Reproducible Medical Research with R

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

2020-04-24

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

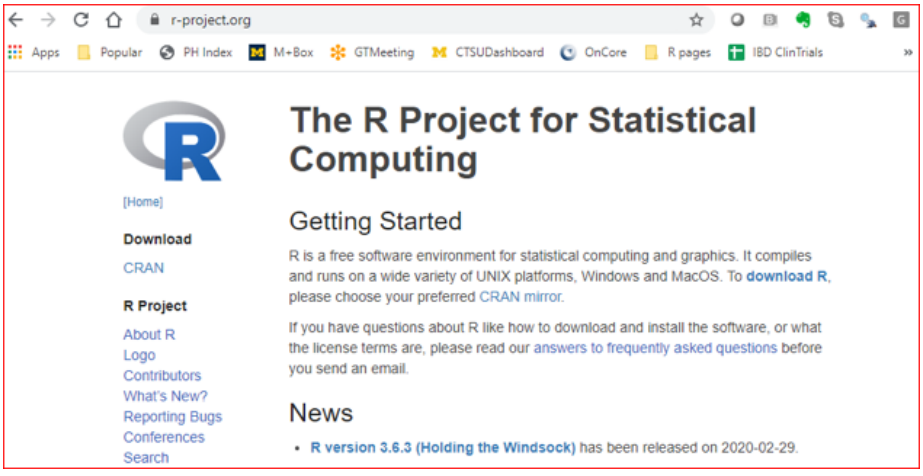
- Getting Started and Installing Your Tools
- Updating R, RStudio, and Your Packages
- Advanced Use of the RStudio IDE
- When You Don't Want to Update Packages (Using *renv*)
- Major R Updates (Where Are My Packages?)

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.

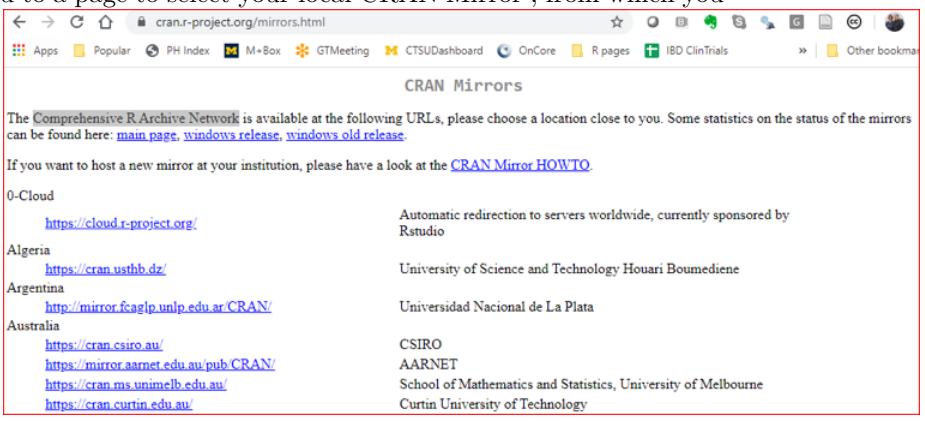
This is provided by the RStudio IDE (Integrated Developer Environment). We want you to install both R and RStudio, in that order.
Let's start with installing R.
R is free and available for download on the web. Go to the r-project website to get started.



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.

Now 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.

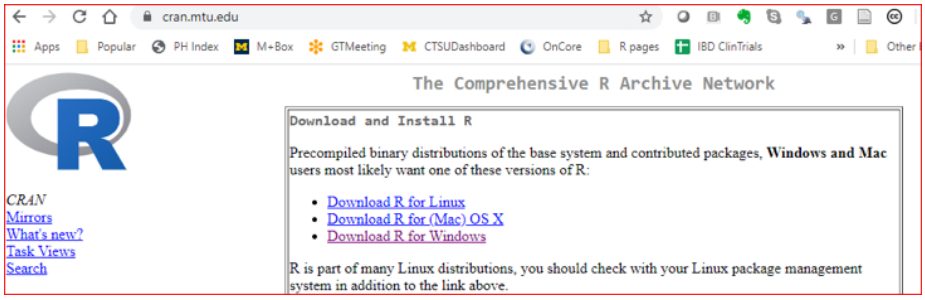


Scroll down to your country (yes, the USA is at the bottom), and a CRAN mirror near you. This is an example from northern Michigan, USA.

USA

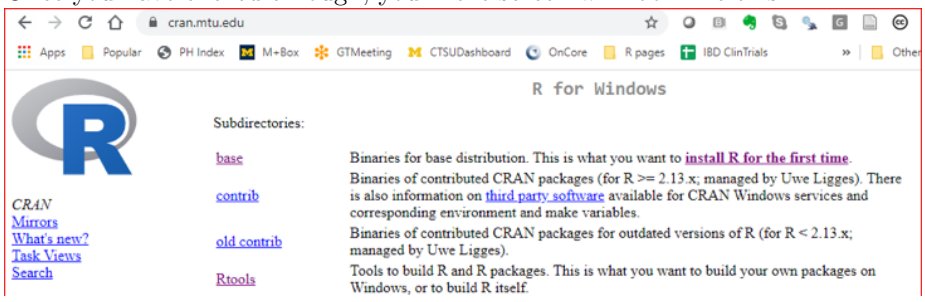
https://mirror.las.iastate.edu/CRAN/	Iowa State University, Ames, IA
https://ftp.ussg.iu.edu/CRAN/	Indiana University
https://rweb.crmda.ku.edu/cran/	University of Kansas, Lawrence, KS
https://cran.mtu.edu/	Michigan Technological University, Houghton, MI
https://repo.miserver.it.umich.edu/cran/	MBNI, 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.



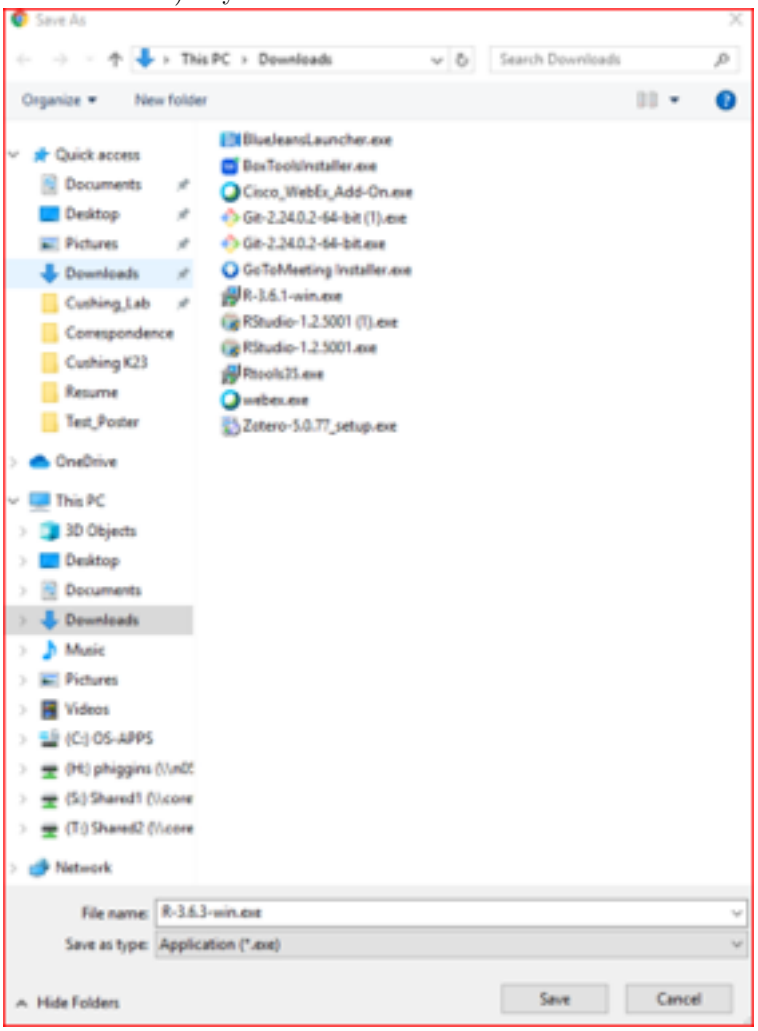
Select the link for the operating system you want to use. We will walk through this with Windows first, then Mac. If you are using a Mac, skip forward to the Mac install. If you are using Linux, you can clearly figure it out on your own (it will look a lot like these).

Once you have clicked through, your next screen will look like this

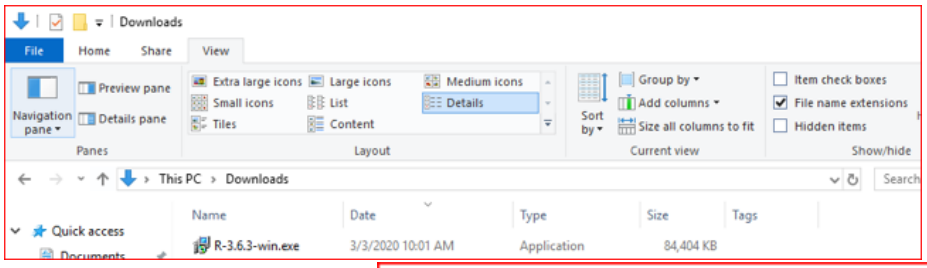


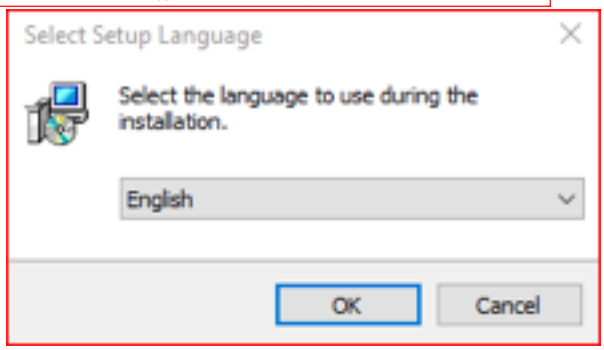
You want to download both base and Rtools (you might need Rtools later). The base link will take you to the latest version, which will look something like this.

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 button to save it.

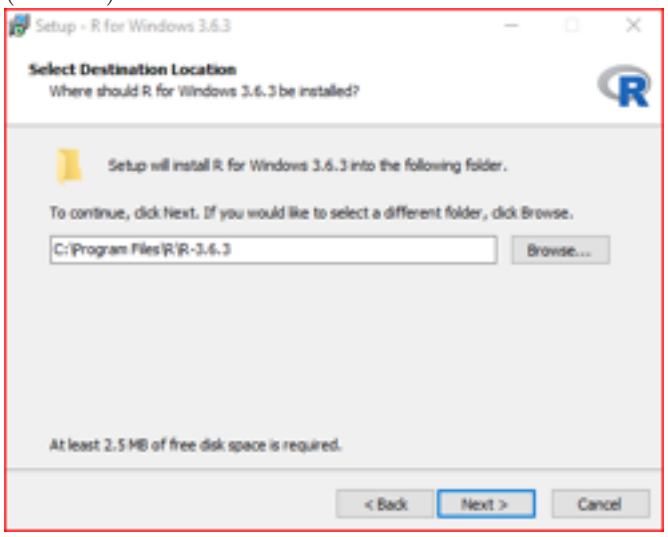


Now, go to your Downloads folder in Windows, and double clinic on the R installation file (R-N.N.N-win.exe). Click Yes to allow this to install.

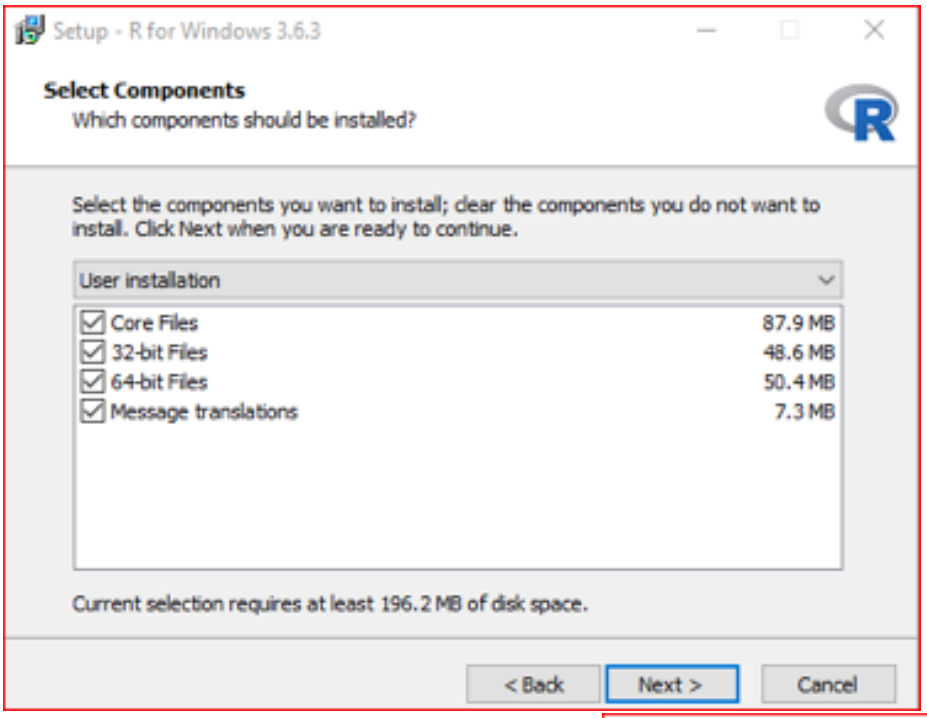




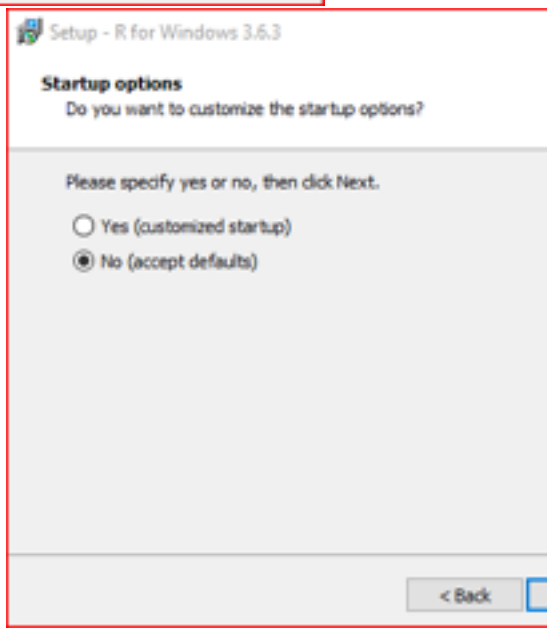
Now select your language option You will be asked to accept the GNU license - do so. Click Yes to allow this to install. Then select where to install - generally use the default- a local (often C) drive - do not install on a shared network drive or in the cloud.



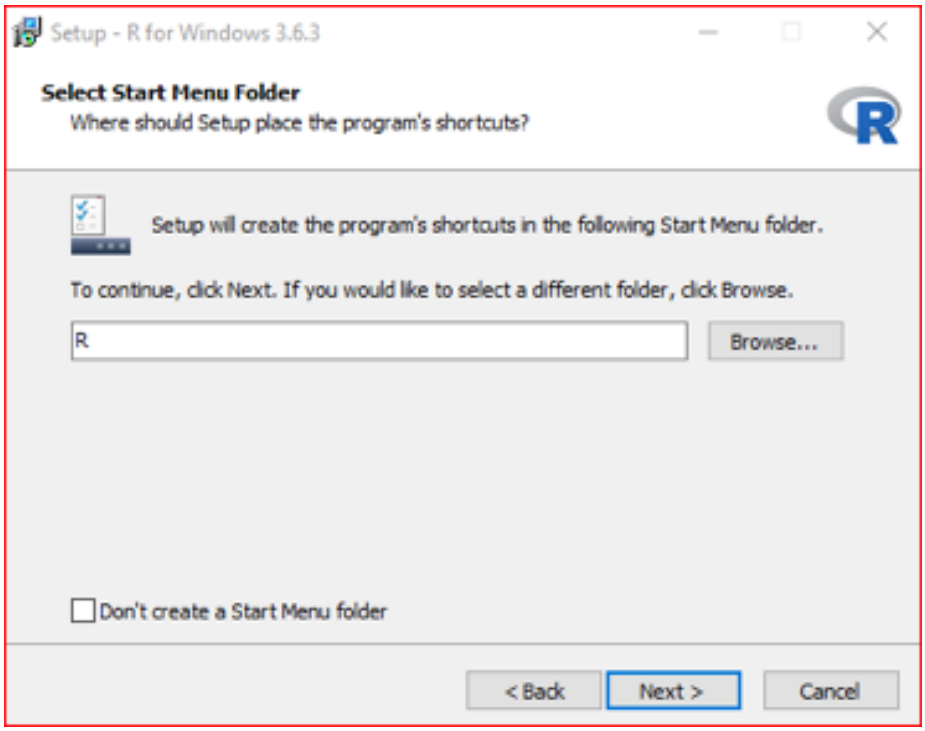
Then select the Components - generally use the defaults, but newer computers can skip the 32 bit ver-



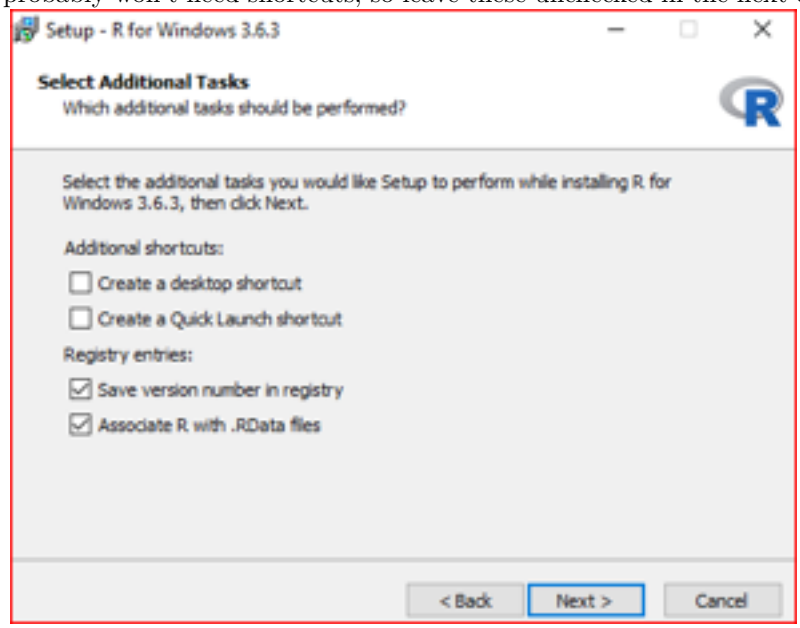
sion.



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.

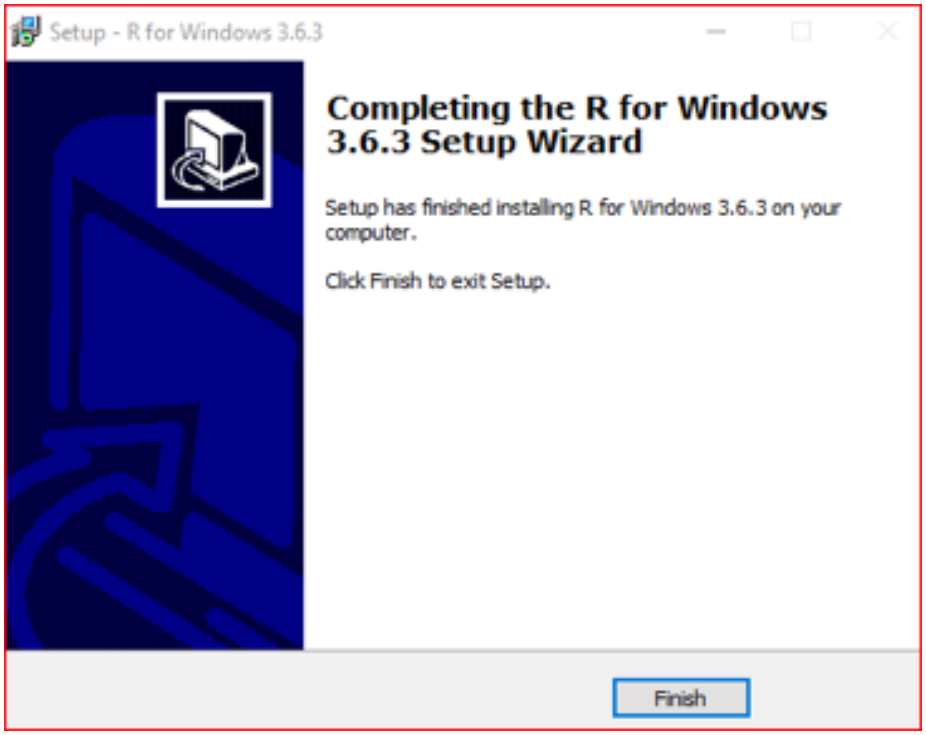
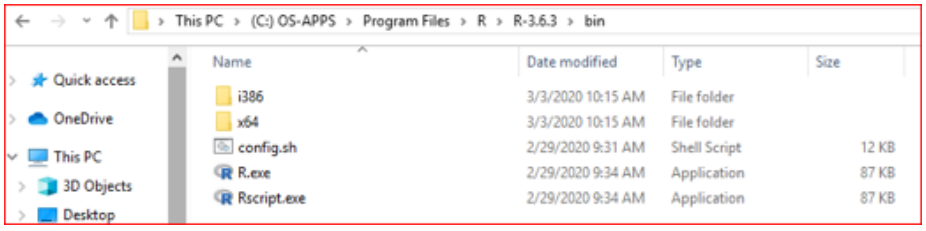


Figure 2.1: install_wizard

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

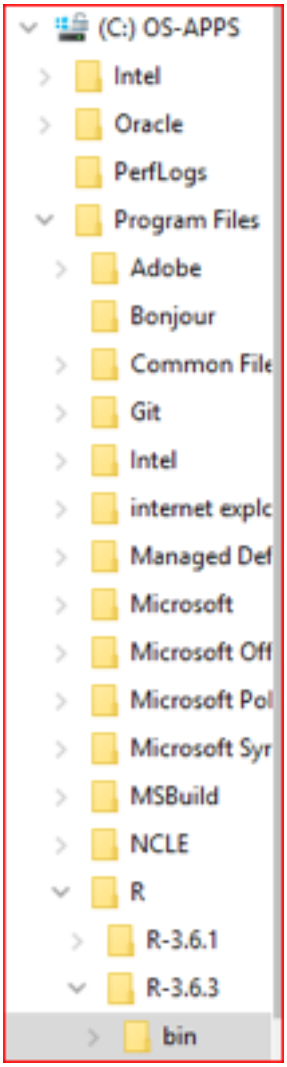
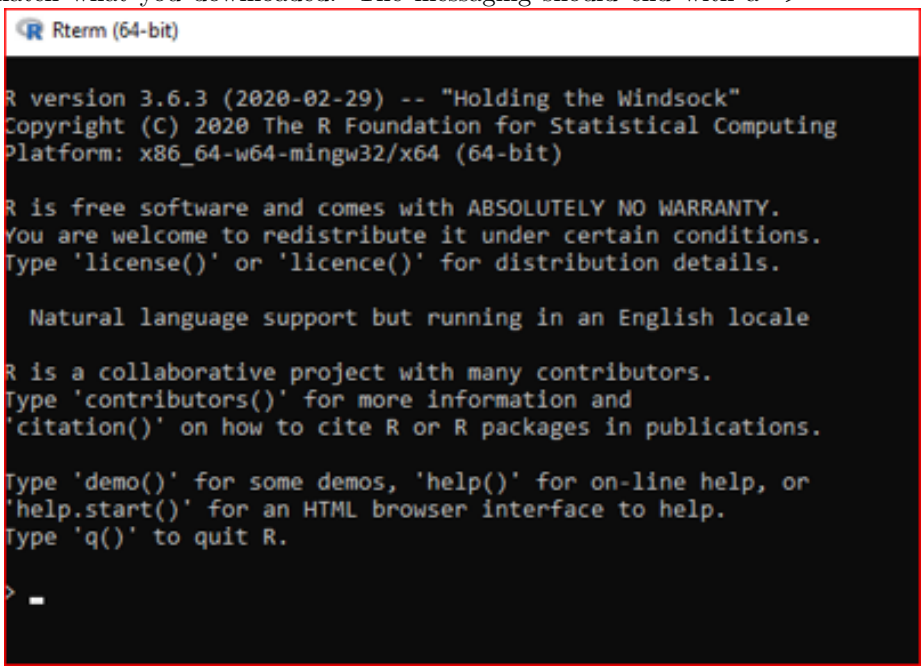


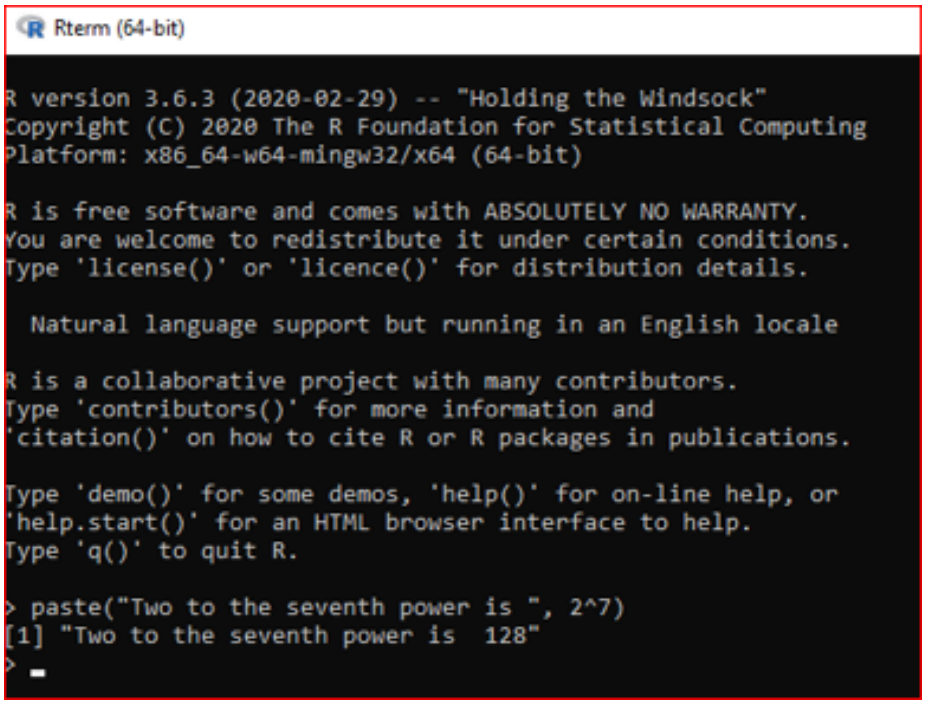
Figure 2.2: install_path2

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:

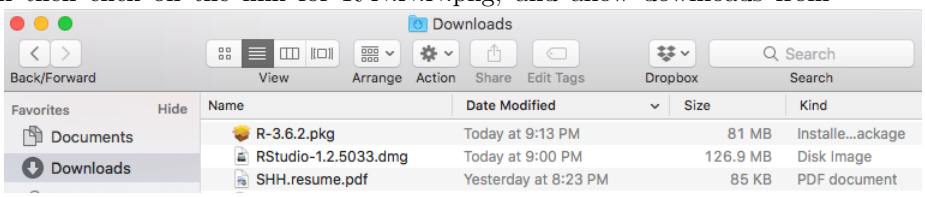


Note that you have explained what is being done and computed the result.

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

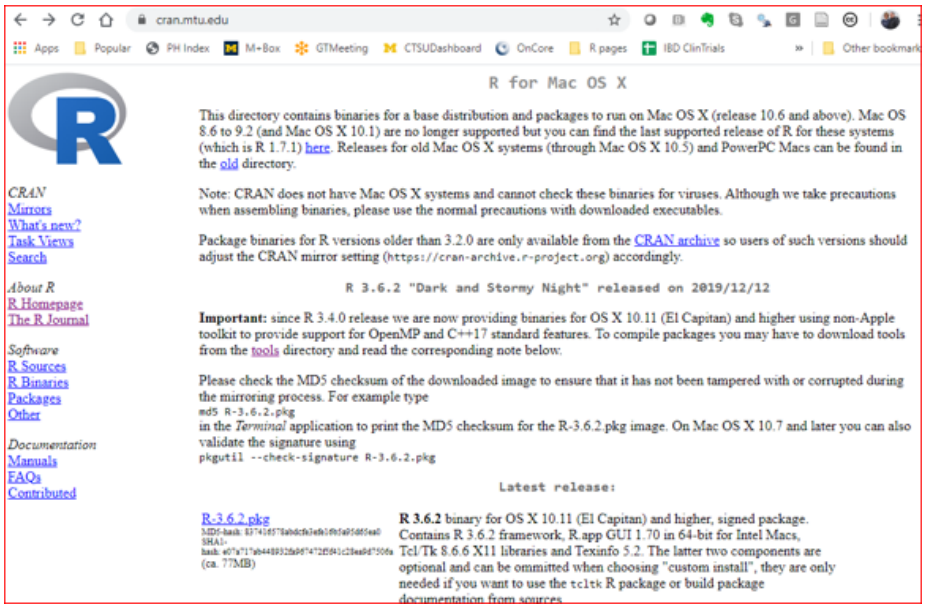
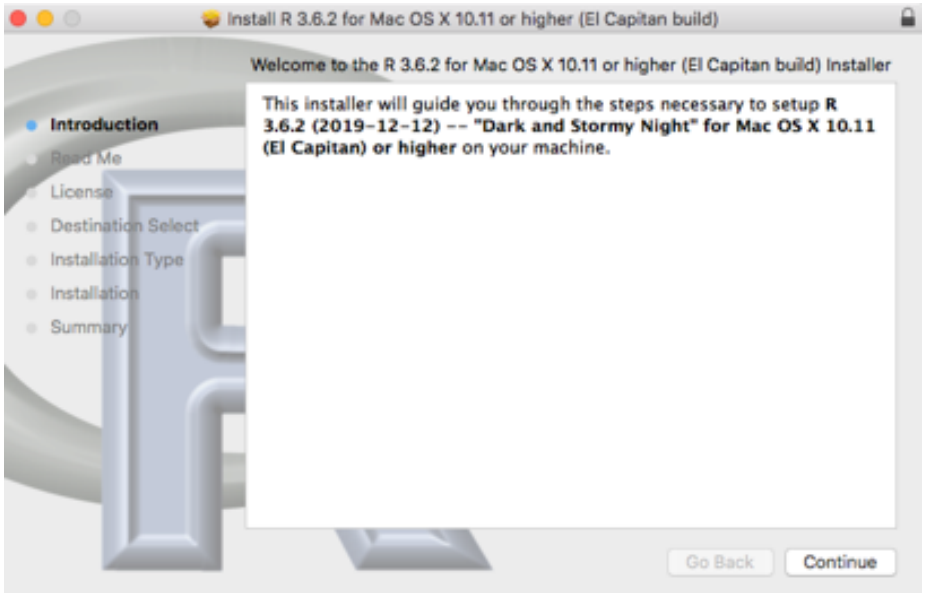
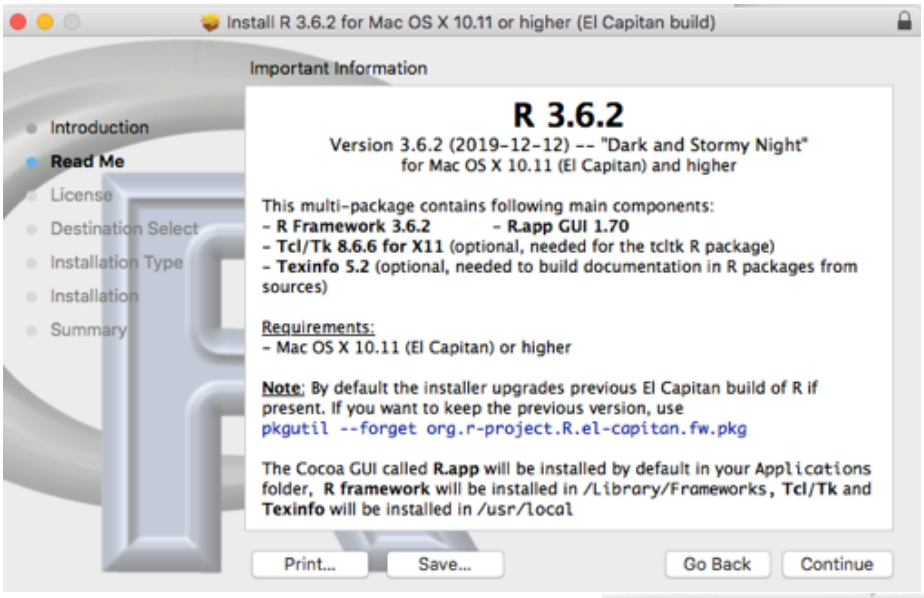
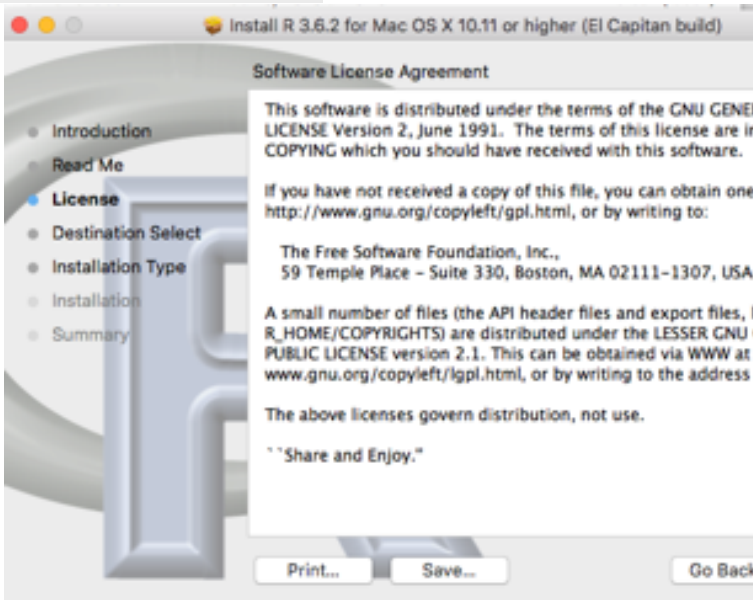


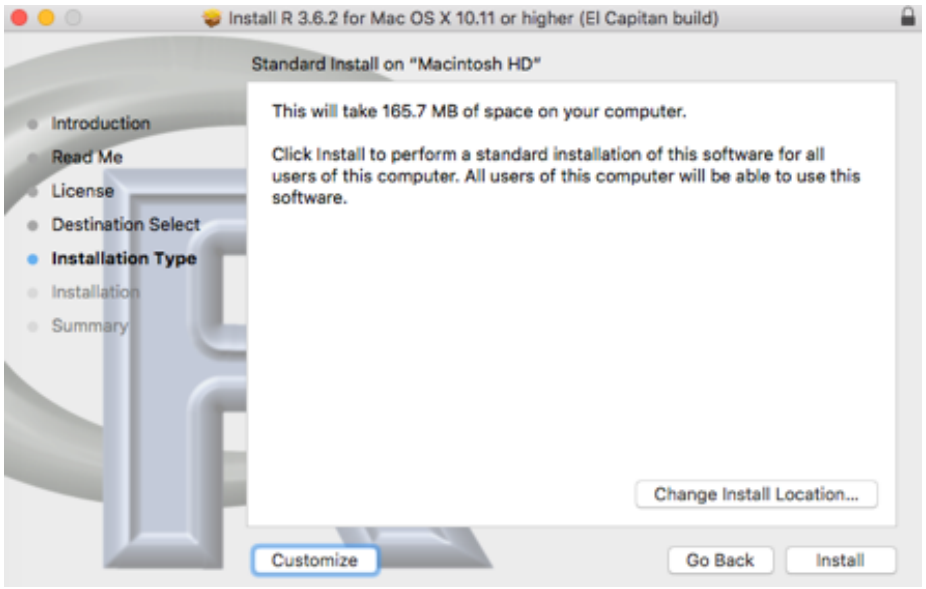
Figure 2.3: install_path







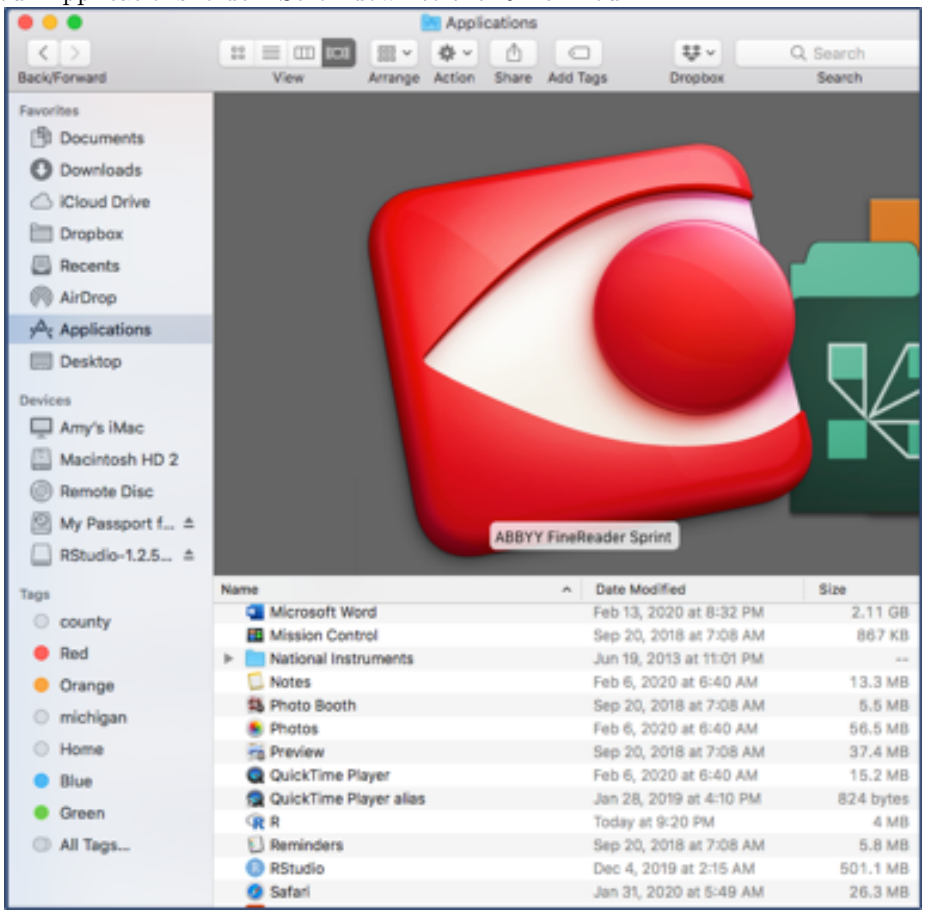
Then you need to agree with the License Agreement, then Click on Install, and provide your Mac password for permission to install.



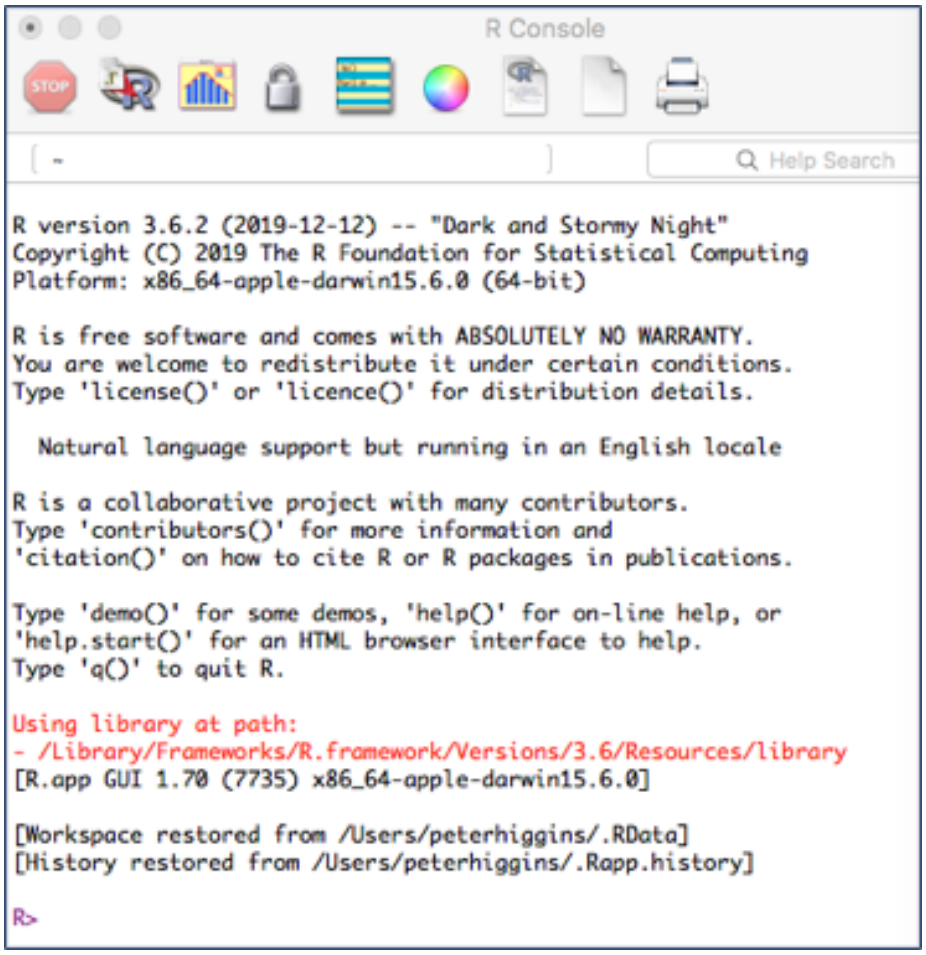
When the installation is complete, click on the Close button. Accept the prompt to move the installer file to the trash.

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)



This should result in

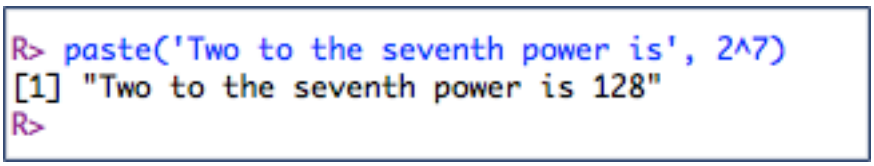


Figure 2.4: mactestR

2.4.4 Successful testing!

Awesome. You are now Ready to R!

2.5 Installing RStudio on your Computer

2.6 Installing Git on your Computer

2.7 Getting Acquainted with the RStudio IDE

2.8 Introduction

There are many books about Data Science. Why does the world need another one, particularly one targeting physicians?

- There is a lot of health care data
- There are a lot of interesting questions in health care
- There are particular and challenging issues in doing data analysis with PHI (Protected Health Information)

Syllabus: Data Science for Physicians (DS4P)

- Instructor: Peter Higgins, MD, PhD, MSc (CRDSA), Professor of Internal Medicine
- Office Hours: MSRB One 6510
- In-person class time
  - MSRB One 6510, Thursday evenings 6:30-8:30 PM

2.9 Course Description and Objectives

2.9.1 Description

A practical introduction to data collection and security, data cleaning, statistical methods and computational tools needed to make sense of data, and methods for reporting and sharing your findings. This course is not a traditional introductory statistics courses in that computing plays a more central role than mathematics and a higher emphasis is placed on "thinking with data." Topics include

- secure HIPPA-compliant data collection
- data cleaning and validation
- data visualization
- data wrangling
- confidence intervals
- hypothesis testing, and
- regression The course has no mathematics or computer science prerequisites.

2.9.2 Objectives

1. Have students engage in the data/science research pipeline in as faithful a manner as possible while maintaining a level suitable for novices.
2. Foster a conceptual understanding of statistical topics and methods using real clinical data whenever possible, and simulation/resampling to support teaching concepts of inference.
3. Use a flipped classroom model by incorporating online learning for new concepts, with limited face-to-face time for real-time problem-solving
4. Introduce best practices for reproducible research and collaboration.
5. Develop statistical literacy by, among other ways, tying in the curriculum to actual clinical data, demonstrating the importance statistics and computing plays in advancing medicine

2.9.3 Topics

Roughly speaking we will cover the following topics (a more detailed outline is found below:

1. Introduction and Tools (R, RStudio, and R Markdown)
2. Data Import, and Handling
3. Data Collection
4. Checking, Validating, And Exploring your Data
5. Data Types

6. Data Wrangling with Tidyr and Dplyr
7. Graphic Summaries for a Single Variable – ggplot package
8. Descriptive Data for a Single Variable
9. Graphic Summaries for Two or More Variables – ggplot2
10. Descriptive Data for Two or More Variables
11. Presenting your Results in a report with RMarkdown
12. Statistical inference
13. Study Design
14. Sample Size and Power
15. Sources of Bias
16. Study Types
17. One variable, single group
18. One variable, two groups
19. Multiple groups
20. Linear Regression
21. Reporting results interactively with Shiny
22. Logistic Regression
23. Meta-Analysis

2.9.4 Learning Resources

- E-Textbooks: Open Intro Statistics, at www.openintro.org
- E-Books on R These are at different levels:

Level: Absolute Beginner Textbook: R Basics
Goal: Set up R and RStudio on a laptop, introduce the concept of an IDE
Link:

Level: New to R & Statistics
Textbook: Modern Dive Goal: Learn basics of Data Management and visualization, introduction to hypothesis testing and statistical modeling
Link:

Level: Comfortable with R
Textbook: Hands-On R Programming
Goal:
Link:

Level: Ready to Understand More
Textbook: R for Data Science
Link:

- Software:
  - Local laptop/desktop free open-source version of R and RStudio

    - Cloud-based RStudio Server, which you can access in your browser via: Note if you are off-campus you must first log into the UM VPN.

- Online:

    - DataCamp. A brower based interactive tool for learning R through short, focused courses, each 3-4 hours long.
    - RStudio. Website with many resources for learning about the RStudio IDE and the tidyverse.
    - r-cookbook – an often useful website with concrete examples of how to use R packages
    - Stack Overflow and Google. Remarkably helpful to search for explanations of error messages, or explanations of problems that someone else has probably also experienced. For using Google, search for any topic or your error message and add "in R"
    - package vignettes – variable quality, but when well done, can be extremely helpful examples of how to use the functions in each package
    - R twitter – follow Rbloggers, #rstats

2.9.5 Evaluation

This course is entirely voluntary. I hope that you will learn valuable skills that will advance your research career. I would like you to progress to using these skills on your own data as quickly as possible, as this will greatly help you reinforce your new skills. There are no grades and no formal evaluations. You can, however, earn certificates on DataCamp for completing courses.

2.9.6 Task Goals

1. Learn concepts through Data Camp

    a. Multiple short courses to correspond with each unit

2. Test yourself with assignments in ModernDive

    a. Chapters corresponding to each unit

3. Three Challenges

    a. Clean data and perform descriptive data analysis on the biofire dataset

b. Clean data and model outcomes in the health satisfaction dataset, producing a final report
c. Use logistic regression to model dichotomous outcomes and produce a Shiny app to allow users to make predictions for future patients

4. Final Project There will be a final capstone project. This is an opportunity for you to use your statistics and data science skills developed during the challenges and perform your own start-to-finish data analysis project. The project will involving you addressing a scientific question by choosing a data set (or preferably, using one of your own), performing an analysis using the concepts and tools we have covered in this course, and writing a report. This can be done solo or with a partner.

2.9.7 Learning Goals

1. Recognize the importance of data collection, identify limitations in data collection methods, and determine how they affect the generalizability of your findings
2. Use statistical software (R) to summarize data numerically and visually, and to perform data analysis.
3. Have a conceptual understanding of statistical inference.
4. Apply estimation and testing methods to analyze single variables or the relationship between two variables in order to understand data relationships and make data-based conclusions.
5. Model numerical response variables and dichotomous response variables using a single explanatory variable or multiple explanatory variables in order to investigate relationships between variables.
6. Interpret results correctly, effectively, and in context without relying on statistical jargon.
7. Critique data-based claims and evaluate data-based decisions.

2.9.8 Tips for success

1. Read materials for each unit
2. Do Data Camp courses for each unit – usually around 1 chapter (1 hour) per day.
3. Do Data Camp daily practice on any day that you don't have time to do a full chapter
4. At end of each course, review material, take notes, copy/reproduce/save code on your laptop
5. Try new skills on your own data, or on one of the open data sets
6. Use RStudio and DataCamp Cheat sheets
7. Annotate your code to help 'future you' understand it.
8. Save and reuse your code for future projects

2.9.9 Expected work load

This course is entirely voluntary. It is expected that you have lots of clinical and/or research work to keep up with, along with the occasional call or night rotation. This is an investment in future skills to help your career. I recommend that you try to do up to one hour a day on most days, and on days when that is not realistic, to just do the 10 minutes of daily practice on DataCamp to keep the information fresh in your mind.
Other learning resources:

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter **??**. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

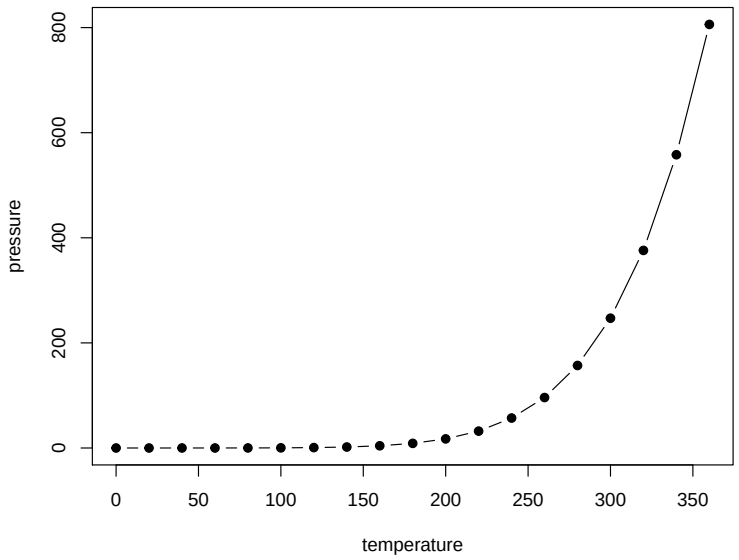


Figure 2.5: Here is a nice figure!

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure **??**. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table **??**.

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (**?**) in this sample book, which was built on top of R Markdown and **knitr** (**?**).

2.10 Access RStudio cloud

2.11 R basics E-book

Use the e-book Rbasics by Chester Ismay

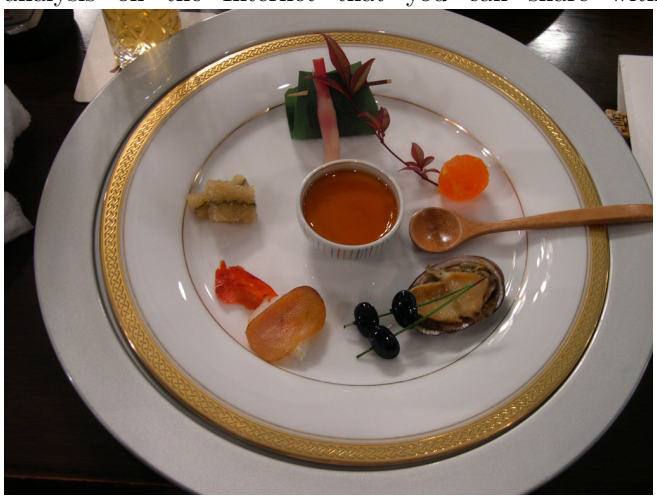
https://ismayc.github.io/rbasics-book/

2.12 RStudio tips document

Chapter 3

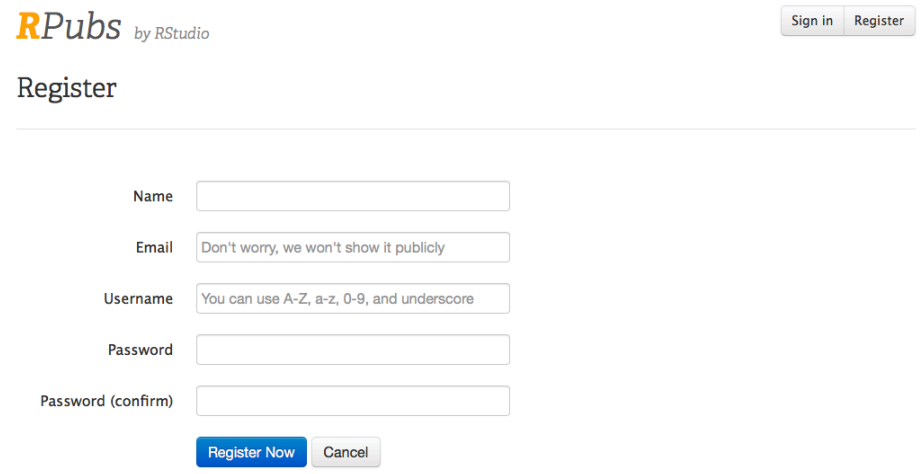
A Tasting Menu of R

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.

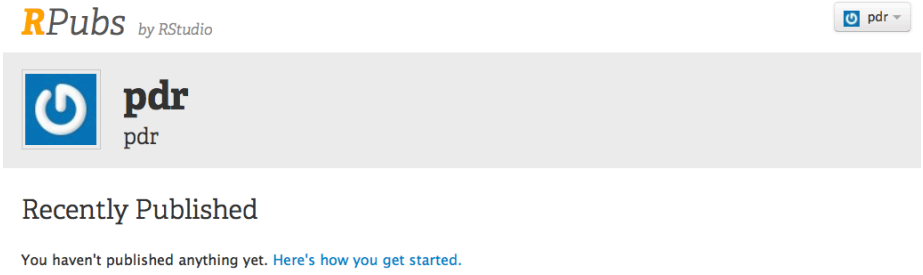


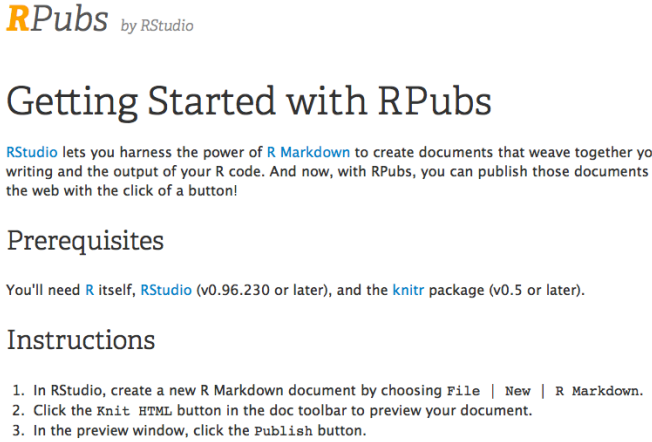
3.1 Setting the Table

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

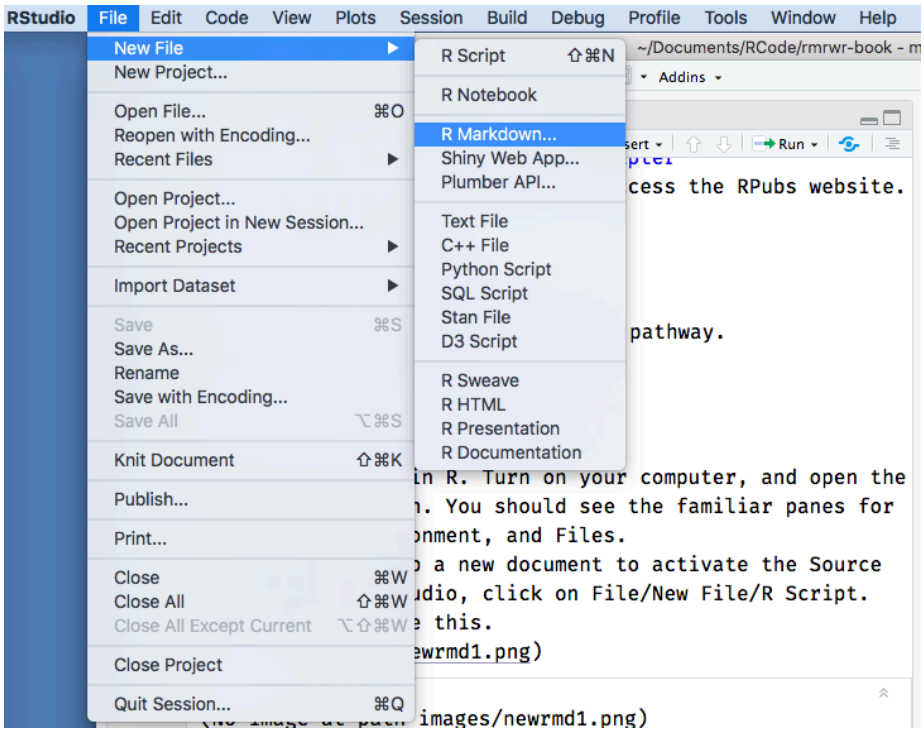
In this chapter, you will need to access the RPubs website.
- https://rpubs.com/

3.5 Pathway for this Chapter

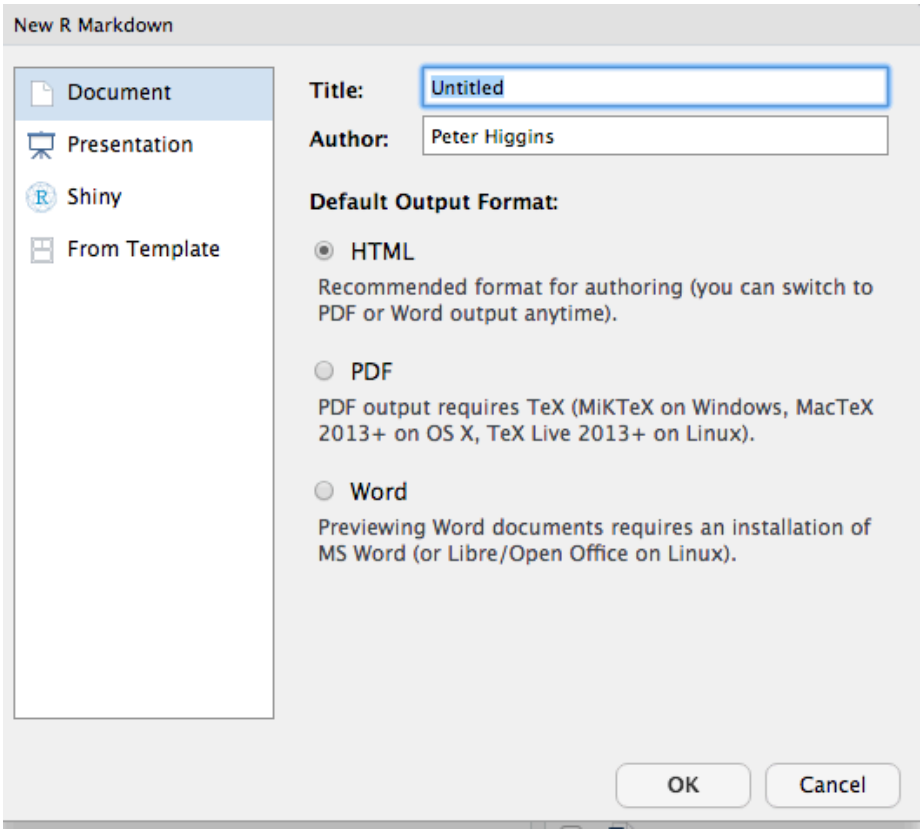
This Chapter is part of the **XXX** pathway. Chapters in this pathway include

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.

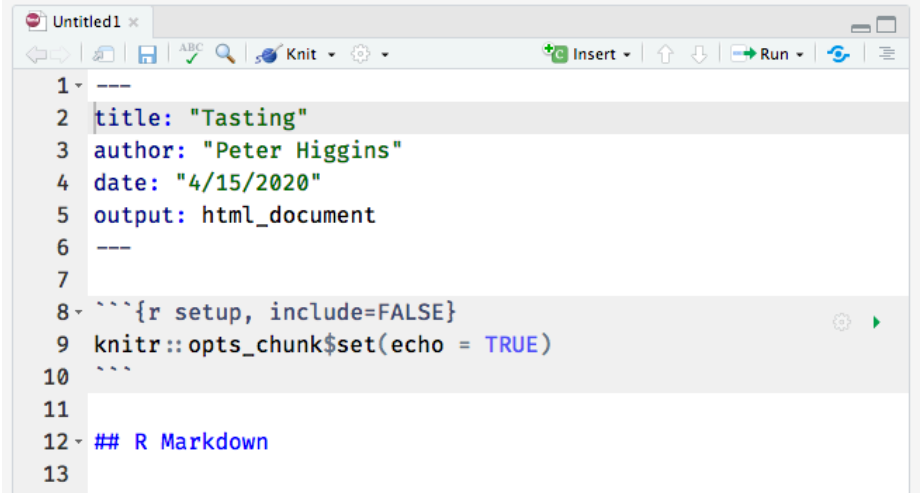


Now you will see the window below. Rename the document from “Untitled” to



"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.
Go ahead, *click* on the play button to run the gifs. All coded in R.

I used a modified version of **?**'s flametree package to produce this fractal tree in 3D, placed it in a LEGO planter for **?**, set down a copy of **?** and **?**'s book for some light reading, and of course, used #rayrender and #rstats to render it all :) https://t.co/qRg2omUftw pic.twitter.com/Ff9VeLtX97

— Tyler Morgan-Wall (**?**) April 14, 2020

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

8.1 Making Table One

8.1.1 Arsenal package, tableby

8.1.2 gtsummary

Chapter 9

Single variable GGplots

9.1 Histograms

9.1.1 dotplots

9.1.2 other?

Chapter 10

Two variable GGplots - continuous vs categorical

10.1 Boxplot

10.2 Violin plot

10.3 geom point or jitter

10.3.1 Overplotting - jitter and alpha

10.4 Ridgeplots

10.5 Challenges

Chapter 11

Two variable GGplots - continuous vs continuous

Chapter 12

Presenting your Results with Rmarkdown

The magic of the knit button

12.1 Making word document reports

12.1.1 setting up a custom template

12.2 Rmarkdown to powerpoint

12.3 Rmarkdown to HTML

12.4 Challenges

Chapter 13

Reproducibility in Your Research

13.0.0.1 Collaborating with Past You and Future You

13.0.0.1.1 General references

13.0.0.1.1.1 https://peerj.com/preprints/3192/

13.0.0.1.1.2 https://peerj.com/preprints/3139/

13.0.0.2 DataCamp GitHub Course

13.0.0.3 The problem of versions and updated packages

13.0.0.3.1 Solutions

13.0.0.3.1.1 Renv package

13.0.0.3.1.2 Rocker (docker) containers

13.0.0.4 R Projects

13.0.0.5 RStudio on Projects

13.0.0.5.1 Multiple scripts and organization of projects

13.0.0.5.2 Version control

13.0.0.5.2.1 https://peerj.com/preprints/3159/

13.0.0.6 Linear and branching projects, and use of the drake package

Chapter 14

Study Design

14.0.0.1 Hypotheses – null and alternative

14.0.0.2 Specific and testable

14.0.0.3 Inclusion and exclusion

14.0.0.4 How many eligible patients – DataDirect query

14.0.0.5 Be conservative – most won't enroll

14.1 Study Designs

14.1.0.1 Cross sectional

14.1.0.2 Cohort

14.1.0.3 Retrospective

14.1.0.4 Rarely prospective – registries, case series more likely

14.1.0.5 All Associations

14.1.0.6 Causal Inference requires randomization

Chapter 15

Sample Size and Power

Using the pwr package

15.0.0.1 Estimating measures in each group

15.0.0.2 Estimating effect size

15.0.0.3 Estimating range and SD

15.0.0.4 Power calc for continuous

15.0.0.5 Power calc for proportion

h <- pwr::ES.h(.14, .08) # effect size pwr::pwr.2p.test(h=0.1934809,n=350,sig.level=0.05,alternative="two.sided")

Chapter 16

Sources of Bias

16.0.1 Generalizability of studies

16.0.2 Bias in design

16.0.3 Recall Bias

16.0.4 Immortal Time bias

16.0.5 Ascertainment bias

16.0.6 Collider bias

16.0.7 Assuming linearity on a scale

This is an example of use of the blockrand package to create a randomization of two assignments with 5 strata in a randomized permuted block design

```
library(blockrand)
library(knitr)
library(kableExtra)
library(tidyverse, quietly = T)
```

```
## -- Attaching packages ------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0.9000      v purrr   0.3.4
## v tibble  3.0.1           v dplyr   0.8.99.9002
## v tidyr   1.0.2           v stringr 1.4.0
## v readr   1.3.1           v forcats 0.5.0
```

```
## -- Conflicts ---------------- tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
```

Our goal is to have 500 assignments available across 5 strata.

```
set.seed(1234)
stratum1 <- blockrand(n = 100,
                      block.prefix = "B_",
                      stratum = "Stratum_1",
                      num.levels = 2,
                      levels = c("Patient Navigator", "Control")
                      ) %>% filter(id <101)
stratum1
```

```
##    id   stratum block.id block.size         treatment
## 1   1 Stratum_1     B_01          8           Control
## 2   2 Stratum_1     B_01          8           Control
## 3   3 Stratum_1     B_01          8           Control
## 4   4 Stratum_1     B_01          8 Patient Navigator
## 5   5 Stratum_1     B_01          8           Control
## 6   6 Stratum_1     B_01          8 Patient Navigator
## 7   7 Stratum_1     B_01          8 Patient Navigator
## 8   8 Stratum_1     B_01          8 Patient Navigator
## 9   9 Stratum_1     B_02          4           Control
## 10 10 Stratum_1     B_02          4           Control
## 11 11 Stratum_1     B_02          4 Patient Navigator
## 12 12 Stratum_1     B_02          4 Patient Navigator
## 13 13 Stratum_1     B_03          4           Control
## 14 14 Stratum_1     B_03          4 Patient Navigator
## 15 15 Stratum_1     B_03          4           Control
## 16 16 Stratum_1     B_03          4 Patient Navigator
## 17 17 Stratum_1     B_04          4           Control
## 18 18 Stratum_1     B_04          4           Control
## 19 19 Stratum_1     B_04          4 Patient Navigator
## 20 20 Stratum_1     B_04          4 Patient Navigator
## 21 21 Stratum_1     B_05          2           Control
## 22 22 Stratum_1     B_05          2 Patient Navigator
## 23 23 Stratum_1     B_06          8 Patient Navigator
## 24 24 Stratum_1     B_06          8           Control
## 25 25 Stratum_1     B_06          8 Patient Navigator
## 26 26 Stratum_1     B_06          8           Control
## 27 27 Stratum_1     B_06          8 Patient Navigator
## 28 28 Stratum_1     B_06          8           Control
```

```
## 29 29 Stratum_1     B_06          8           Control
## 30 30 Stratum_1     B_06          8 Patient Navigator
## 31 31 Stratum_1     B_07          8 Patient Navigator
## 32 32 Stratum_1     B_07          8           Control
## 33 33 Stratum_1     B_07          8 Patient Navigator
## 34 34 Stratum_1     B_07          8           Control
## 35 35 Stratum_1     B_07          8 Patient Navigator
## 36 36 Stratum_1     B_07          8 Patient Navigator
## 37 37 Stratum_1     B_07          8           Control
## 38 38 Stratum_1     B_07          8           Control
## 39 39 Stratum_1     B_08          8           Control
## 40 40 Stratum_1     B_08          8           Control
## 41 41 Stratum_1     B_08          8 Patient Navigator
## 42 42 Stratum_1     B_08          8 Patient Navigator
## 43 43 Stratum_1     B_08          8 Patient Navigator
## 44 44 Stratum_1     B_08          8           Control
## 45 45 Stratum_1     B_08          8 Patient Navigator
## 46 46 Stratum_1     B_08          8           Control
## 47 47 Stratum_1     B_09          4           Control
## 48 48 Stratum_1     B_09          4 Patient Navigator
## 49 49 Stratum_1     B_09          4           Control
## 50 50 Stratum_1     B_09          4 Patient Navigator
## 51 51 Stratum_1     B_10          2           Control
## 52 52 Stratum_1     B_10          2 Patient Navigator
## 53 53 Stratum_1     B_11          2           Control
## 54 54 Stratum_1     B_11          2 Patient Navigator
## 55 55 Stratum_1     B_12          2 Patient Navigator
## 56 56 Stratum_1     B_12          2           Control
## 57 57 Stratum_1     B_13          8           Control
## 58 58 Stratum_1     B_13          8 Patient Navigator
## 59 59 Stratum_1     B_13          8           Control
## 60 60 Stratum_1     B_13          8 Patient Navigator
## 61 61 Stratum_1     B_13          8 Patient Navigator
## 62 62 Stratum_1     B_13          8 Patient Navigator
## 63 63 Stratum_1     B_13          8           Control
## 64 64 Stratum_1     B_13          8           Control
## 65 65 Stratum_1     B_14          4 Patient Navigator
## 66 66 Stratum_1     B_14          4           Control
## 67 67 Stratum_1     B_14          4           Control
## 68 68 Stratum_1     B_14          4 Patient Navigator
## 69 69 Stratum_1     B_15          2 Patient Navigator
## 70 70 Stratum_1     B_15          2           Control
## 71 71 Stratum_1     B_16          4 Patient Navigator
## 72 72 Stratum_1     B_16          4           Control
## 73 73 Stratum_1     B_16          4           Control
## 74 74 Stratum_1     B_16          4 Patient Navigator
```

```
## 75 75 Stratum_1     B_17          8           Control
## 76 76 Stratum_1     B_17          8 Patient Navigator
## 77 77 Stratum_1     B_17          8 Patient Navigator
## 78 78 Stratum_1     B_17          8 Patient Navigator
## 79 79 Stratum_1     B_17          8           Control
## 80 80 Stratum_1     B_17          8           Control
##  [ reached 'max' / getOption("max.print") -- omitted 20 rows ]
```

Now stratum 2

```
set.seed(0714)
stratum2 <- blockrand(n = 100,
                      block.prefix = "B_",
                      stratum = "Stratum_2",
                      num.levels = 2,
                      levels = c("Patient Navigator", "Control")
                      ) %>% filter(id <101)
stratum2
```

```
##    id   stratum block.id block.size         treatment
## 1   1 Stratum_2     B_01          6 Patient Navigator
## 2   2 Stratum_2     B_01          6 Patient Navigator
## 3   3 Stratum_2     B_01          6           Control
## 4   4 Stratum_2     B_01          6           Control
## 5   5 Stratum_2     B_01          6 Patient Navigator
## 6   6 Stratum_2     B_01          6           Control
## 7   7 Stratum_2     B_02          4           Control
## 8   8 Stratum_2     B_02          4 Patient Navigator
## 9   9 Stratum_2     B_02          4           Control
## 10 10 Stratum_2     B_02          4 Patient Navigator
## 11 11 Stratum_2     B_03          4           Control
## 12 12 Stratum_2     B_03          4 Patient Navigator
## 13 13 Stratum_2     B_03          4 Patient Navigator
## 14 14 Stratum_2     B_03          4           Control
## 15 15 Stratum_2     B_04          4           Control
## 16 16 Stratum_2     B_04          4           Control
## 17 17 Stratum_2     B_04          4 Patient Navigator
## 18 18 Stratum_2     B_04          4 Patient Navigator
## 19 19 Stratum_2     B_05          4           Control
## 20 20 Stratum_2     B_05          4           Control
## 21 21 Stratum_2     B_05          4 Patient Navigator
## 22 22 Stratum_2     B_05          4 Patient Navigator
## 23 23 Stratum_2     B_06          8 Patient Navigator
## 24 24 Stratum_2     B_06          8 Patient Navigator
## 25 25 Stratum_2     B_06          8 Patient Navigator
```

```
## 26 26 Stratum_2     B_06          8           Control
## 27 27 Stratum_2     B_06          8           Control
## 28 28 Stratum_2     B_06          8 Patient Navigator
## 29 29 Stratum_2     B_06          8           Control
## 30 30 Stratum_2     B_06          8           Control
## 31 31 Stratum_2     B_07          2 Patient Navigator
## 32 32 Stratum_2     B_07          2           Control
## 33 33 Stratum_2     B_08          2 Patient Navigator
## 34 34 Stratum_2     B_08          2           Control
## 35 35 Stratum_2     B_09          6 Patient Navigator
## 36 36 Stratum_2     B_09          6           Control
## 37 37 Stratum_2     B_09          6           Control
## 38 38 Stratum_2     B_09          6 Patient Navigator
## 39 39 Stratum_2     B_09          6 Patient Navigator
## 40 40 Stratum_2     B_09          6           Control
## 41 41 Stratum_2     B_10          4 Patient Navigator
## 42 42 Stratum_2     B_10          4           Control
## 43 43 Stratum_2     B_10          4 Patient Navigator
## 44 44 Stratum_2     B_10          4           Control
## 45 45 Stratum_2     B_11          2           Control
## 46 46 Stratum_2     B_11          2 Patient Navigator
## 47 47 Stratum_2     B_12          4 Patient Navigator
## 48 48 Stratum_2     B_12          4           Control
## 49 49 Stratum_2     B_12          4 Patient Navigator
## 50 50 Stratum_2     B_12          4           Control
## 51 51 Stratum_2     B_13          6           Control
## 52 52 Stratum_2     B_13          6           Control
## 53 53 Stratum_2     B_13          6           Control
## 54 54 Stratum_2     B_13          6 Patient Navigator
## 55 55 Stratum_2     B_13          6 Patient Navigator
## 56 56 Stratum_2     B_13          6 Patient Navigator
## 57 57 Stratum_2     B_14          8 Patient Navigator
## 58 58 Stratum_2     B_14          8           Control
## 59 59 Stratum_2     B_14          8 Patient Navigator
## 60 60 Stratum_2     B_14          8           Control
## 61 61 Stratum_2     B_14          8           Control
## 62 62 Stratum_2     B_14          8 Patient Navigator
## 63 63 Stratum_2     B_14          8           Control
## 64 64 Stratum_2     B_14          8 Patient Navigator
## 65 65 Stratum_2     B_15          2 Patient Navigator
## 66 66 Stratum_2     B_15          2           Control
## 67 67 Stratum_2     B_16          2 Patient Navigator
## 68 68 Stratum_2     B_16          2           Control
## 69 69 Stratum_2     B_17          2           Control
## 70 70 Stratum_2     B_17          2 Patient Navigator
## 71 71 Stratum_2     B_18          4 Patient Navigator
```

```
## 72 72 Stratum_2     B_18          4 Patient Navigator
## 73 73 Stratum_2     B_18          4           Control
## 74 74 Stratum_2     B_18          4           Control
## 75 75 Stratum_2     B_19          6           Control
## 76 76 Stratum_2     B_19          6 Patient Navigator
## 77 77 Stratum_2     B_19          6           Control
## 78 78 Stratum_2     B_19          6 Patient Navigator
## 79 79 Stratum_2     B_19          6 Patient Navigator
## 80 80 Stratum_2     B_19          6           Control
##  [ reached 'max' / getOption("max.print") -- omitted 20 rows ]
```

Now stratum 3

```
set.seed(1965)
stratum3 <- blockrand(n = 100,
                      block.prefix = "B_",
                      stratum = "Stratum_3",
                      num.levels = 2,
                      levels = c("Patient Navigator", "Control")
                      ) %>% filter(id <101)
stratum3
```

```
##    id   stratum block.id block.size         treatment
## 1   1 Stratum_3     B_01          8           Control
## 2   2 Stratum_3     B_01          8 Patient Navigator
## 3   3 Stratum_3     B_01          8 Patient Navigator
## 4   4 Stratum_3     B_01          8           Control
## 5   5 Stratum_3     B_01          8           Control
## 6   6 Stratum_3     B_01          8           Control
## 7   7 Stratum_3     B_01          8 Patient Navigator
## 8   8 Stratum_3     B_01          8 Patient Navigator
## 9   9 Stratum_3     B_02          4           Control
## 10 10 Stratum_3     B_02          4 Patient Navigator
## 11 11 Stratum_3     B_02          4 Patient Navigator
## 12 12 Stratum_3     B_02          4           Control
## 13 13 Stratum_3     B_03          6 Patient Navigator
## 14 14 Stratum_3     B_03          6 Patient Navigator
## 15 15 Stratum_3     B_03          6           Control
## 16 16 Stratum_3     B_03          6 Patient Navigator
## 17 17 Stratum_3     B_03          6           Control
## 18 18 Stratum_3     B_03          6           Control
## 19 19 Stratum_3     B_04          2           Control
## 20 20 Stratum_3     B_04          2 Patient Navigator
## 21 21 Stratum_3     B_05          2           Control
## 22 22 Stratum_3     B_05          2 Patient Navigator
```

```
## 23 23 Stratum_3     B_06          6           Control
## 24 24 Stratum_3     B_06          6           Control
## 25 25 Stratum_3     B_06          6 Patient Navigator
## 26 26 Stratum_3     B_06          6           Control
## 27 27 Stratum_3     B_06          6 Patient Navigator
## 28 28 Stratum_3     B_06          6 Patient Navigator
## 29 29 Stratum_3     B_07          6 Patient Navigator
## 30 30 Stratum_3     B_07          6           Control
## 31 31 Stratum_3     B_07          6           Control
## 32 32 Stratum_3     B_07          6           Control
## 33 33 Stratum_3     B_07          6 Patient Navigator
## 34 34 Stratum_3     B_07          6 Patient Navigator
## 35 35 Stratum_3     B_08          8           Control
## 36 36 Stratum_3     B_08          8           Control
## 37 37 Stratum_3     B_08          8           Control
## 38 38 Stratum_3     B_08          8           Control
## 39 39 Stratum_3     B_08          8 Patient Navigator
## 40 40 Stratum_3     B_08          8 Patient Navigator
## 41 41 Stratum_3     B_08          8 Patient Navigator
## 42 42 Stratum_3     B_08          8 Patient Navigator
## 43 43 Stratum_3     B_09          6 Patient Navigator
## 44 44 Stratum_3     B_09          6           Control
## 45 45 Stratum_3     B_09          6           Control
## 46 46 Stratum_3     B_09          6 Patient Navigator
## 47 47 Stratum_3     B_09          6           Control
## 48 48 Stratum_3     B_09          6 Patient Navigator
## 49 49 Stratum_3     B_10          6           Control
## 50 50 Stratum_3     B_10          6 Patient Navigator
## 51 51 Stratum_3     B_10          6           Control
## 52 52 Stratum_3     B_10          6           Control
## 53 53 Stratum_3     B_10          6 Patient Navigator
## 54 54 Stratum_3     B_10          6 Patient Navigator
## 55 55 Stratum_3     B_11          8 Patient Navigator
## 56 56 Stratum_3     B_11          8 Patient Navigator
## 57 57 Stratum_3     B_11          8           Control
## 58 58 Stratum_3     B_11          8           Control
## 59 59 Stratum_3     B_11          8 Patient Navigator
## 60 60 Stratum_3     B_11          8           Control
## 61 61 Stratum_3     B_11          8 Patient Navigator
## 62 62 Stratum_3     B_11          8           Control
## 63 63 Stratum_3     B_12          6           Control
## 64 64 Stratum_3     B_12          6           Control
## 65 65 Stratum_3     B_12          6 Patient Navigator
## 66 66 Stratum_3     B_12          6 Patient Navigator
## 67 67 Stratum_3     B_12          6 Patient Navigator
## 68 68 Stratum_3     B_12          6           Control
```

```
## 69 69 Stratum_3     B_13          4 Patient Navigator
## 70 70 Stratum_3     B_13          4 Patient Navigator
## 71 71 Stratum_3     B_13          4           Control
## 72 72 Stratum_3     B_13          4           Control
## 73 73 Stratum_3     B_14          2 Patient Navigator
## 74 74 Stratum_3     B_14          2           Control
## 75 75 Stratum_3     B_15          6           Control
## 76 76 Stratum_3     B_15          6           Control
## 77 77 Stratum_3     B_15          6 Patient Navigator
## 78 78 Stratum_3     B_15          6 Patient Navigator
## 79 79 Stratum_3     B_15          6 Patient Navigator
## 80 80 Stratum_3     B_15          6           Control
##  [ reached 'max' / getOption("max.print") -- omitted 20 rows ]
```

Now stratum 4

```
set.seed(314159)
stratum4 <- blockrand(n = 100,
                      block.prefix = "B_",
                      stratum = "Stratum_4",
                      num.levels = 2,
                      levels = c("Patient Navigator", "Control")
                      ) %>% filter(id <101)
stratum4
```

```
##    id   stratum block.id block.size         treatment
## 1   1 Stratum_4     B_01          4           Control
## 2   2 Stratum_4     B_01          4 Patient Navigator
## 3   3 Stratum_4     B_01          4 Patient Navigator
## 4   4 Stratum_4     B_01          4           Control
## 5   5 Stratum_4     B_02          2           Control
## 6   6 Stratum_4     B_02          2 Patient Navigator
## 7   7 Stratum_4     B_03          2 Patient Navigator
## 8   8 Stratum_4     B_03          2           Control
## 9   9 Stratum_4     B_04          6           Control
## 10 10 Stratum_4     B_04          6 Patient Navigator
## 11 11 Stratum_4     B_04          6 Patient Navigator
## 12 12 Stratum_4     B_04          6           Control
## 13 13 Stratum_4     B_04          6 Patient Navigator
## 14 14 Stratum_4     B_04          6           Control
## 15 15 Stratum_4     B_05          2 Patient Navigator
## 16 16 Stratum_4     B_05          2           Control
## 17 17 Stratum_4     B_06          2 Patient Navigator
## 18 18 Stratum_4     B_06          2           Control
## 19 19 Stratum_4     B_07          4 Patient Navigator
```

```
## 20 20 Stratum_4     B_07          4           Control
## 21 21 Stratum_4     B_07          4 Patient Navigator
## 22 22 Stratum_4     B_07          4           Control
## 23 23 Stratum_4     B_08          2 Patient Navigator
## 24 24 Stratum_4     B_08          2           Control
## 25 25 Stratum_4     B_09          4           Control
## 26 26 Stratum_4     B_09          4 Patient Navigator
## 27 27 Stratum_4     B_09          4           Control
## 28 28 Stratum_4     B_09          4 Patient Navigator
## 29 29 Stratum_4     B_10          2 Patient Navigator
## 30 30 Stratum_4     B_10          2           Control
## 31 31 Stratum_4     B_11          6           Control
## 32 32 Stratum_4     B_11          6 Patient Navigator
## 33 33 Stratum_4     B_11          6           Control
## 34 34 Stratum_4     B_11          6 Patient Navigator
## 35 35 Stratum_4     B_11          6           Control
## 36 36 Stratum_4     B_11          6 Patient Navigator
## 37 37 Stratum_4     B_12          8           Control
## 38 38 Stratum_4     B_12          8 Patient Navigator
## 39 39 Stratum_4     B_12          8           Control
## 40 40 Stratum_4     B_12          8           Control
## 41 41 Stratum_4     B_12          8 Patient Navigator
## 42 42 Stratum_4     B_12          8           Control
## 43 43 Stratum_4     B_12          8 Patient Navigator
## 44 44 Stratum_4     B_12          8 Patient Navigator
## 45 45 Stratum_4     B_13          4 Patient Navigator
## 46 46 Stratum_4     B_13          4 Patient Navigator
## 47 47 Stratum_4     B_13          4           Control
## 48 48 Stratum_4     B_13          4           Control
## 49 49 Stratum_4     B_14          4 Patient Navigator
## 50 50 Stratum_4     B_14          4 Patient Navigator
## 51 51 Stratum_4     B_14          4           Control
## 52 52 Stratum_4     B_14          4           Control
## 53 53 Stratum_4     B_15          4 Patient Navigator
## 54 54 Stratum_4     B_15          4 Patient Navigator
## 55 55 Stratum_4     B_15          4           Control
## 56 56 Stratum_4     B_15          4           Control
## 57 57 Stratum_4     B_16          4 Patient Navigator
## 58 58 Stratum_4     B_16          4           Control
## 59 59 Stratum_4     B_16          4 Patient Navigator
## 60 60 Stratum_4     B_16          4           Control
## 61 61 Stratum_4     B_17          4           Control
## 62 62 Stratum_4     B_17          4 Patient Navigator
## 63 63 Stratum_4     B_17          4           Control
## 64 64 Stratum_4     B_17          4 Patient Navigator
## 65 65 Stratum_4     B_18          6           Control
```

```
## 66 66 Stratum_4     B_18          6           Control
## 67 67 Stratum_4     B_18          6 Patient Navigator
## 68 68 Stratum_4     B_18          6 Patient Navigator
## 69 69 Stratum_4     B_18          6           Control
## 70 70 Stratum_4     B_18          6 Patient Navigator
## 71 71 Stratum_4     B_19          6 Patient Navigator
## 72 72 Stratum_4     B_19          6           Control
## 73 73 Stratum_4     B_19          6           Control
## 74 74 Stratum_4     B_19          6 Patient Navigator
## 75 75 Stratum_4     B_19          6 Patient Navigator
## 76 76 Stratum_4     B_19          6           Control
## 77 77 Stratum_4     B_20          6           Control
## 78 78 Stratum_4     B_20          6 Patient Navigator
## 79 79 Stratum_4     B_20          6 Patient Navigator
## 80 80 Stratum_4     B_20          6           Control
##  [ reached 'max' / getOption("max.print") -- omitted 20 rows ]
```

Now Stratum 5

```
set.seed(2718)
stratum5 <- blockrand(n = 100,
                      block.prefix = "B_",
                      stratum = "Stratum_5",
                      num.levels = 2,
                      levels = c("Patient Navigator", "Control")
                      ) %>% filter(id <101)
stratum5
```

```
##    id   stratum block.id block.size         treatment
## 1   1 Stratum_5     B_01          2 Patient Navigator
## 2   2 Stratum_5     B_01          2           Control
## 3   3 Stratum_5     B_02          8           Control
## 4   4 Stratum_5     B_02          8           Control
## 5   5 Stratum_5     B_02          8           Control
## 6   6 Stratum_5     B_02          8           Control
## 7   7 Stratum_5     B_02          8 Patient Navigator
## 8   8 Stratum_5     B_02          8 Patient Navigator
## 9   9 Stratum_5     B_02          8 Patient Navigator
## 10 10 Stratum_5     B_02          8 Patient Navigator
## 11 11 Stratum_5     B_03          4 Patient Navigator
## 12 12 Stratum_5     B_03          4 Patient Navigator
## 13 13 Stratum_5     B_03          4           Control
## 14 14 Stratum_5     B_03          4           Control
## 15 15 Stratum_5     B_04          8 Patient Navigator
## 16 16 Stratum_5     B_04          8           Control
```

```
## 17 17 Stratum_5     B_04         8 Patient Navigator
## 18 18 Stratum_5     B_04         8           Control
## 19 19 Stratum_5     B_04         8 Patient Navigator
## 20 20 Stratum_5     B_04         8           Control
## 21 21 Stratum_5     B_04         8 Patient Navigator
## 22 22 Stratum_5     B_04         8           Control
## 23 23 Stratum_5     B_05         6           Control
## 24 24 Stratum_5     B_05         6 Patient Navigator
## 25 25 Stratum_5     B_05         6 Patient Navigator
## 26 26 Stratum_5     B_05         6 Patient Navigator
## 27 27 Stratum_5     B_05         6           Control
## 28 28 Stratum_5     B_05         6           Control
## 29 29 Stratum_5     B_06         6           Control
## 30 30 Stratum_5     B_06         6 Patient Navigator
## 31 31 Stratum_5     B_06         6           Control
## 32 32 Stratum_5     B_06         6 Patient Navigator
## 33 33 Stratum_5     B_06         6           Control
## 34 34 Stratum_5     B_06         6 Patient Navigator
## 35 35 Stratum_5     B_07         2           Control
## 36 36 Stratum_5     B_07         2 Patient Navigator
## 37 37 Stratum_5     B_08         4           Control
## 38 38 Stratum_5     B_08         4 Patient Navigator
## 39 39 Stratum_5     B_08         4           Control
## 40 40 Stratum_5     B_08         4 Patient Navigator
## 41 41 Stratum_5     B_09         6 Patient Navigator
## 42 42 Stratum_5     B_09         6 Patient Navigator
## 43 43 Stratum_5     B_09         6 Patient Navigator
## 44 44 Stratum_5     B_09         6           Control
## 45 45 Stratum_5     B_09         6           Control
## 46 46 Stratum_5     B_09         6           Control
## 47 47 Stratum_5     B_10         6 Patient Navigator
## 48 48 Stratum_5     B_10         6           Control
## 49 49 Stratum_5     B_10         6           Control
## 50 50 Stratum_5     B_10         6           Control
## 51 51 Stratum_5     B_10         6 Patient Navigator
## 52 52 Stratum_5     B_10         6 Patient Navigator
## 53 53 Stratum_5     B_11         2 Patient Navigator
## 54 54 Stratum_5     B_11         2           Control
## 55 55 Stratum_5     B_12         6           Control
## 56 56 Stratum_5     B_12         6           Control
## 57 57 Stratum_5     B_12         6 Patient Navigator
## 58 58 Stratum_5     B_12         6 Patient Navigator
## 59 59 Stratum_5     B_12         6 Patient Navigator
## 60 60 Stratum_5     B_12         6           Control
## 61 61 Stratum_5     B_13         2           Control
## 62 62 Stratum_5     B_13         2 Patient Navigator
```

```
## 63 63 Stratum_5     B_14          2 Patient Navigator
## 64 64 Stratum_5     B_14          2           Control
## 65 65 Stratum_5     B_15          6 Patient Navigator
## 66 66 Stratum_5     B_15          6 Patient Navigator
## 67 67 Stratum_5     B_15          6 Patient Navigator
## 68 68 Stratum_5     B_15          6           Control
## 69 69 Stratum_5     B_15          6           Control
## 70 70 Stratum_5     B_15          6           Control
## 71 71 Stratum_5     B_16          4 Patient Navigator
## 72 72 Stratum_5     B_16          4 Patient Navigator
## 73 73 Stratum_5     B_16          4           Control
## 74 74 Stratum_5     B_16          4           Control
## 75 75 Stratum_5     B_17          6 Patient Navigator
## 76 76 Stratum_5     B_17          6           Control
## 77 77 Stratum_5     B_17          6 Patient Navigator
## 78 78 Stratum_5     B_17          6           Control
## 79 79 Stratum_5     B_17          6 Patient Navigator
## 80 80 Stratum_5     B_17          6           Control
##  [ reached 'max' / getOption("max.print") -- omitted 20 rows ]
```

now bind these together in one assignment sheet

```
assign500 <- rbind(stratum1, stratum2, stratum3, stratum4, stratum5)
assign500 %>%
  rename(num = id) %>%
  select(stratum, num, treatment, block.id, block.size) %>%
  knitr::kable() %>%
  kable_styling()
```

stratum	num	treatment	block.id	block.size
Stratum_1	1	Control	B_01	8
Stratum_1	2	Control	B_01	8
Stratum_1	3	Control	B_01	8
Stratum_1	4	Patient Navigator	B_01	8
Stratum_1	5	Control	B_01	8
Stratum_1	6	Patient Navigator	B_01	8
Stratum_1	7	Patient Navigator	B_01	8
Stratum_1	8	Patient Navigator	B_01	8
Stratum_1	9	Control	B_02	4
Stratum_1	10	Control	B_02	4
Stratum_1	11	Patient Navigator	B_02	4
Stratum_1	12	Patient Navigator	B_02	4
Stratum_1	13	Control	B_03	4
Stratum_1	14	Patient Navigator	B_03	4
Stratum_1	15	Control	B_03	4
Stratum_1	16	Patient Navigator	B_03	4
Stratum_1	17	Control	B_04	4
Stratum_1	18	Control	B_04	4
Stratum_1	19	Patient Navigator	B_04	4
Stratum_1	20	Patient Navigator	B_04	4
Stratum_1	21	Control	B_05	2
Stratum_1	22	Patient Navigator	B_05	2
Stratum_1	23	Patient Navigator	B_06	8
Stratum_1	24	Control	B_06	8
Stratum_1	25	Patient Navigator	B_06	8
Stratum_1	26	Control	B_06	8
Stratum_1	27	Patient Navigator	B_06	8
Stratum_1	28	Control	B_06	8
Stratum_1	29	Control	B_06	8
Stratum_1	30	Patient Navigator	B_06	8
Stratum_1	31	Patient Navigator	B_07	8
Stratum_1	32	Control	B_07	8
Stratum_1	33	Patient Navigator	B_07	8
Stratum_1	34	Control	B_07	8
Stratum_1	35	Patient Navigator	B_07	8
Stratum_1	36	Patient Navigator	B_07	8
Stratum_1	37	Control	B_07	8
Stratum_1	38	Control	B_07	8
Stratum_1	39	Control	B_08	8
Stratum_1	40	Control	B_08	8
Stratum_1	41	Patient Navigator	B_08	8
Stratum_1	42	Patient Navigator	B_08	8
Stratum_1	43	Patient Navigator	B_08	8
Stratum_1	44	Control	B_08	8
Stratum_1	45	Patient Navigator	B_08	8
Stratum_1	46	Control	B_08	8
Stratum_1	47	Control	B_09	4
Stratum_1	48	Patient Navigator	B_09	4
Stratum_1	49	Control	B_09	4
Stratum_1	50	Patient Navigator	B_09	4
Stratum_1	51	Control	B_10	2
Stratum_1	52	Patient Navigator	B_10	2
Stratum_1	53	Control	B_11	2

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Chapter 17

One variable, single group

17.0.0.1 Continuous value – t test

17.0.0.2 Challenges for single continuous outcome

17.0.0.3 Estimate single Proportion

17.0.0.4 Challenges for single proportion

Chapter 18

One variable, two groups

18.0.0.1 Variable 1 is greater in group A vs group B

18.0.0.1.1 Test for skew – if not, T test

18.0.0.1.1.1 Challenges

18.0.0.1.2 If yes, Wilcoxon, non parametric

18.0.0.1.2.1 Challenges

18.0.0.2 Variable 2 proportion is greater in group A vs group B

18.0.0.2.1 If no rare cells, chi square

18.0.0.2.1.1 Challenges

18.0.0.2.2 If rare cells, Fischer exact test

18.0.0.2.2.1 Challenges

Chapter 19

One variable, Multiple groups

19.0.0.1 Fun with ANOVA - kind of a waste of time

19.0.0.1.1 Use regression instead

Chapter 20

Linear Regression

20.0.0.1 Single outcome, multiple possible predictors

20.0.0.1.1 Challenges

20.0.0.2 One predictor at a time – multiple univariate models – modelr and broom packages

20.0.0.2.1 Challenges

20.0.0.3 Choosing predictors for multivariate modeling – testing, dealing with collinearity

20.0.0.3.1 Challenges

20.0.0.4 Multivariate modeling

20.0.0.4.1 Challenges

20.0.0.5 Model fit checking

20.0.0.5.1 Challenges

20.0.0.6 presenting model results with RMarkdown

20.0.0.6.1 Challenges

Chapter 21

Sharing Models with Shiny

21.0.0.1 Basics of Shiny

21.0.0.2 Practice with model from linear regression chapter

Chapter 22

Logistic Regression

22.0.0.1 Concepts and OR

22.0.0.2 Modeling

22.0.0.3 Estimating GOF

22.0.0.4 AuROC

22.0.0.5 Sens Spec PPV NPV

22.0.0.6 Confusion Matrix

22.0.0.7 Present results with broom, RMarkdown

22.0.0.8 Make user-friendly model predictions with Shiny

Chapter 23

Meta-Analysis

23.0.0.1 Data search

23.0.0.2 Data collection

23.0.0.3 Data Exclusion

23.0.0.4 Data extraction and checking

23.0.0.5 Using Metafor package

23.0.0.6 Making Figures

23.0.0.7 Writing up results in RMarkdown

23.0.1 Take a look at the data

```
glimpse(mockstudy)
```

```
## Rows: 1,499
## Columns: 14
## $ case        <int> 110754, 99706, 105271, 105001, 1122...
## $ age         <int> 67, 74, 50, 71, 69, 56, 50, 57, 51,...
## $ arm         <chr> "F: FOLFOX", "A: IFL", "A: IFL", "G...
## $ sex         <fct> Male, Female, Female, Female, Femal...
## $ race        <chr> "Caucasian", "Caucasian", "Caucasia...
## $ fu.time     <int> 922, 270, 175, 128, 233, 120, 369, ...
## $ fu.stat     <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
```

```
## $ ps          <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,...
## $ hgb         <dbl> 11.5, 10.7, 11.1, 12.6, 13.0, 10.2,...
## $ bmi         <dbl> 25.09861, 19.49786, NA, 29.42922, 2...
## $ alk.phos    <int> 160, 290, 700, 771, 350, 569, 162, ...
## $ ast         <int> 35, 52, 100, 68, 35, 27, 16, 12, 25...
## $ mdquality.s <int> NA, 1, 1, 1, NA, 1, 1, 1, 1, 1, NA,...
## $ age.ord     <ord> 60-69, 70-79, 40-49, 70-79, 60-69, ...
```

23.0.2 Set up a chisquared table

with text interpretation with inline r code.

```
mockstudy %>%
  tabyl(arm, fu.stat) ->
outcome_table
names(outcome_table) <- c("Study Arm", "Lived", "Died")
outcome_table %>%
  gt()
```

Study Arm	Lived	Died
A: IFL	18	410
F: FOLFOX	99	592
G: IROX	26	354

```
mockstudy %>%
  tabyl(arm, fu.stat) %>%
  column_to_rownames('arm') %>%
  chisq.test() ->
results

results$statistic
```

```
## X-squared
##  35.66764
```

```
results$parameter
```

```
## df
##  2
```

23.0.3 Study Results

This is a statement of study results.

In the evaluation of followup status by study arm, the null hypothesis of independence was rejected, with a chi-squared statistic of 35.67, with 2 degrees of freedom, and a p value of 0, using the Pearson's Chi-squared test method.

23.0.4 Start with a barplot

for percent survival tag it as panel A for a multipanel plot

```
mockstudy %>%
  group_by(arm) %>%
  summarize(surv = length(which(fu.stat==1)),
         died = length(which(fu.stat==2)),
         pct_surv = surv*100/(died+surv)) %>%
  select(arm, surv, died, pct_surv) %>%
  ggplot() +
  aes(x=arm, y = pct_surv, fill=arm,
      label= round(pct_surv,1)) +
  geom_bar(stat= 'identity') +
  labs(y= "Percent Survived \n To End of Followup", x= "Study Arm", tag ="A") +
  theme_classic2(base_size = 15) +
  theme(legend.position = 'none') +
  geom_text(size=5, vjust=-0.3) +
  scale_fill_manual(values = c("black", "green", "blue")) +
  ylim(0,18) ->
p1
```

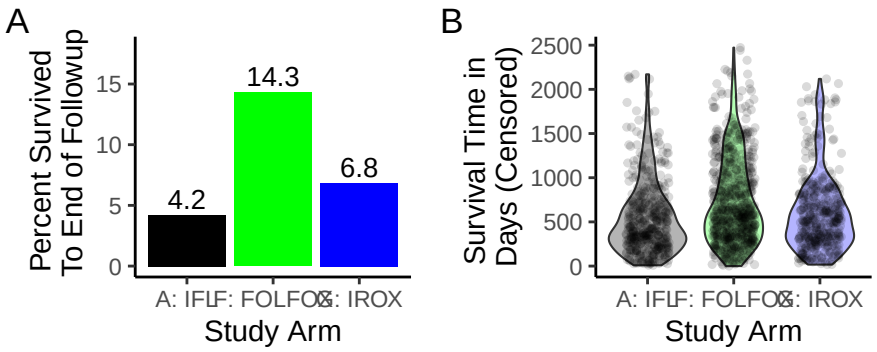
23.0.5 Now add a boxplot, make it multipanel

tagged as panel B

```
mockstudy %>%
  group_by(arm) %>%
ggplot() +
  aes(x=arm, y = fu.time, fill=arm) +
  geom_violin(alpha =0.3) +
  geom_jitter(width =0.25, alpha=0.15) +
  labs(y= "Survival Time in \nDays (Censored)", x= "Study Arm", tag = "B") +
  theme_classic2(base_size = 15) +
  theme(legend.position = 'none') +
  scale_fill_manual(values = c("black", "green", "blue")) ->
```

```
p2

p1 + p2   + plot_layout(ncol=2, heights = c(4,4))
```



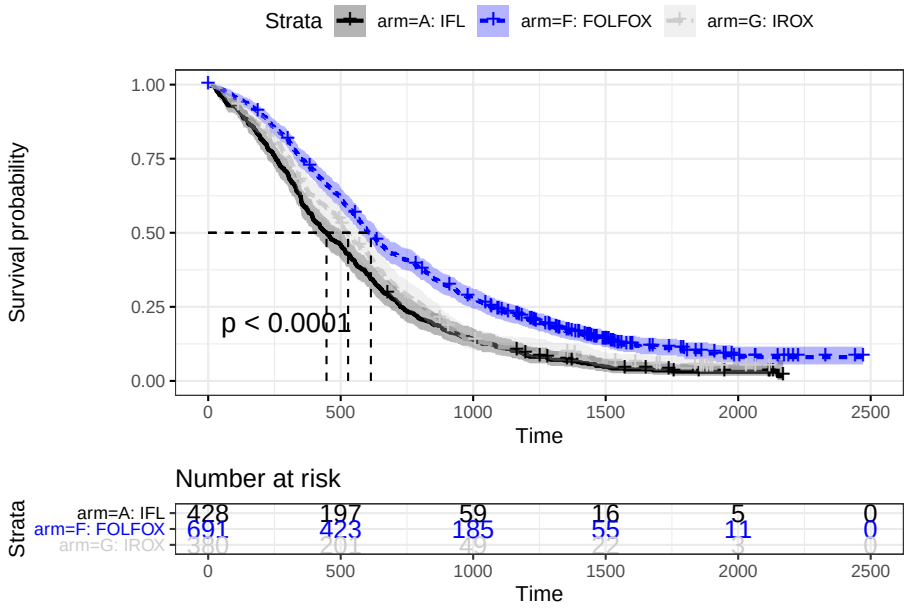
23.0.6 Now add a survival curve

For some reason, patchwork does not work with this survival curve

```
survfit(formula = Surv(fu.time, fu.stat) ~ arm, data= mockstudy) ->
fit

ggsurvplot(fit,
           pval = TRUE, conf.int = TRUE,
           risk.table = TRUE,
           risk.table.col = "strata",
           linetype = "strata",
           surv.median.line = "hv",
           ggtheme = theme_bw(),
           palette = c("black", "blue", "gray80"))
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```




```
mockstudy %>%
  tabyl(arm, fu.stat) %>%
  column_to_rownames('arm') %>%
  chisq.test() ->
results

results$statistic
```

```
## X-squared
##  35.66764
```

```
results$parameter
```

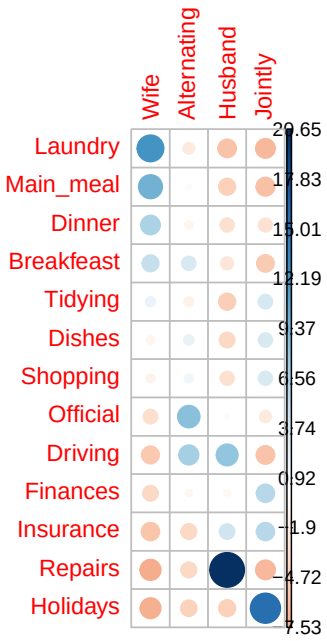
```
## df
##  2
```

23.1 Study Results

This is an R Markdown document.

In the evaluation of followup status by study arm, the null hypothesis of independence was rejected, with a chi-squared statistic of 35.67, with 2 degrees of freedom, and a p value of 0, using the Pearson's Chi-squared test method.

```
##
##  Pearson's Chi-squared test
##
## data:  .
## X-squared = 1944.5, df = 36, p-value <
## 0.00000000000000022
```



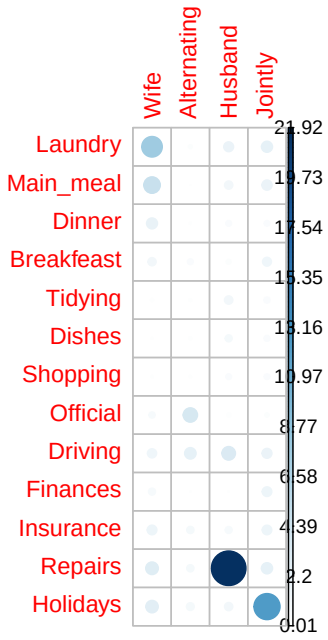
In the evaluation of household tasks by person responsible, the null hypothesis of independence was rejected, with a chi-squared statistic of 1944.46, with 36 degrees of freedom, and a p value of 0, using the Pearson's Chi-squared test method.

```
housetasks %>%
  as.data.frame() %>%
  rownames_to_column("Task") %>%
  filter(Task=="Finances" | Task=="Official") %>%
  select(Task, Wife, Husband) %>%
  column_to_rownames("Task") %>%
  chisq.test()
```

```
##
##  Pearson's Chi-squared test with Yates' continuity
##  correction
##
```

```
## data:  .
## X-squared = 0.0082362, df = 1, p-value = 0.9277
```

```
##              Wife Alternating Husband Jointly
## Laundry     7.738       0.272   1.777   2.246
## Main_meal   4.976       0.012   1.243   1.903
## Dinner      2.197       0.073   0.600   0.560
## Breakfeast  1.222       0.615   0.408   1.443
## Tidying     0.149       0.133   1.270   0.661
## Dishes      0.063       0.178   0.891   0.625
## Shopping    0.085       0.090   0.581   0.586
## Official    0.688       3.771   0.010   0.311
## Driving     1.538       2.403   3.374   1.789
## Finances    0.886       0.037   0.028   1.700
## Insurance   1.705       0.941   0.868   1.683
## Repairs     2.919       0.947  21.921   2.275
## Holidays    2.831       1.098   1.233  12.445
```



23.2 Comparing Two Measures of Centrality

- Means and Medians
- t tests and wilcoxon
- for numerical continuous data

23.3 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)?

23.4 How Skewed is Too Skewed?

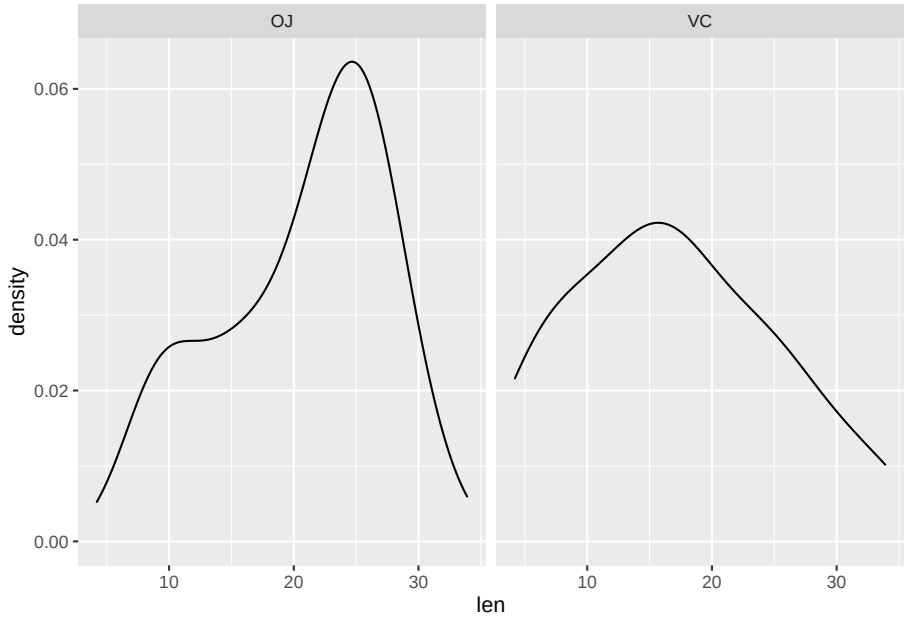
- Formal test of normality = Shapiro-Wilk test
- Use base data set called ToothGrowth

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

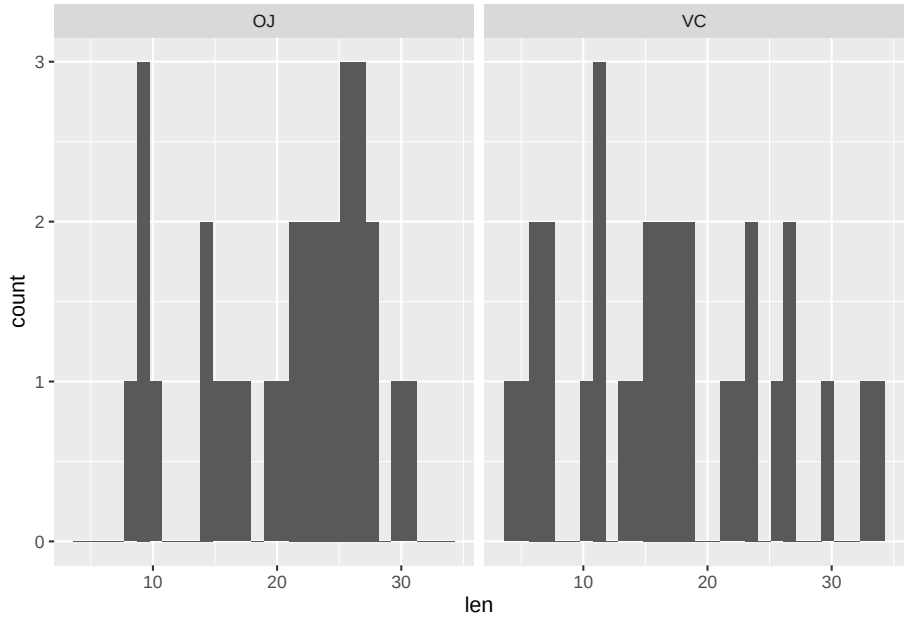
```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

23.5 Visualize the Distribution of data$len in ggplot

- Use geom_histogram or geom_density
- Bonus points: facet by supplement treatment
- Your turn to try it




```
data %>%
ggplot(mapping = aes(len)) +
  geom_histogram() +
  #geom_density() +
  facet_wrap(~supp)
```



23.6 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$len %>%
shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.96743, p-value = 0.1091
```

23.7 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

23.8 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

23.9 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?

23.10 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

23.11 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

23.12 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

23.13 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

23.14 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

23.15 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?

23.16 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
```

23.17 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

23.18 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)
```

23.19 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
```

23.20 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

23.21 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

23.22 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
```

23.23 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>
```

23.24 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

23.25 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

23.26 Comparing Two Measures of Centrality

- Means and Medians
- t tests and wilcoxon
- for numerical continuous data

23.27 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)?

23.28 How Skewed is Too Skewed?

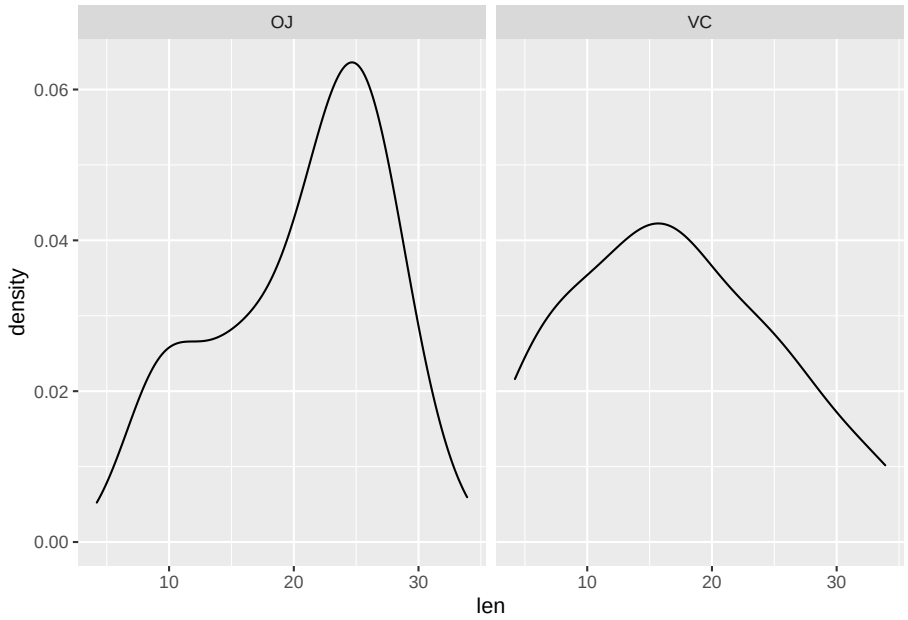
- Formal test of normality = Shapiro-Wilk test
- Use base data set called ToothGrowth

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

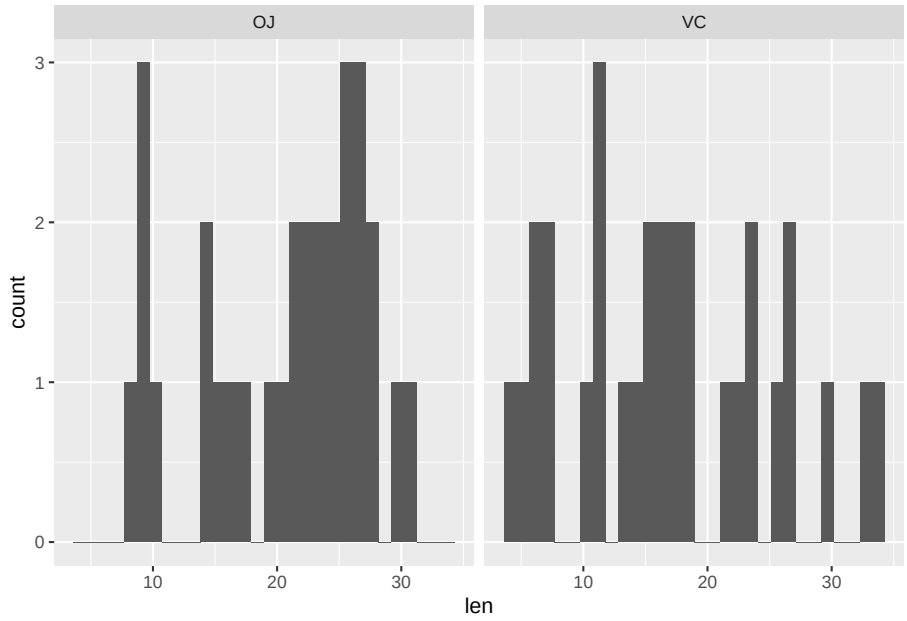
```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

23.29 Visualize the Distribution of data$len in ggplot

- Use geom_histogram or geom_density
- Bonus points: facet by supplement treatment
- Your turn to try it



```
data %>%
ggplot(mapping = aes(len)) +
  geom_histogram() +
  #geom_density() +
  facet_wrap(~supp)
```



23.30 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$len %>%
shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.96743, p-value = 0.1091
```

23.31 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

23.32 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

23.33 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?

23.34 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

23.35 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

23.36 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

23.37 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

23.38 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

23.39 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?

23.40 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
```

23.41 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

23.42 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)
```

23.43 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
```

23.44 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

23.45 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

23.46 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
```

23.47 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>
```

23.48 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

23.49 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

23.50 Review from t test part 1

- Comparing Two Measures of Centrality
  - is it normal (Shapiro test)?
    - if yes, use t test

    - if no, use Wilcoxon rank test (nonparametric)
  - are the observations paired?
    - if yes, use paired Wilcoxon or paired t test
    - if no, can use usual versions
  - are the variances equal in 2 groups (Levene test)?
    - if yes, OK to use Student's t test
    - if no, must use Welch t test (unequal var)

23.51 Review how to do t test part 1

- one sample
  - t.test(df$sleep_total, alternative = "two.sided", mu = 8)
- formula interface: outcome ~ groupvar df %>% filter(vore %in% c("herbi", "carni")) %>% t.test(formula = sleep_total ~ vore, data = .)
- use broom::tidy() to get results into a dataframe
- can use results in text with inline code
- use backticks before and after, start with r i.e. My result is [backtick]r code here[backtick].

23.52 What We have Left to Cover

- General info on Skewness and Kurtosis
- When to use t tests vs. rank (wilcoxon)
- Levene test for equal variances
  - When to use Welch t vs. Student's t
- Paired t and Wilcoxon tests

23.53 Skewness

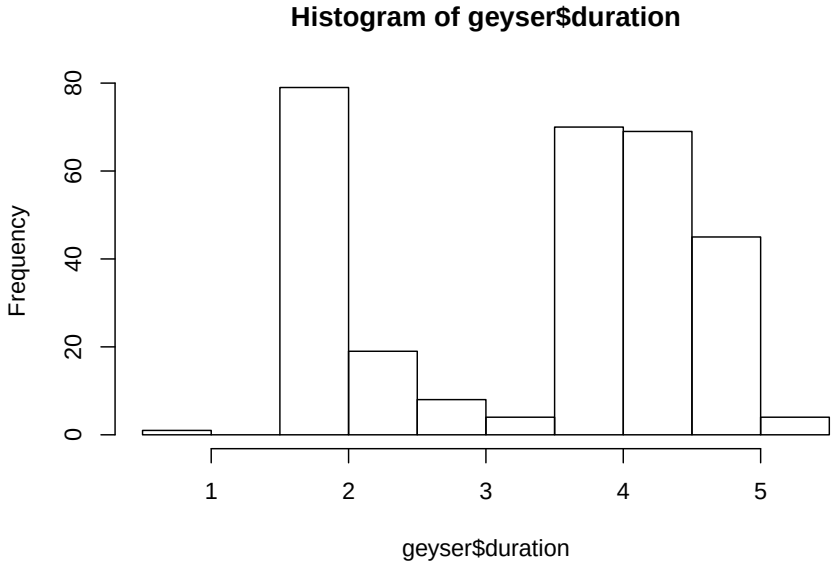
- A term for asymmetric data - obviously not normal
- Left-skewed - long leftward tail
- aka negatively skewed - mean less than the median - typical of age at death in US, number of questions answered on a survey, sq_ft/$ in Ann Arbor housing - often has a natural upper bound
- Right-skewed - long rightward tail
  - aka positively skewed
  - mean greater than the median
  - typical of length of stay, medical costs, children per family
  - mostly low, but some huge outliers
  - often has a natural lower bound

23.54 Skewness example: Old Faithful geyser eruption duration

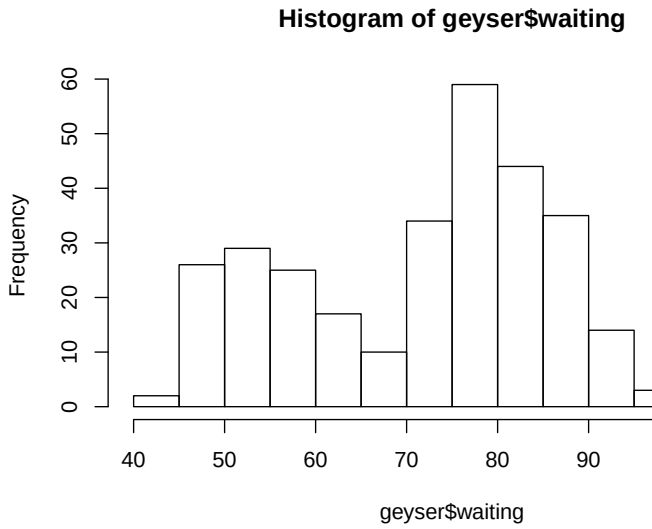
- Duration of eruptions in minutes is negatively skewed

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```



23.55 Skewness example: Old Faithful eruption wait times



- Wait time for eruptions in minutes is negatively skewed
  ## How Skewed is Too Skewed?
- Formal test of normality = Shapiro-Wilk test
- Use base data set called ToothGrowth
- test the len variable

```
library(tidyverse)
library(MASS)
shapiro.test(geyser$waiting)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  geyser$waiting
## W = 0.93871, p-value = 0.0000000008541
```

23.56 Kurtosis

- Descriptive of how tightly bunched data are
  - leptokurtotic - histogram has tall peak, little spread
    - think leaping high = lepto

    - values <3 are platykurtotic
  - platykurtotic - histogram is mostly flat, wide spread
    - think flat, like a platypus beak
    - values <3 are platykurtotic
  - kurtosis values ~ 3 are mesokurtotic

```
library(tidyverse)
library(MASS)
#kurtosis(geyser$waiting)
```

23.57 Decision tree for Comparing 2 group data

1. Is it normal (Shapiro test)?

- yes - t.test
- no - wilcoxon

2. Is it paired data?

- use paired variants of either t.test or wilcoxon

3. If normal, are the variances of the two groups equal (Levene test)?

- if yes, use Student t test (var.equal=TRUE)
- if no, use Welch t test (var.equal = FALSE)

23.58 Tooth length data

- We know it is reasonably normal from Shapiro test
  - Ok to use t test
- Should we use the Student version (var.equal) or the Welch version?
  - Welch (var.equal = FALSE) is the default
- Determine this with Levene test for 2 groups (in car package)
  - use Bartlett test if >2 groups and normally distributed

```
library(tidyverse)
library(car)
data <- ToothGrowth
data %>%
  filter(dose > 0.9) %>%
  leveneTest(len ~ as.factor(dose), data = .)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.6144 0.2116
##       38
```

```
# note has to be as.factor - will not evaluate if groupvar is numeric
```

23.59 Interpretation of Levene's test

- the null hypothesis is that all populations variances are equal;
- the alternative hypothesis is that at least two of them differ.

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.6144 0.2116
##       38
```

- will not reject null, OK to use Student t test

23.60 Arguments for t.test

x = vector of continuous numerical data alternative = c("two.sided", "less", "greater"), mu = 0 paired = FALSE var.equal = FALSE conf.level = 0.95 Open the t.test help - type in console "?t.test"

23.61 Mammal sleep hours for t test

First with Student t test, compare herbivores to omnivores by sleep_rem

```
df <- msleep
df %>%
  filter(vore != 'carni') %>%
  filter(vore != 'insecti') %>%
  t.test(sleep_rem ~ vore, data=., var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  sleep_rem by vore
## t = -1.9638, df = 40, p-value = 0.05653
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.19495934  0.01718156
## sample estimates:
## mean in group herbi  mean in group omni
##            1.366667            1.955556
```

Now with Welch's t test, do the same thing compare herbivores to omnivores by sleep_rem

```
df <- msleep
df %>%
  filter(vore %in% c('omni','herbi')) %>%
  t.test(sleep_rem ~ vore, data=., var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  sleep_rem by vore
## t = -1.937, df = 34.768, p-value = 0.06091
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.20623797  0.02846019
## sample estimates:
## mean in group herbi  mean in group omni
##            1.366667            1.955556
```

23.62 Paired T test

```
library(PairedData)
# matched exposed and control children
# exposed have parents who worked in factory with lead
head(BloodLead)

t.test(x=BloodLead$Exposed, y= BloodLead$Control, paired=TRUE,
       alternative = "two.sided")
```

23.63 Paired Wilcoxon test

```r
library(PairedData)
# matched exposed and control children
# exposed have parents who worked in factory with lead
head(BloodLead)

wilcox.test(x=BloodLead$Exposed, y= BloodLead$Control,
              paired=TRUE,alternative = "two.sided")
```

```
- Try it out, see if you can interpret results


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




<!--chapter:end:io31-ttest_part2.Rmd-->

---
title: "Building Table One for a Clinical Study"
output: html_notebook
author: Peter D.R. Higgins
---

We will start by loading libraries and data.


```r
library(arsenal)
library(knitr)
library(survival)
library(tidyverse)

## -- Attaching packages ------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0.9000      v purrr   0.3.4
## v tibble  3.0.1           v dplyr   0.8.99.9002
```
```

```
## v tidyr   1.0.2           v stringr 1.4.0
## v readr   1.3.1           v forcats 0.5.0

## -- Conflicts ---------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data(mockstudy)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(keyring)
library(REDCapR)
```

```
dim(mockstudy)
```

```
## [1] 1499   14
```

```
glimpse(mockstudy)
```

```
## Rows: 1,499
## Columns: 14
## $ case        <int> 110754, 99706, 105271, 105001, 1122...
## $ age         <int> 67, 74, 50, 71, 69, 56, 50, 57, 51,...
## $ arm         <chr> "F: FOLFOX", "A: IFL", "A: IFL", "G...
## $ sex         <fct> Male, Female, Female, Female, Femal...
## $ race        <chr> "Caucasian", "Caucasian", "Caucasia...
## $ fu.time     <int> 922, 270, 175, 128, 233, 120, 369, ...
## $ fu.stat     <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ ps          <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,...
## $ hgb         <dbl> 11.5, 10.7, 11.1, 12.6, 13.0, 10.2,...
## $ bmi         <dbl> 25.09861, 19.49786, NA, 29.42922, 2...
## $ alk.phos    <int> 160, 290, 700, 771, 350, 569, 162, ...
## $ ast         <int> 35, 52, 100, 68, 35, 27, 16, 12, 25...
## $ mdquality.s <int> NA, 1, 1, 1, NA, 1, 1, 1, 1, 1, NA,...
## $ age.ord     <ord> 60-69, 70-79, 40-49, 70-79, 60-69, ...
```

23.63.1 Basic Table 1

Let's make a basic Table 1 grouped by arm with details on sex and age in each group.

```
#summary by groups
tab1 <- tableby(arm ~ sex + age, data = mockstudy)
summary(tab1, text=TRUE)
```

```
##
##
## |             | A: IFL (N=428)  | F: FOLFOX (N=691) | G: IROX (N=380) | Total (N=1499)  | p va
## |:------------|:---------------:|:-----------------:|:---------------:|:---------------:|-----
## |sex          |                 |                   |                 |                 |   0.
## |-  Male      |   277 (64.7%)   |    411 (59.5%)    |   228 (60.0%)   |   916 (61.1%)   |
## |-  Female    |   151 (35.3%)   |    280 (40.5%)    |   152 (40.0%)   |   583 (38.9%)   |
## |Age in Years |                 |                   |                 |                 |   0.
## |-  Mean (SD) | 59.673 (11.365) |  60.301 (11.632)  | 59.763 (11.499) | 59.985 (11.519) |
## |-  Range     | 27.000 - 88.000 |  19.000 - 88.000  | 26.000 - 85.000 | 19.000 - 88.000 |
```

23.63.2 Table 1 without groups

Let's make a Table 1 - but ungrouped, with stats on BMI, sex, Age in each group.

```
#summary without groups
tab.noby <- tableby(~ bmi + sex +age, data = mockstudy)
summary(tab.noby)
```

```
##
##
## |                            | Overall (N=1499) |
## |:---------------------------|:----------------:|
## |**Body Mass Index (kg/m^2)** |                  |
## |   N-Miss     |        33        |
## |   Mean (SD)  |  27.206 (5.432)  |
## |   Range      | 14.053 - 60.243  |
## |**sex**                     |                  |
## |   Male       |   916 (61.1%)    |
## |   Female     |   583 (38.9%)    |
## |**Age in Years**            |                  |
## |   Mean (SD)  | 59.985 (11.519)  |
## |   Range      | 19.000 - 88.000  |
```

23.63.3 Table 1 grouped, build right hand side

Let's make a Table 1 by arm, with stats on ps, sex, Age in each group.

```
myvars <- names(mockstudy)
rhs <- paste(myvars[8:10], collapse = '+')
rhs
```

```
## [1] "ps+hgb+bmi"
```

```
as.formula(paste('arm ~', rhs))
```

```
## arm ~ ps + hgb + bmi
```

```
summary(tableby(as.formula(paste('arm ~', rhs)), data=mockstudy))
```

```
##
##
## |                              | A: IFL (N=428)  | F: FOLFOX (N=691) | G: IROX (N=380
## |:-----------------------------|:---------------:|:-----------------:|:--------------
## |**ps**                        |                 |                   |
## |   N-Miss      |       69        |        141        |       56
## |   Mean (SD)   |  0.529 (0.597)  |   0.547 (0.595)   |  0.537 (0.606)
## |   Range       |  0.000 - 2.000  |   0.000 - 2.000   |  0.000 - 2.000
## |**hgb**                       |                 |                   |
## |   N-Miss      |       69        |        141        |       56
## |   Mean (SD)   | 12.276 (1.686)  |  12.381 (1.763)   | 12.373 (1.680)
## |   Range       | 9.060 - 17.300  |  9.000 - 18.200   | 9.000 - 17.000
## |**Body Mass Index (kg/m^2)**  |                 |                   |
## |   N-Miss      |        9        |        20         |        4
## |   Mean (SD)   | 27.290 (5.552)  |  27.210 (5.173)   | 27.106 (5.751)
## |   Range       | 14.053 - 53.008 |  16.649 - 49.130  | 15.430 - 60.24
```

23.63.4 Table 1 digit control

Let's make a Table 1 but now control # of digits

```
summary(tableby(arm ~ sex + fu.time, data=mockstudy), digits=4, digits.p =2, digits.pct
```

```
##
##
## |                              |   A: IFL (N=428)    |  F: FOLFOX (N=691)  |   G: IROX
```

```
## |:--------------------------|:------------------:|:------------------:|:------------------
## |**sex**                    |                    |                    |
## |   Male     |     277 (64.7%)    |     411 (59.5%)    |     228 (60.0%)
## |   Female   |     151 (35.3%)    |     280 (40.5%)    |     152 (40.0%)
## |**fu.time**                |                    |                    |
## |   Mean (SD) | 553.5841 (419.6065) | 731.2460 (487.7443) | 607.2421 (435.5092)
## |   Range    | 9.0000 - 2170.0000 | 0.0000 - 2472.0000 | 17.0000 - 2118.0000
```

23.63.5 Table 1 out to word document

Let's make a Table 1 and write to MS Word

```
tab1 <- tableby(arm ~ sex + age, data = mockstudy)
write2word(tab1, file = 'table1.docx')
```

```
##
##
## processing file: table1.docx.Rmd

##
  |
  |                                                  |   0%
  |
  |..................................................| 100%
##   ordinary text without R code

## output file: table1.docx.knit.md

## /Applications/RStudio.app/Contents/MacOS/pandoc/pandoc +RTS -K512m -RTS table1.docx.utf8.md --

##
## Output created: table1.docx
```

23.63.6 Table 1 with new data

Let's make a Table 1 with a new dataset from REDCap We will assign arms 1 and 2, then make a table one

```
fake_df <- REDCapR::redcap_read_oneshot(
  redcap_uri = "https://bbmc.ouhsc.edu/redcap/api/",
  token      = "F304DEC3793FECC3B6DEEFF66302CAD3"
)$data
```

```
## 500 records and 13 columns were read from REDCap in 0.5 seconds.  The http status co
```

```
#assign arms
fake_df$arm <- c(rep(1,250), rep(2, 250))
fake_df$race<- as.factor(fake_df$race)
```

```
rhs <- paste(names(fake_df[c(8:11)]), collapse = '+')
tab2 <- summary(tableby(as.formula(paste('arm ~', rhs)), data=fake_df), pfootnote=TRUE)
tab2
```

```
##
##
## |                            |     1 (N=250)     |     2 (N=250)     |   Total (N=50
## |:---------------------------|:-----------------:|:-----------------:|:-------------
## |**race**                    |                   |                   |
## |   1         |     12 (4.8%)     |     7 (2.8%)      |     19 (3.8%)
## |   3         |     24 (9.6%)     |    32 (12.8%)     |    56 (11.2%)
## |   4         |    176 (70.4%)    |    176 (70.4%)    |    352 (70.4%
## |   5         |    31 (12.4%)     |    28 (11.2%)     |    59 (11.8%)
## |   6         |     7 (2.8%)      |     7 (2.8%)      |     14 (2.8%)
## |**gender**                  |                   |                   |
## |   Mean (SD) |   0.460 (0.499)   |   0.500 (0.501)   |   0.480 (0.50
## |   Range     |   0.000 - 1.000   |   0.000 - 1.000   |   0.000 - 1.0
## |**height**                  |                   |                   |
## |   Mean (SD) | 173.092 (10.610)  |  172.476 (9.725)  | 172.784 (10.1
## |   Range     | 142.500 - 204.300 | 142.900 - 205.300 | 142.500 - 205
## |**weight**                  |                   |                   |
## |   Mean (SD) | 110.356 (22.806)  | 109.940 (26.311)  | 110.148 (24.5
## |   Range     | 48.000 - 171.000  | 36.000 - 189.000  | 36.000 - 189
## 1. Pearson's Chi-squared test
## 2. Linear Model ANOVA
```

```
# write2word(tab2, file = 'table3.docx')
# write2html(tab2, "~/table3.html")
```

23.63.7 Take a look at the data

```
glimpse(mockstudy)
```

```
## Rows: 1,499
## Columns: 14
## $ case        <int> 110754, 99706, 105271, 105001, 1122...
```

```
## $ age          <int> 67, 74, 50, 71, 69, 56, 50, 57, 51,...
## $ arm          <chr> "F: FOLFOX", "A: IFL", "A: IFL", "G...
## $ sex          <fct> Male, Female, Female, Female, Femal...
## $ race         <chr> "Caucasian", "Caucasian", "Caucasia...
## $ fu.time      <int> 922, 270, 175, 128, 233, 120, 369, ...
## $ fu.stat      <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ ps           <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,...
## $ hgb          <dbl> 11.5, 10.7, 11.1, 12.6, 13.0, 10.2,...
## $ bmi          <dbl> 25.09861, 19.49786, NA, 29.42922, 2...
## $ alk.phos     <int> 160, 290, 700, 771, 350, 569, 162, ...
## $ ast          <int> 35, 52, 100, 68, 35, 27, 16, 12, 25...
## $ mdquality.s <int> NA, 1, 1, 1, NA, 1, 1, 1, 1, 1, NA,...
## $ age.ord      <ord> 60-69, 70-79, 40-49, 70-79, 60-69, ...
```

23.63.8 Set up a chisquared table

with text interpretation with inline r code.

```
mockstudy %>%
  tabyl(arm, fu.stat) ->
outcome_table
names(outcome_table) <- c("Study Arm", "Lived", "Died")
outcome_table %>%
  gt()
```

Study Arm	Lived	Died
A: IFL	18	410
F: FOLFOX	99	592
G: IROX	26	354

```
mockstudy %>%
  tabyl(arm, fu.stat) %>%
  column_to_rownames('arm') %>%
  chisq.test() ->
results

results$statistic
```

```
## X-squared
##  35.66764
```

```
results$parameter
```

```
## df
##  2
```

23.63.9 Study Results

This is a statement of study results.

In the evaluation of followup status by study arm, the null hypothesis of independence was rejected, with a chi-squared statistic of 35.67, with 2 degrees of freedom, and a p value of 0, using the Pearson's Chi-squared test method.

23.63.10 Start with a barplot

for percent survival tag it as panel A for a multipanel plot

```
mockstudy %>%
  group_by(arm) %>%
  summarize(surv = length(which(fu.stat==1)),
         died = length(which(fu.stat==2)),
         pct_surv = surv*100/(died+surv)) %>%
  select(arm, surv, died, pct_surv) %>%
  ggplot() +
  aes(x=arm, y = pct_surv, fill=arm) +
  geom_bar(stat= 'identity') +
  labs(y= "Percent Survived", x= "Study Arm", tag ="A") +
  theme_minimal() +
  scale_fill_manual(values = c("black", "blue", "grey80")) ->
p1
```

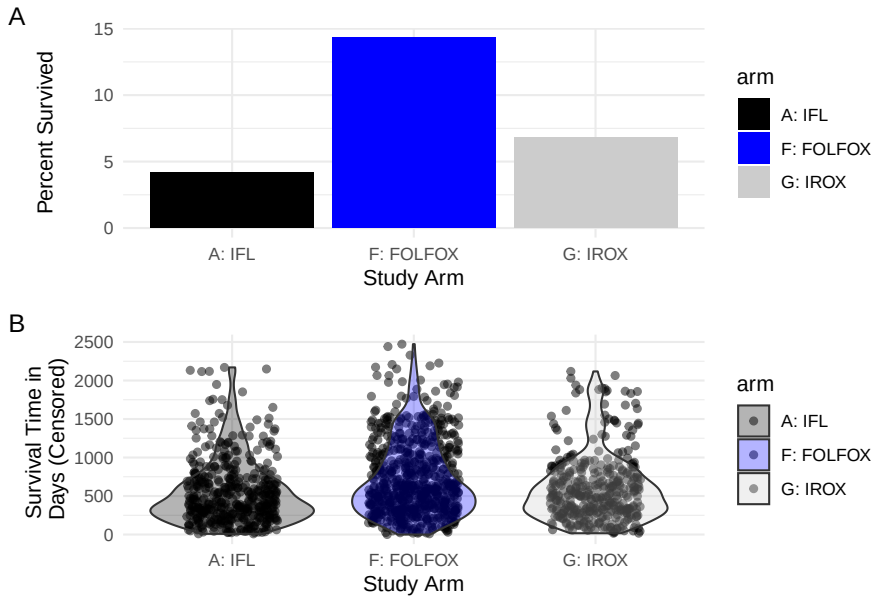
23.63.11 Now add a boxplot, make it multipanel

tagged as panel B

```
mockstudy %>%
  group_by(arm) %>%
ggplot() +
  aes(x=arm, y = fu.time, fill=arm) +
  geom_jitter(width =0.25, alpha=0.5) +
  geom_violin(alpha =0.3) +
  labs(y= "Survival Time in \nDays (Censored)", x= "Study Arm", tag = "B") +
```

```
  theme_minimal() +
  scale_fill_manual(values = c("black", "blue", "grey80")) ->
p2

p1 + p2   + plot_layout(ncol=1, heights = c(4,4))
```



23.63.12 Now add a survival curve

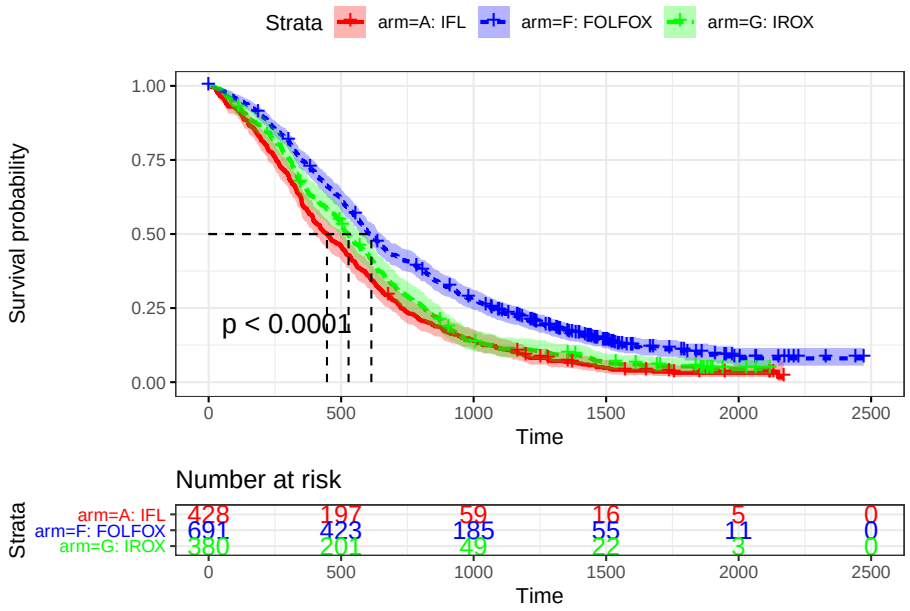
For some reason, patchwork does not work with this survival curve

```
survfit(formula = Surv(fu.time, fu.stat) ~ arm, data= mockstudy) ->
fit

ggsurvplot(fit,
           pval = TRUE, conf.int = TRUE,
           risk.table = TRUE,
           risk.table.col = "strata",
           linetype = "strata",
           surv.median.line = "hv",
           ggtheme = theme_bw(),
           palette = c("red", "blue", "green"))
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
```

```
## Results may be unexpected or may change in future versions of ggplot2.
```



Let's start by loading libraries.

```
library(tidyverse)
library(broom)
library(purrr)
library(knitr)
library(kableExtra)
library(magrittr)
```

We will start with the mtcars dataset, with its many numeric variables.

Note that mtcars is a particularly convenient dataset for modeling. There are no character variables, and no extra (non-predictor) variables to remove. If there are predictors in your dataset that are of type(character), you probably want to convert them to factors, and order them into levels (if they are ordinal, rather than categorical) before modeling. The *forcats* package can be very helpful for this.

We will plan for modeling to predict the outcome variable, mpg, or miles per gallon. Check out all the available predictors below.

How well will they predict mpg? Should you put some of them, or all of them, into a multivariate model?

Usually we look at how well single predictors predict the outcome variable in univariate models, then select the promising ones ($p < 0.20$, or $p < 0.10$, depending on how stringent you want to be) to put into a multivariate model.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0
##                   gear carb
## Mazda RX4            4    4
## Mazda RX4 Wag        4    4
## Datsun 710           4    1
## Hornet 4 Drive       3    1
## Hornet Sportabout    3    2
## Valiant              3    1
```

In an R notebook, you can add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

Now let's look at an example of a single univariate predictor model, using cylinders as the predictor of mpg, using the *lm* function.

With the broom package, we can *tidy* the model to get a dataframe as our output.

```
model <- mtcars %>%
  lm(mpg ~ cyl, data = .)
model %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    37.9      2.07      18.3  8.37e-18
## 2 cyl            -2.88     0.322     -8.92 6.11e-10
```

We can also use the broom function *glance* to get statistics for the whole model.

```
model %>%
  glance()
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic  p.value    df
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>
## 1     0.726         0.717  3.21      79.6 6.11e-10     2
## # ... with 5 more variables: logLik <dbl>, AIC <dbl>,
## #   BIC <dbl>, deviance <dbl>, df.residual <int>
```

Now let's look at testing multiple univariate predictors for mpg - by constructing and summarizing multiple univariate models from one chain of tidy code using the *map* function from the **purrr** package.

This is a bit complicated.

1. First we define full_df as mtcars, with both predictor variables and the outcome variable, mpg

2. Then we use *select* to remove mpg (our outcome variable), to get a new dataframe of only predictor variables. We will pipe this predictors dataframe (predictors_df) into subsequent functions, but will also retain the ability to call the entire mtcars dataframe (which still includes the outcome variable mpg).

3. Then we *map* the *lm* function over all the predictor variables (.x), using the full data from mtcars, which includes our outcome variable, mpg.

4. We then *tidy* the resulting 10 models and save this as a list of 10 dataframes, each named for their predictor variable.

5. Then we save the predictor variable names from this list to a vector called pred, which we will use later to label the predictors.

6. Then we *bind_rows* (stack the rows) of the tidy dataframes from the 10 models in the list together into a single dataframe, and *filter* out the intercept rows.

7. Then we add a column variable called predictor, and *select* the 3 variables we want, in the proper order.

8. We *arrange* the rows by p value, then save this dataframe as an object called bivariate_table, and display it nicely in HTML with the *kable* function from knitr.