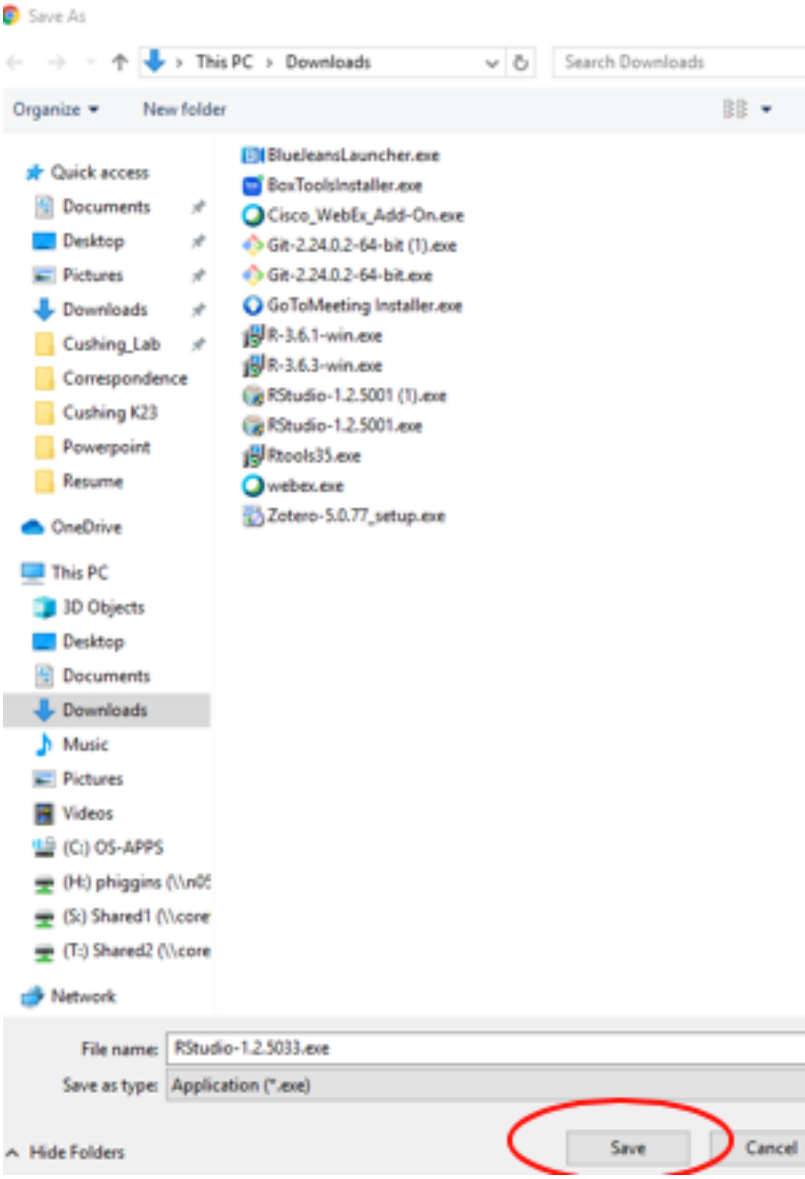

Desktop

Now select the right version for your Operating syxtem - Windows or Mac.

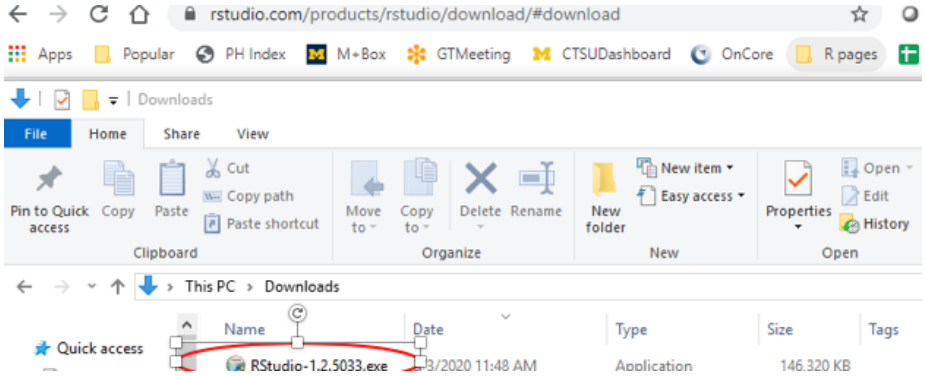

rstudio.com/products/rstudio/download/#download
Apps
Popular
PH Index
M+Box
GTMeeting
CTSUDashboard
OnCore
R pages
IBD ClinTrials
Other
RStudio Desktop 1.2.5033 - Release Notes
1. Install R. RStudio requires R 3.0.1+.
2. Download RStudio Desktop. Recommended for your system:
DOWNLOAD RSTUDIO FOR WINDOWS
1.2.5033 | 149.83MB
Requires Windows 10/8/7 (64-bit)
All Installers
Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.
RStudio 1.2 requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.
OS
Download
Size
SHA-256
Windows 10/8/7
RStudio-1.2.5033.exe
149.83 MB
7fd3bc1b
macOS 10.12+
RStudio-1.2.5033.dmg
126.89 MB
b67c9875

### 2.5.1 Windows Install

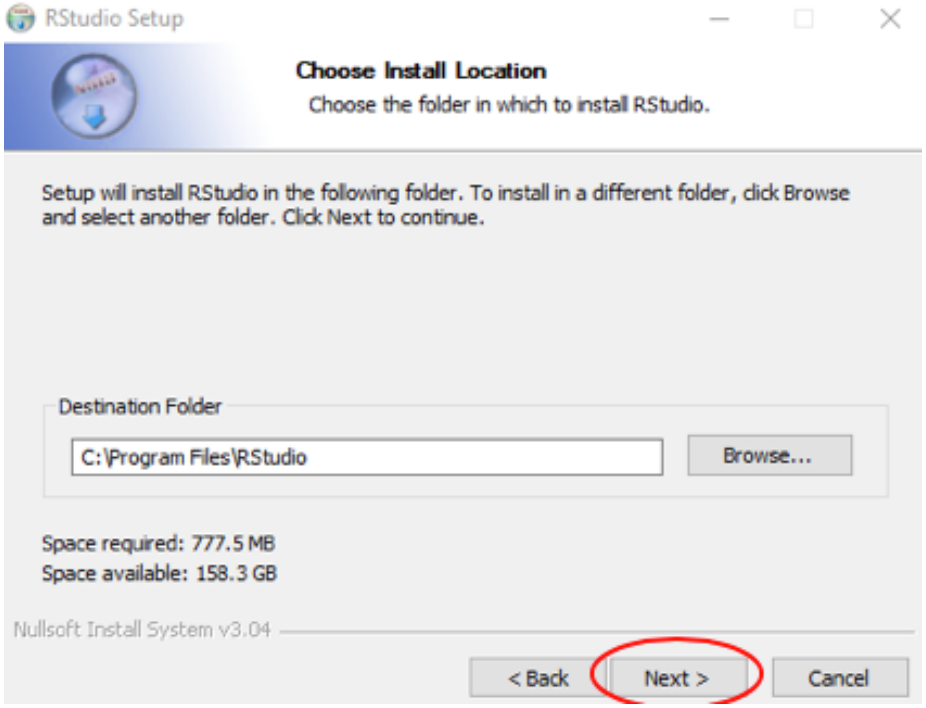
Now save the Rstudio.N.N.N.exe file (Ns will be digits representing the version number) to your downloads folder.



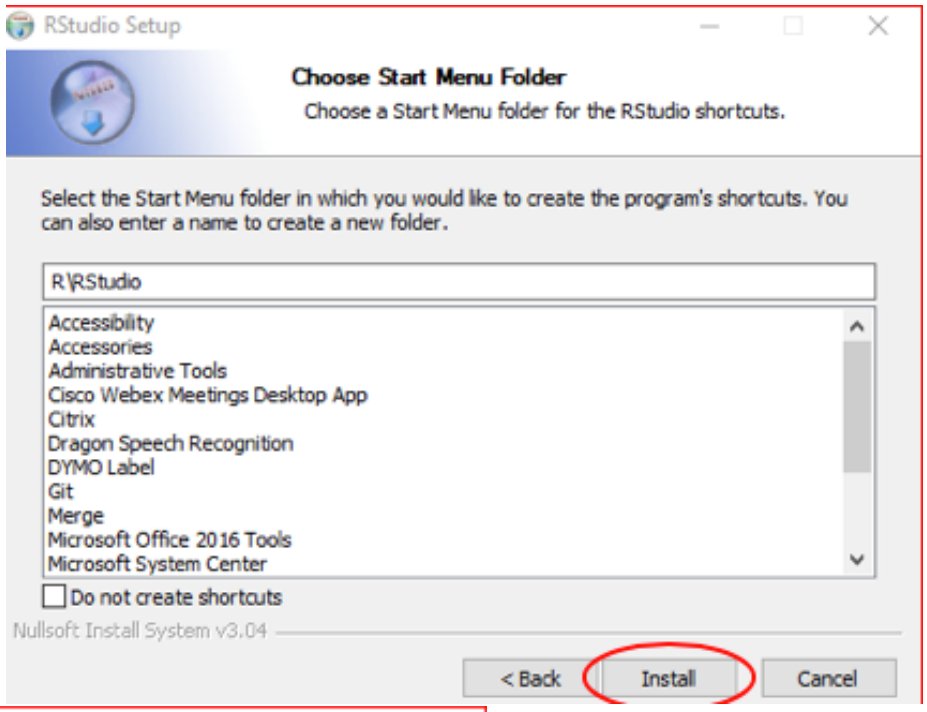

Now go to your downloads folder, and double click on the Rstudio.N.N.N.exe file.



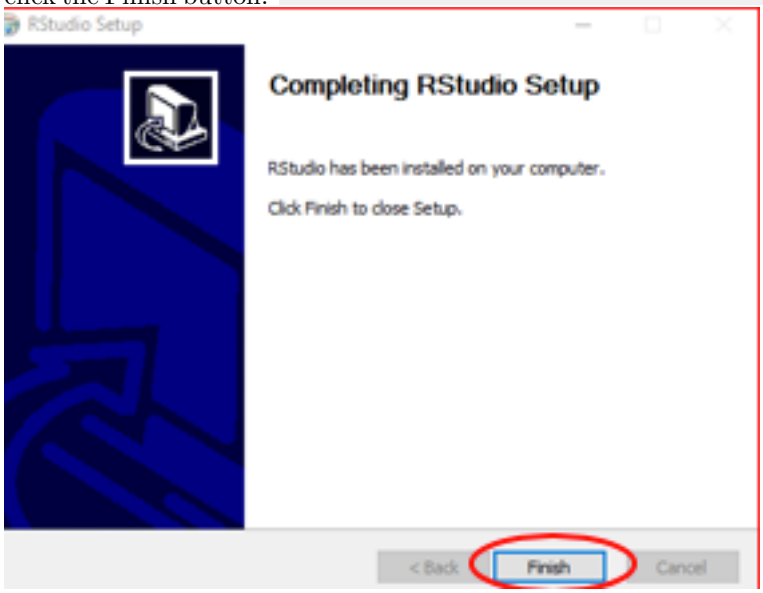

Allow this app to make changes. Click Next to Continue, and Agree to the Install Location.




Click Install to put Rstudio in the default Start Menu Folder, and when done,




click the Finish button.




Now select your preferred language option, accept the GNU license, Click Yes to allow this to install. Select where to install. This is generally on a local (often C:) drive, and usually **not** a shared network drive or in the cloud.

### 2.5.2 Testing Windows RStudio

Now you should be ready to test your Windows installation of Rstudio. Open your Start menu Program list, and find Rstudio. Pin it as a favorite now. Click to Open Rstudio. Within the Console window of Rstudio, an instance of R is started up. Check that the version number matches the version of R that you downloaded. Now run a test at the prompt (">") in the Console window. Type in `paste("Three to the 5th power is", 3^5)`

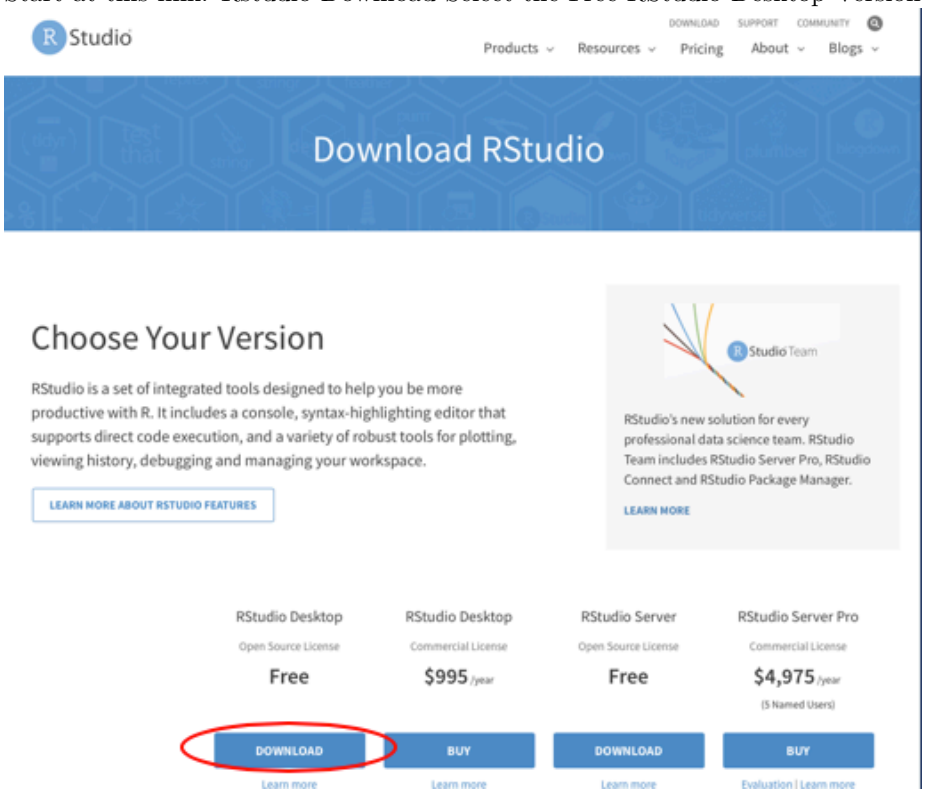
**do not leave out the quotes or the comma**

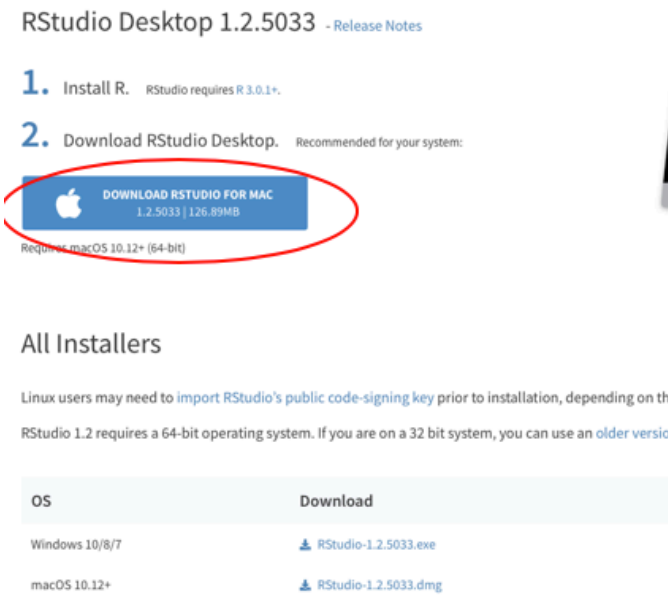
```
R> paste("Three to the 5th power
[1] "Three to the 5th power is 24
```

Then press the enter key and this should be your result:
A successful result means that you are ready to roll in Rstudio and R!

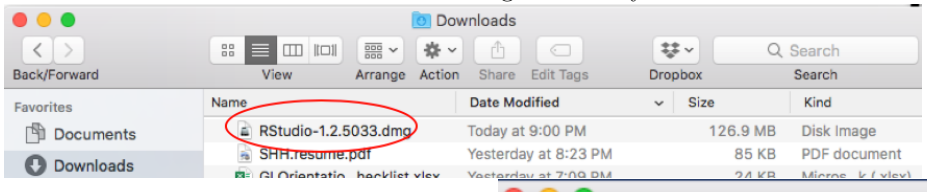
### 2.5.3 Installing RStudio on the Mac

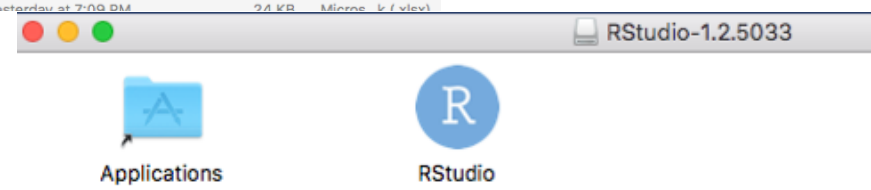
Start at this link: Rstudio Download Select the Free RStudio Desktop Version






Then click on the big button to Download RStudio for Mac.
After the Download is complete, go to Finder and the Downloads Folder. Double click on the Rstudio.N.N.N.dmg file in your Downloads folder.




This will open a window that looks like this
Use your mouse to drag the RStudio icon into the Applications folder. Now go back to Finder, then into the Applications folder. Double click on the RStudio icon, and click OK to Open. Pin your RStudio to the Dock. Double Click to run RStudio. RStudio will open an instance of R inside the Console pane of RStudio with the version number of R that you installed, and a “>” prompt.

### 2.5.4 Testing the Mac Installation of RStudio

Type in `paste("Three to the 5th power is", 3^5)` **do not leave out the quotes or the comma** Then press the enter key and this should be your result.

```
R> paste("Three to the 5th power is", 3^5)
[1] "Three to the 5th power is 243"
```

A successful result means that you are ready to roll in Rstudio and R!

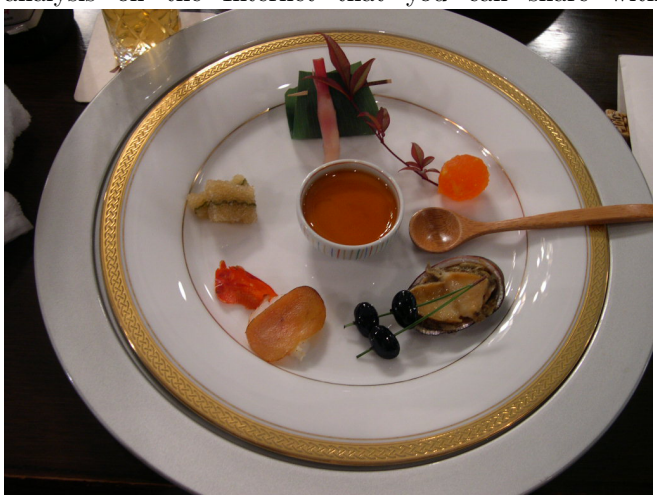

## 2.6 Installing Git on your Computer

## 2.7 Getting Acquainted with the RStudio IDE

# Chapter 3

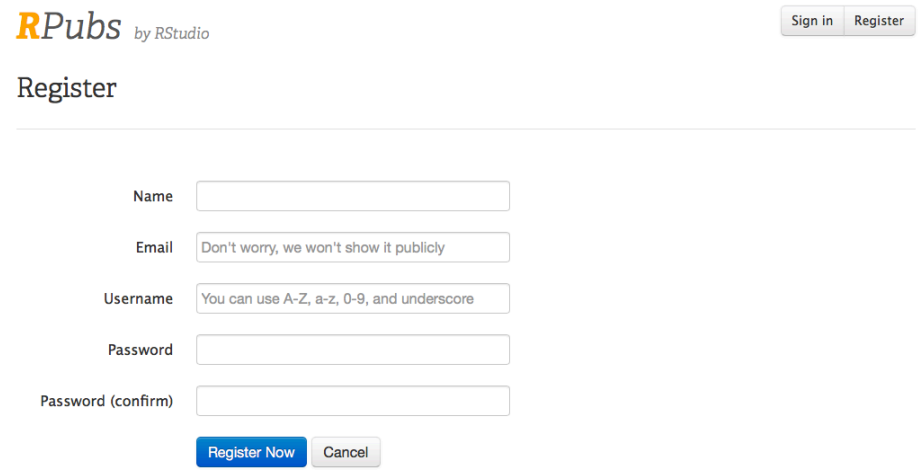
# A Tasting Menu of R

In this chapter, we will introduce you to a lot of neat things that you can do with R and RStudio, and you will publish a simple data analysis on the Internet that you can share with friends and family.

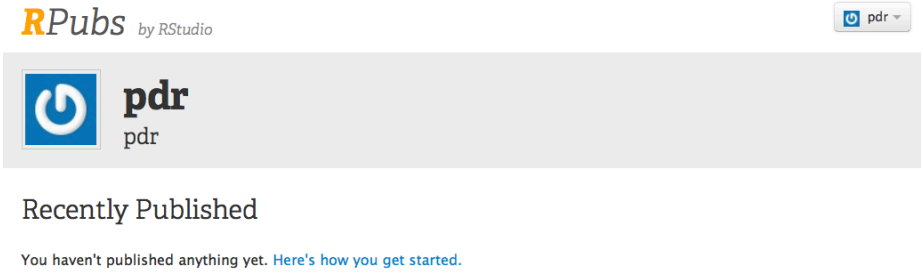


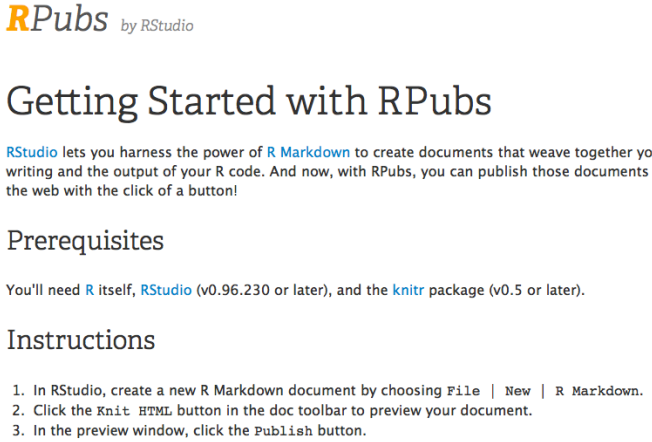
## 3.1 Setting the Table

At the end of this chapter, you will publish a data analysis to *RPubs*, a free website site where you can share your data analyses and visualizations. First you will need to set up an account on RPubs. Start by opening a new tab in your browser, and navigating to this RPubs link. It should look like the image below.




Enter your name, email, username and password, and click on the *Register Now* button, and you will be set up to use RPubs.
This will bring you to this page. In the image below, we have set up an account for pdr.




Click on the *Here's How You Get Started* link.
You are now all set up and ready to go. Now you have a place on the internet to share your R creations!

## 3.2 Goals for this Chapter

- Open a New Rmarkdown document
- Read in Data from a file
- Wrangle Your Data
- Visualize Your Data
- Publish your work to RPubs
- Check out Interactive Plots
- Check out Animated Graphics
- Check out a Clinical Trial Dashboard
- Check out a Shiny App

## 3.3 Packages needed for this Chapter

You will need to enter this line of code into your console, to make sure that the tidyverse package is installed on your computer. `install.packages("tidyverse")`

In the setup chunk of your Rmarkdown document, you will need to access the tidyverse package with one line of code: `library(tidyverse)`

## 3.4 Website links needed for this Chapter

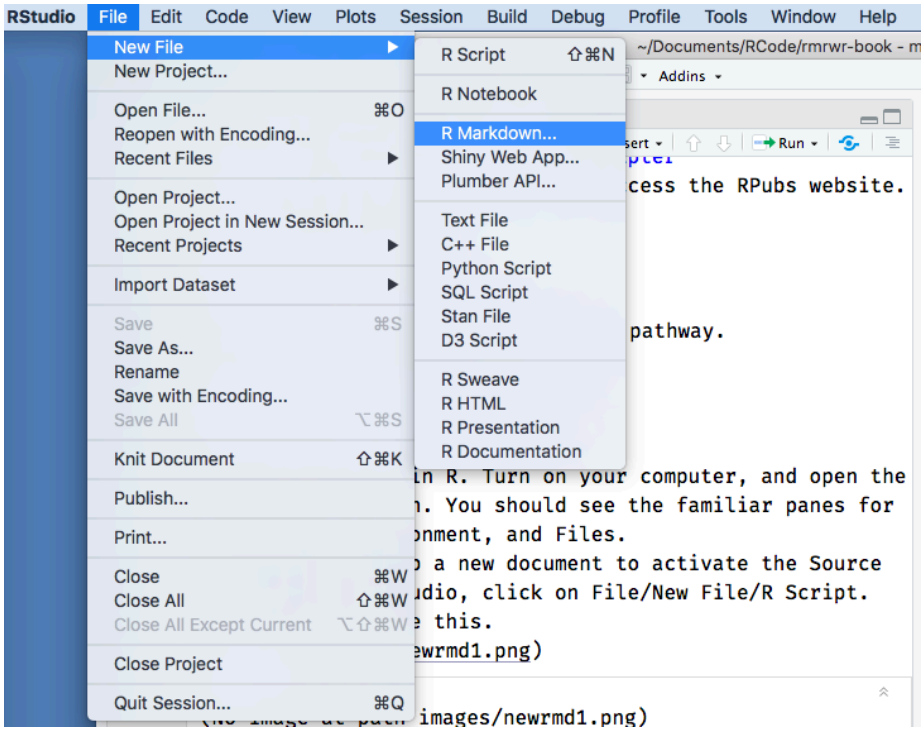
In this chapter, you will need to access the RPubs website.
- https://rpubs.com/

## 3.5 Pathway for this Chapter

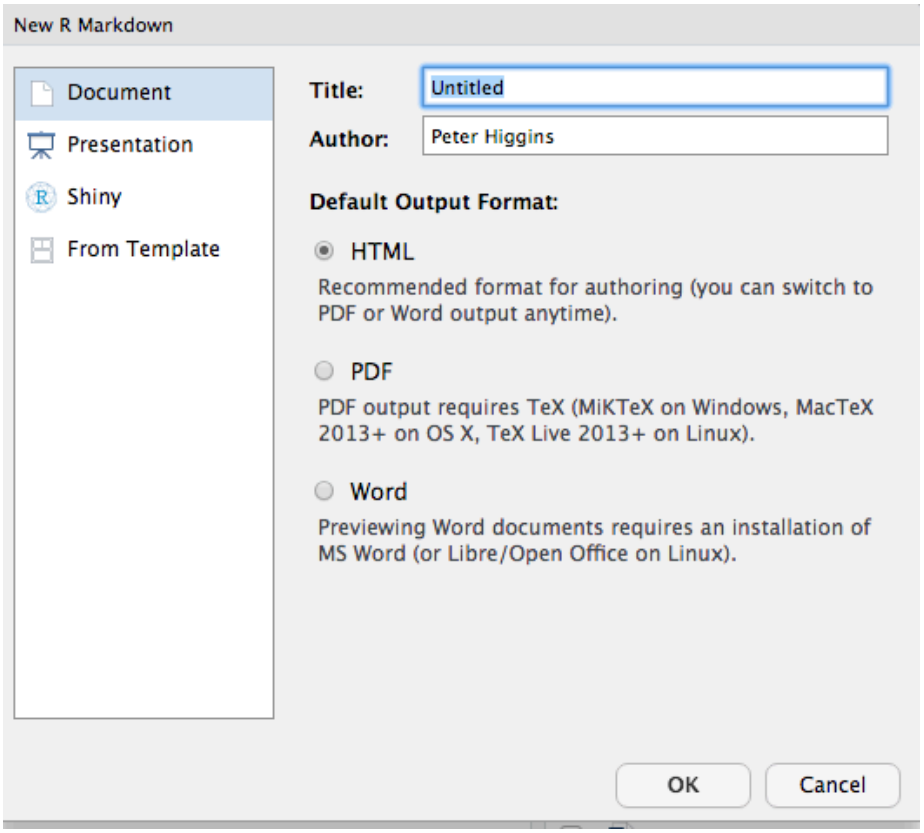
This Chapter is part of the **XXX** pathway. Chapters in this pathway include

## 3.6 Open a New Rmarkdown document

Let's get started in R. Turn on your computer, and open the RStudio application. You should see the familiar panes for the Console, Environment, and Files.
You need to open up a new document to activate the Source pane. While in RStudio, click on File/New File/R Script. It should look like this.

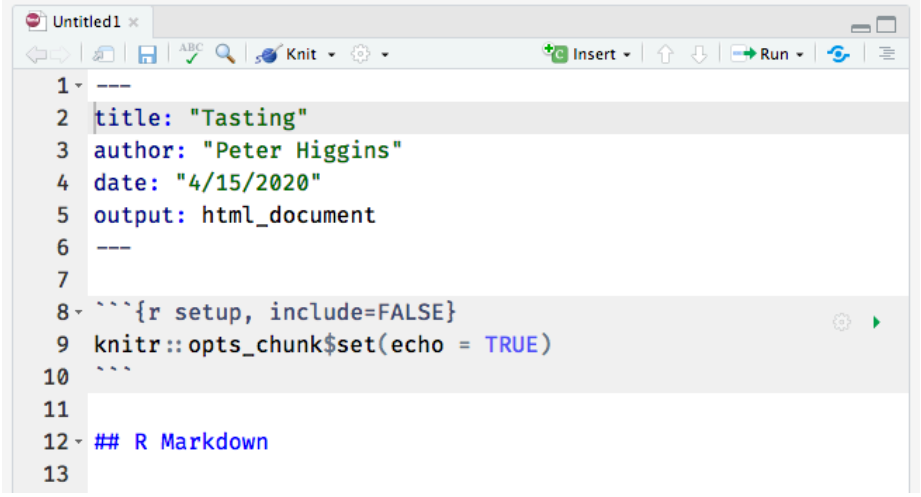



Now you will see the window below. Rename the document from “Untitled” to




"Tasting", and click the OK button.

Now the file is open, and looks like the window below. Click on the save icon (like a floppy disk in the top left), and save this document as tasting.Rmd.




You have created a new Rmarkdown document. An Rmarkdown document

lets you mix data, code, and descriptive text. It is very helpful for presenting and explaining data and visualizations. An Rmarkdown document can be converted (Knit) to HTML for a web page, Microsoft Word, Powerpoint, PDF, and several other formats.

Code chunks are in a gray color, and both start and end with 3 backticks (“‘).

Text can be body text, or can be headers and titles. The number of hashtags before some header text defines what level the header is.
You can insert links, pictures, and YouTube videos into Rmarkdown documents if it is helpful to explain your point. The first code chunk in each Rmarkdown document is named `setup`. The name comes after the left curly brace and the r (`{r`) at the beginning of the setup chunk. The letter `r` tells RStudio that what is coming on the next line is R code (RStudio can also use SQL, C++, python, and several other languages). After the comma, you can define options for this code chunk. In this case, the option `include` is set to FALSE, so that when this Rmarkdown document is knitted, this code chunk will not appear.

## 3.7 Read in Data from a file

## 3.8 Wrangle Your Data

## 3.9 Visualize Your Data

## 3.10 Publish your work to RPubs

## 3.11 The Dessert Cart

Below are some examples of neat things you can do with medical data in R. These are more advanced approaches, but completely doable when you have more experience with R.

### 3.11.1 Interactive Plots

### 3.11.2 Animated Graphics

### 3.11.3 A Clinical Trial Dashboard

### 3.11.4 A Shiny App

### 3.11.5 An Example of Synergy in the R Community

One of the remarkable things about the open source R community is that people build all kinds of new R functions and packages that are useful to them, and then share them publicly with tools like *Github* so that they can be useful to others. Often combining bits of several packages leads to **emergent properties** - completely new creations that can only occur because all of the parts (packages) are present. The collaborative nature of the R community, in this case on Twitter (follow the #rstats hashtag), can lead to surprising collaborations and outcomes.
See the example below.

# Chapter 4

# Updating R, RStudio, and Your Packages

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

# Chapter 5

# Major R Updates (Where Are My Packages?)

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

# Chapter 6

# Checking, Validating, And Asserting things about your Data

So you have imported your data! Great! Now to start the analysis!

Not so fast, cowboy!
First you need to validate your data.

#### 6.0.0.1 Cleaning – names with janitor package to snake_case

##### 6.0.0.1.1 A few words about tidyverse style

#### 6.0.0.2 Finding Missing data – naniar and visdat packages

#### 6.0.0.3 Validating data – validate package

### 6.0.1 Asserting properties of your data with assertr

#### 6.0.1.1 Evaluating – str, glimpse

#### 6.0.1.2 Exploring- skimr package

#### 6.0.1.3 Histograms

#### 6.0.1.4 Correlations – ggally extension of ggplot2, and corrr package

# Chapter 7

# Time Series data with the Tidyverts Packages

Fun text here. All kinds of crazy examples. Time series with data from influenza pandemic of 1918-19, perhaps. This is a book for anyone in the medical field interested in analyzing the data available to them to better understand health, disease, or delivery of care. This could include nurses, dieticians, psychologists, and PhDs in related fields, as well as medical students, residents, fellows, or doctors in practice.
I expect that most learners will be using this book in their spare time at night and on weekends, as the medical school curriculum is already packed full, and there is no room to add skills in reproducible research to the standard curriculum. This book is designed for self-teaching, and many hints and solutions will be provided to avoid roadblocks and frustration.

## 7.1 Tsibble

Time series tibble

Tidyverts webpage

## 7.2 Fable

Tidy forecasting

## 7.3 Feasts

Feature extraction and Statistics

## 7.4 Slider

Rolling anaylsis with window functions.
Slider packagedown page

# Chapter 8

# Descriptive Data Tables

In this chapter, we will focus on making the descriptive table of the participants in your study, often colloquially know as "Table One", based on its usual placement in a medical manuscript.
Before we plunge in, I would like to make one point of warning. It is quite common in a multiple-arm randomized controlled trial to compare the distribution of particular baseline characteristics of the subjects between arms with a p value, usually in a column at the far right. This is silly, as this produces a whole column of p values, corresponding to the multiple comparisons performed. With 20 comparisons, by chance, you are likely to get one or more "significant" p values. These are not helpful or meaningful, and are considered bad statistical practice.
Let me quote the CONSORT guidelines on the publications of clinical trials. *"Unfortunately significance tests of baseline differences are still common; they were reported in half of 50 RCTs published in leading general journals in 1997. Such significance tests assess the probability that observed baseline differences could have occurred by chance; however, we already know that any differences are caused by chance. Tests of baseline differences are not necessarily wrong, just illogical. Such hypothesis*
*testing is superfluous and can mislead investigators and their readers. Rather, comparisons at baseline should be based on consideration of the prognostic strength of the variables measured and the size >of any chance imbalances that have occurred."* CONSORT STATEMENT

Despite this, some journals and editors still ask for these p values. Please resist, and quote the CONSORT statement. If you must do this, please do it only under duress.

## 8.1 Making Table One

### 8.1.1 The *tableby* function in the arsenal package

### 8.1.2 The gtsummary package with flextable

This is a newer approach which offers many of the same features as tableby. The gtsummary package is a companion to/built upon the gt package, ("gt" for grammar of tables), which is supported by RStudio. The gtsummary package, like gt, is designed to produce nice html output with lots of nice formatting. However, as a nice bonus, gtsummary includes a neat function *as_flextable*, which converts your resulting table into a flextable, which can be knit to a Microsoft Word Document or a Powerpoint presentation with Rmarkdown. This means that you can make a table once, and be able to produce output in HTML for webpages, Microsoft Word for manuscripts, and MS Powerpoint for presentations from the same file without any conversion issues. The only question is how and when you prefer to format your table. Both gt and flextable have great options for formatting your tables. You can do this in gt, then do *as_flextable*, or you can convert to a flextable first, then do your formatting. You can choose based on your comfort and familiarity with flextable vs. gt. Both have excellent explanatory websites, with flextable here and gtsummary here.

## 8.2 Making An Adverse Events Table

## 8.3 Making A Results Table

# Chapter 9

# Comparing Two Measures of Centrality

A common question in medical research is whether one group had a better outcome than another group. These outcomes can be measured with dichotomous outcomes like death or hospitalization, but continuous outcomes like systolic blood pressure, endoscopic score, or ejection fraction are more commonly available, and provide more statistical power, and usually require a smaller sample size. There is a tendency in clinical research to focus on dichotomous outcomes, even to the point of converting continuous measures to dichotomous ones (aka "dichotomania", see Frank Harrell comments here), for fear of detecting and acting upon a small change in a continuous outcome that is not clinically meaningful. While this can be a concern, especially in very large, over-powered studies, it can be addressed by aiming for a continuous difference that is at least as large as one that many clinicians agree (*a priori*) is clinically important (the MCID, or Minimum Clinically Important Difference). The most common comparison of two groups with a continuous outcome is to look at the means or medians, and determine whether the available evidence suggests that these are equal (the null hypothesis). This can be done for means with Student's t-test. Let's start by looking at the cytomegalovirus data set. This includes data on 64 patients who received bone marrow stem cell transplant, and looks at their time to activation of CMV (cytomegalovirus). In the code chunk below, we group the data by donor cmv status (`donor.cmv`), and look at the mean time to CMV activation (`time.to.cmv` variable). Run the code (using the green arrow at the top right of the code chunk below) to see the difference in time to CMV activation in months between groups.

Try out some other grouping variables in the `group_by` statement, in place of `donor.cmv`. Consider variables like race, sex, and recipient.cmv. Edit the code and run it again with the green arrow at the top right.

```
## # A tibble: 2 x 2
##     sex mean_time2cmv
## * <dbl>         <dbl>
## 1     0          13.7
## 2     1          12.7
```

That seems like a big difference for donor.cmv, between 13.7303333333333 months and 12.7441176470588 months. And it makes theoretical sense that having a CMV positive donor is more likely to be associated with early activation of CMV in the recipient. But is it a significant difference, one that would be very unlikely to happen by chance? That depends on things like the number of people in each group, and the standard deviation in each group. That is the kind of question you can answer with a t-test, or for particularly skewed data like hospital length of stay or medical charges, a Wilcoxon test.

## 9.1 Common Problem

- Comparing two groups
  - Mean or median vs. expected
  - Two arms of study - independent
  - Pre and post / spouse and partner / left vs right arm – paired groups
- Are the means significantly different?
- Or the medians (if not normally distributed)?

### 9.1.1 How Skewed is Too Skewed?

- Formal test of normality = Shapiro-Wilk test
- Use base data set called ToothGrowth

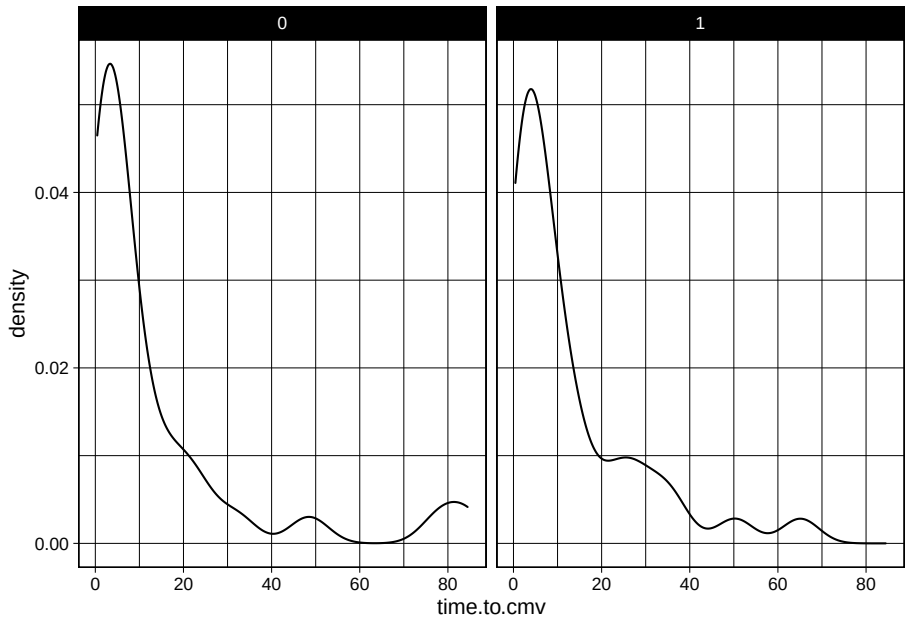
```
library(tidyverse)
data <- cytomegalovirus
head(data)
```

```
##   ID age sex race                   diagnosis
## 1  1  61   1    0       acute myeloid leukemia
## 2  2  62   1    1         non-Hodgkin lymphoma
## 3  3  63   0    1         non-Hodgkin lymphoma
## 4  4  33   0    1             Hodgkin lymphoma
## 5  5  54   0    1 acute lymphoblastic leukemia
## 6  6  55   1    1                myelofibrosis
##   diagnosis.type time.to.transplant prior.radiation
```

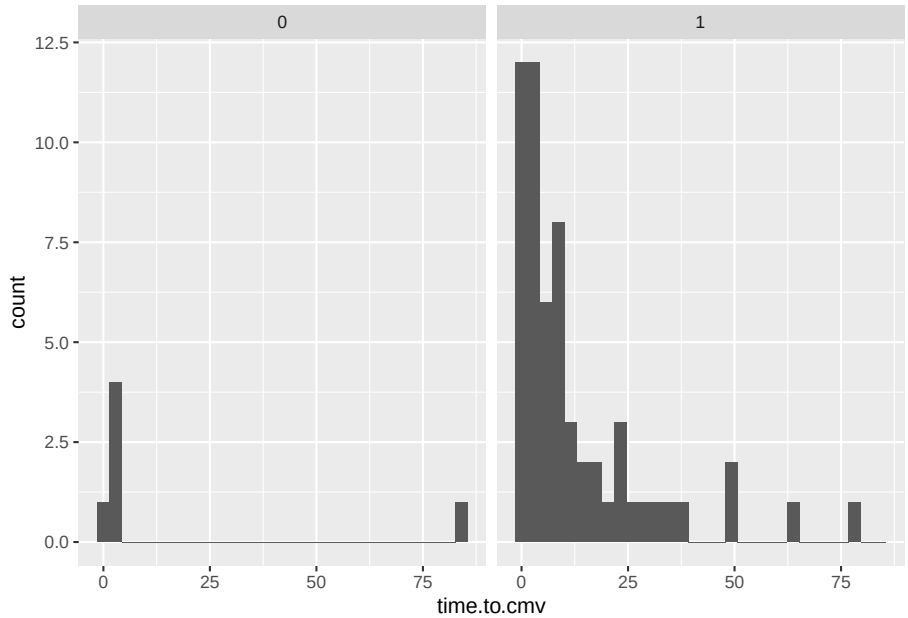
```
## 1              1               5.16               0
## 2              0              79.05               1
## 3              0              35.58               0
## 4              0              33.02               1
## 5              0              11.40               0
## 6              1               2.43               0
##   prior.chemo prior.transplant recipient.cmv donor.cmv
## 1           2                0             1         0
## 2           3                0             0         0
## 3           4                0             1         1
## 4           4                0             1         0
## 5           5                0             1         1
## 6           0                0             1         1
##   donor.sex TNC.dose CD34.dose CD3.dose CD8.dose TBI.dose
## 1         0    18.31      2.29     3.21     0.95      200
## 2         1     4.26      2.04       NA       NA      200
## 3         0     8.09      6.97     2.19     0.59      200
## 4         1    21.02      6.09     4.87     2.32      200
## 5         0    14.70      2.36     6.55     2.40      400
## 6         1     4.29      6.91     2.53     0.86      200
##   C1/C2 aKIRs cmv time.to.cmv agvhd time.to.agvhd cgvhd
## 1     0     1   1        3.91     1          3.55     0
## 2     1     5   0       65.12     0         65.12     0
## 3     0     3   0        3.75     0          3.75     0
## 4     0     2   0       48.49     1         28.55     1
## 5     0     6   0        4.37     1          2.79     0
## 6     0     2   1        4.53     1          3.88     0
##   time.to.cgvhd
## 1          6.28
## 2         65.12
## 3          3.75
## 4         10.45
## 5          4.37
## 6          6.87
```

### 9.1.2 Visualize the Distribution of data variables in ggplot

- Use geom_histogram or geom_density (pick one or the other)
- look at the distribution of CD3.dose or time.to.cmv
- Bonus points: facet by sex or race or donor.cmv
- Your turn to try it




```
data %>%
ggplot(mapping = aes(time.to.cmv)) +
  geom_histogram() +
  facet_wrap(~race)
```



### 9.1.3 Visualize the Distribution of data$len in ggplot

- The OJ group is left skewed
- May be problematic for using means
- formally test with Shapiro-Wilk

```
data$time.to.cmv %>%
shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.68261, p-value = 0.0000000001762
```

### 9.1.4 Results of Shapiro-Wilk

- p-value = 0.1091
- p not $< 0.05$
- Acceptably close to normal
- OK to compare means rather than medians
- can use t test rather than wilcoxon test
  - if p is $< 0.05$, use wilcoxon test
  - also known as Mann-Whitney test
  - a rank-based (non-parametric) test

### 9.1.5 Try it yourself

- use df <- msleep

```
df <- msleep
head(df$sleep_total)
```

```
## [1] 12.1 17.0 14.4 14.9  4.0 14.4
```

- test the normality of total sleep hours in mammals

### 9.1.6 Mammal sleep hours

```
shapiro.test(df$sleep_total)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$sleep_total
## W = 0.97973, p-value = 0.2143
```

- meets criteria - acceptable to consider normally distributed
- now consider - is the mean roughly 8 hours of sleep per day?

## 9.2 One Sample T test

- univariate test
  - Ho: mean is 8 hours
  - Ha: mean is not 8 hours
- can use t test because shapiro.test is NS

### 9.2.1 How to do One Sample T test

```
t.test(df$sleep_total, alternative = "two.sided",
       mu = 8)
```

- Try it out, see if you can interpret results

### 9.2.2 Interpreting the One Sample T test

```
##
##  One Sample t-test
##
## data:  df$sleep_total
## t = 4.9822, df = 82, p-value = 0.000003437
## alternative hypothesis: true mean is not equal to 8
## 95 percent confidence interval:
##   9.461972 11.405497
## sample estimates:
## mean of x
##  10.43373
```

- p is highly significant
  - can reject the null, accept alternative
  - sample mean 10.43, CI 9.46-11.41

### 9.2.3 What are the arguments of the t.test function?

- x = vector of continuous numerical data
- y= NULL - optional 2nd vector of continuous numerical data
- alternative = c("two.sided", "less", "greater"),
- mu = 0
- paired = FALSE
- var.equal = FALSE
- conf.level = 0.95
- documentation

## 9.3 Fine, but what about 2 groups?

- consider df$vore

```
table(df$vore)
```

```
##
##   carni   herbi insecti    omni
##      19      32       5      20
```

- hypothesis - herbivores need more time to get food, sleep less than carnivores
- how to test this?
  - normal, so can use t test for 2 groups

### 9.3.1 Setting up 2 group t test

- formula interface: outcome ~ groupvar

```
df %>%
  filter(vore %in% c("herbi", "carni")) %>%
  t.test(formula = sleep_total ~ vore, data = .)
```

- Try it yourself
- What do the results mean?

### 9.3.2 Results of the 2 group t test

```
##
##  Welch Two Sample t-test
##
## data:  sleep_total by vore
## t = 0.63232, df = 39.31, p-value = 0.5308
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.911365  3.650509
## sample estimates:
## mean in group carni mean in group herbi
##           10.378947            9.509375
```

### 9.3.3 Interpreting the 2 group t test

- Welch t-test (not Student)
  - Welch does NOT assume equal variances in each group
- p value NS
- accept null hypothesis
  - Ho: means of groups roughly equal
  - Ha: means are different
  - 95% CI crosses 0
- Carnivores sleep a little more, but not a lot

### 9.3.4 2 group t test with wide data

- You want to compare column A with column B (data are not tidy)
- Do mammals spend more time awake than asleep?

```
t.test(x = df$sleep_total, y = df$awake, data = msleep)
```

### 9.3.5 Results of 2 group t test with wide data

```
t.test(x = df$sleep_total, y = df$awake, data = msleep)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  df$sleep_total and df$awake
## t = -4.5353, df = 164, p-value = 0.00001106
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.498066 -1.769404
## sample estimates:
## mean of x mean of y
##  10.43373  13.56747
```

## 9.4 3 Assumptions of Student's t test

1. Sample is normally distributed (test with Shapiro)
2. Variances are homogeneous (homoskedasticity) (test with Levene)
3. Observations are independent

- not paired like left vs. right colon
- not paired like spouse and partner
- not paired like measurements pre and post Rx

### 9.4.1 Testing Assumptions of Student's t test

- Normality - test with Shapiro
  - If not normal, Wilcoxon > t test
- Equal Variances - test with Levene
  - If not equal, Welch t > Student's t
- Observations are independent
  - Think about data collection
  - are some observations correlated with some others?
  - If correlated, use paired t test

## 9.5 Getting results out of t.test

- Use the tidy function from the broom package
- Do carnivores have bigger brains than insectivores?

```
df %>%
  filter(vore %in% c("carni", "insecti")) %>%
t.test(formula = brainwt ~ vore, data = .) %>%
  tidy() ->
```

```
result
result
```

### 9.5.1 Getting results out of t.test

```
## # A tibble: 1 x 9
##   estimate1 estimate2 statistic p.value parameter conf.low
##       <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>
## 1    0.0793    0.0216      1.20   0.253        12  -0.0471
## # ... with 3 more variables: conf.high <dbl>, method <chr>,
## #   alternative <chr>
```

## 9.6 Reporting the results from t.test using inline code

- use backticks before and after, start with r
  - i.e. My result is [backtick]r code here[backtick].
- The mean brain weight for carnivores was 0.0792556
- The mean brain weight for herbivores was 0.02155
  - The difference was
- The t statistic for this Two Sample t-test was 1.1995501
- The p value was 0.2534631
  - The confidence interval was from -0.05 to 0.16

### 9.6.1 For Next Time

- Skewness and Kurtosis
- Review Normality
  - When to use Wilcoxon
- Levene test for equal variances
  - When to use Welch t vs. Student's t
- Paired t and Wilcoxon tests

# Title holder