

EPIB 607: Inferential Statistics

Sahir Bhatnagar and James Hanley

2018-09-10

Contents

I	Preface	5
1	Welcome	7
1.1	Objectives	7
1.2	Audience	7
1.3	About these notes	7
1.4	R Code Conventions	8
1.5	Rendering Mathematical Formulae	8
1.6	Development	8
1.7	About the authors	8
1.8	License	9
2	Course Information	11
2.1	Teaching strategy	11
2.2	A focus on computation	11
2.3	DataCamp	12
2.4	Grade Distribution	12
3	Target Syllabus	13
3.1	Descriptive Statistics	13
3.2	Sampling Distributions	13
3.3	Introduction to Inference	14
3.4	One-sample Inference	14
3.5	Two-sample Inference	14
3.6	Regression	14
3.7	Nonparametric Statistics	14
4	Prerequisites	15
4.1	Git	15
4.2	R and RStudio	15
5	Schedule	17
5.1	Week 1	17
5.2	Week 2	17
5.3	Week 3	18
5.4	Week 4	18
5.5	Week 5	18
5.6	Week 6	19
5.7	Week 7	19
5.8	Week 8	19
5.9	Week 9	20
5.10	Week 10	20
5.11	Week 11	20

5.12	Week 12	21
5.13	Week 13	21
5.14	Week 14	21
6	Slides	23
7	Assignments	25
8	DALITE	27
9	Terms and Concepts	29
II	Part I	31
10	Data Graphics	33
10.1	Visual Cues	33
10.2	Coordinate systems	33
10.3	Scale	34
10.4	Context	34
10.5	Facets and layers	35
10.6	Color	35
10.7	Examples	37
10.8	Exercises	40
A	Vectorization, *apply and for loops	43
A.1	Vectorization	43
A.2	Family of *apply functions	44
A.3	Creating dynamic documents with mapply	47
B	Appendix B	49

Part I

Preface

Chapter 1

Welcome

Welcome to the course notes for [EPIB 607: Inferential Statistics](#) at McGill University.

1.1 Objectives

The aim of this course is to provide students with basic principles of statistical inference so that they can:

1. Visualize/Analyze/Interpret data using statistical methods **with a computer**.
2. Understand the statistical results in a scientific paper.
3. Apply statistical methods in their own research.
4. Use the methods learned in this course as a foundation for more advanced biostatistics courses.

1.2 Audience

The principal audience is researchers in the natural and social sciences who haven't had an introductory course in statistics (or did have one a long time ago). This audience accepts that statistics has penetrated the life sciences pervasively and is required knowledge for both doing research and understanding scientific papers.

1.3 About these notes

These notes are a collection of useful links, videos, online resources and papers for an introductory course in statistics. The instructors have found that no single book sufficiently teaches all the topics covered in this course. Part of this is due to advancements in computing which have far outpaced the publication of modern textbooks. Indeed, the computer has replaced many of the calculations that were traditionally taught to be done by hand. We direct the readers to what we think is a good learning resource for a given topic (following the **Flipped Classroom** strategy). We also provide our own commentary and notes when we think its useful.

1.4 R Code Conventions

We use `R` code throughout these notes. When R code is displayed¹ it will be typeset using a `monospace` font with syntax highlighting enabled to ensure the differentiation of functions, variables, and so on. For example, the following adds 1 to 1

```
a = 1L + 1L
a
```

Each code segment may contain actual output from R. Such output will appear in grey font prefixed by `#>`. For example, the output of the above code segment would look like so:

```
[1] 2
```

1.5 Rendering Mathematical Formulae

Throughout these notes, there will be mathematical symbols used to express the material. Depending on the version of the book, there are two different rendering engines.

- For the online version, the text uses [MathJax](#) to render mathematical notation for the web. In the event the formulae does not load for a specific chapter, first try to refresh the page. 9 times out of 10 the issue is related to the software library not loading quickly. You can also right-click to see the corresponding LaTeX code used to produce the equation.
- For the pdf version, the text is built using the recommended AMS LaTeX symbolic packages. As a result, there should be no issue displaying equations. An example of a mathematical rendering capabilities would be given as:

$$a^2 + b^2 = c^2$$

1.6 Development

This book is built with [bookdown](#) and is open source and freely available. This approach encourages contributions, ensures reproducibility and provides access to the material worldwide. The online version of the book is hosted at sahirbhatnagar.com/EPIB607 and kept up-to-date thanks to [Travis](#). The entire source code is available at <https://github.com/sahirbhatnagar/EPIB607>.

If you notice any errors, we would be grateful if you would let us know by filing an issue [here](#) or making a pull request by clicking the edit button in the top-left corner of the text:

The version of the book you are reading now was built on 2018-09-10 and was built on [Travis](#).

1.7 About the authors

Sahir Bhatnagar James Hanley

¹<https://raw.githubusercontent.com/coatless/spm/master/index.Rmd>

**Figure 1.1**

- Sahir R. Bhatnagar: Assistant Professor of Biostatistics - McGill University, Montreal, Canada.
 - Website: <https://sahirbhatnagar.com/>
 - Twitter: [syfi_24](#)
 - GitHub: <https://github.com/sahirbhatnagar>
- James A. Hanley: Professor of Biostatistics - McGill University, Montreal, Canada.
 - Webpage: <http://www.medicine.mcgill.ca/epidemiology/hanley/>

1.8 License

This work is licensed under a Creative Commons Attribution 4.0 International License

Chapter 2

Course Information

- Instructor: [Sahir Bhatnagar](#)
- Teaching Assistants: [Kody Crowell](#), [Guanbo Wang](#), [Himasara Marasinghe](#)
- Website: <http://sahirbhatnagar.com/EPIB607/>
- Lectures: Monday 11:30am - 1:30pm, Thursday 8:30am - 10:30am
- Location: McMed 1034
- Office Hours: TBD
- Prerequisite(s): Calculus and Algebra
- Texts: *The Practice of Statistics in the Life Sciences*, 3rd Edition by Baldi & Moore.

2.1 Teaching strategy

This course will follow the **Flipped Classroom** model. Here, students are expected to have engaged with the material before coming to class (based on very precise pre-class instructions). The students will then be expected to answer a series of conceptual multiple choice questions using the [DALITE](#) online platform ([Bhatnagar et al., 2016](#)).

This allows the instructor to delegate the delivery of basic content and definitions to textbooks and videos, and enforces the idea that students cannot be simply passive recipients of information. This approach then allows the professor to focus valuable class time on nurturing efficient discussions surrounding the ideas within the content, guiding interactive exploration of typical misconceptions, and promoting collaborative problem solving with peers.

2.2 A focus on computation

Classic introductory statistics textbooks were written during a time when computers were still in their infancy. As such, even the newer editions heavily rely on *by-hand* computations such as looking up tables for tail probabilities. We take a modern approach and introduce computational methods in statistics with the statistical software program R.



Figure 2.1

2.3 DataCamp

This class is supported by [DataCamp](#), the most intuitive learning platform for data science. Learn R, Python and SQL the way you learn best through a combination of short expert videos and hands-on-the-keyboard exercises. Take over 100+ courses by expert instructors on topics such as importing data, data visualization or machine learning and learn faster through immediate and personalised feedback on every exercise.

You will be asked to complete some of the courses in DataCamp for background reading or for assignments. You can sign up for a free account at [this link](#). Note: you are required to sign up with a @mail.mcgill.ca or @mcgill.ca email address.

2.4 Grade Distribution

Assignments	40%
DALITE	15%
Midterm	15%
Project	10%
Final Exam	20%

Chapter 3

Target Syllabus

Abbreviation	Description
JH	James Hanley notes
EM	Erica Moodie notes
OS	Olli Saarela notes
AAO	Against all odds video series
B&M	The practice of statistics in the life sciences by Baldi and Moore, 3rd edition
Freedman	Statistics by Freedman, Pisani, Purves, Adhikari, 2nd edition
dataviz	Fundamentals of Data Visualization by Claus O. Wilke

3.1 Descriptive Statistics

Topic	Video	Readings
Histograms	[AAO unit 3](https://www.learner.org/courses/againstalldds/unitpages/unit03.html)	[AAO unit 3, pages 1-6](https://www.learner.org/courses/againstalldds/unitpages/unit03.html)
Density Plots	–	[dataviz chapter 7](https://www.datavizproject.com/chapter7/)
Measures of Center	[AAO unit 4](https://www.learner.org/courses/againstalldds/unitpages/unit04.html)	[AAO unit 4, pages 1-6](https://www.learner.org/courses/againstalldds/unitpages/unit04.html)
Boxplots	[AAO unit 5](https://www.learner.org/courses/againstalldds/unitpages/unit05.html)	[AAO unit 5, pages 1-5](https://www.learner.org/courses/againstalldds/unitpages/unit05.html)
Standard Deviation	[AAO unit 6](https://www.learner.org/courses/againstalldds/unitpages/unit06.html)	[AAO unit 6, pages 1-3](https://www.learner.org/courses/againstalldds/unitpages/unit06.html)
Data Visualization	[Hans Rosling BBC](https://www.youtube.com/watch?v=jbkSRLYSojo)	[JH notes on Descriptives]

3.2 Sampling Distributions

Topic	Video	Readings
Parameters and Statistics	–	[[JH section 1]](https://www.learner.org/courses/againstalldds/unitpages/unit22.html)
Sampling Distributions	[AAO unit 22](https://www.learner.org/courses/againstalldds/unitpages/unit22.html)	[[AAO unit 22, pages 1-3]](https://www.learner.org/courses/againstalldds/unitpages/unit22.html)
Central Limit Theorem	[AAO unit 22](https://www.learner.org/courses/againstalldds/unitpages/unit22.html)	[B&M pages 321-330]
The Bootstrap	–	[[Computer-Intensive Statistics]]

3.3 Introduction to Inference

Topic	Video	Readings
Confidence Intervals	[AAO unit 24](https://www.learner.org/courses/againstalldds/unitpages/unit24.html)	[[AAO unit 24, pages 1-11]](https://www.learner.org/courses/againstalldds/unitpages/unit24.html)
Tests of Significance and P-values	[AAO unit 25](https://www.learner.org/courses/againstalldds/unitpages/unit25.html)	[[AAO unit 25, pages 1-11]](https://www.learner.org/courses/againstalldds/unitpages/unit25.html)

3.4 One-sample Inference

Topic	Video	Readings
Means	[AAO unit 26](https://www.learner.org/courses/againstalldds/unitpages/unit26.html)	[[AAO unit 26, pages 1-11]](https://www.learner.org/courses/againstalldds/unitpages/unit26.html)
Proportions	[AAO unit 28](https://www.learner.org/courses/againstalldds/unitpages/unit28.html)	[[AAO unit 28, pages 1-11]](https://www.learner.org/courses/againstalldds/unitpages/unit28.html)
Rates	–	–

3.5 Two-sample Inference

Topic	Video	Readings
Means	[AAO unit 27](https://www.learner.org/courses/againstalldds/unitpages/unit27.html)	[[AAO unit 27, pages 1-11]](https://www.learner.org/courses/againstalldds/unitpages/unit27.html)
Proportions, Fisher's Exact	–	[[B&M chapter 20, pages 1-11]](https://www.barnesandnoble.com/read/9780393978222/chapter20)
Rates	–	–

3.6 Regression

Topic	Video	Readings
Linear Regression	[AAO unit 30](https://www.learner.org/courses/againstalldds/unitpages/unit30.html)	[[AAO unit 30, pages 1-11]](https://www.learner.org/courses/againstalldds/unitpages/unit30.html)
Logistic, Poisson Regression	–	[[OS notes]](https://www.learner.org/courses/againstalldds/unitpages/unit30.html)

3.7 Nonparametric Statistics

Topic	Video	Readings	Exercises
Wilcoxon Signed Rank Test	–	[[JH notes]](http://www.medicine.mcgill.ca/epidemiology/hanley/c607/ch14/jh_ch_14.pdf)	–
Kruskal-Wallis Test	–	[[EM notes]](https://www.dropbox.com/s/hotrocv75sm7q8/InfStatPart5.pdf?dl=0)	–

Chapter 4

Prerequisites

4.1 Git

You need to first install the [git](#) version control system. Follow [Chapter 1: Installing Git](#) for step-by-step installation instructions with screenshots.

4.2 R and RStudio

Complete the following DataCamp courses:

Topic	DataCamp Courses
Working with the RStudio IDE (Part 1)	This short course will guide you through installing both R and RStudio . RStudio is a software application that facilitates how you interact with R.
Introduction to R	In this course you will get a hands-on introduction to the basic commands in R. With the knowledge gained in this course, you will be ready to perform a data analysis.
Reporting with R Markdown	You will learn how to create reproducible reports using R and Markdown. All assignments for this course must be submitted in this format.
Version Control with RStudio IDE (Chapter 2 only)	You will learn how to use RStudio to version control your code. All assignments for this course must be submitted to a GitHub repository.

Chapter 5

Schedule

5.1 Week 1

5.1.1 Thursday September 6

[1. Introduction to EPIB607](https://docs.google.com/presentation/d/15c0YIS2KJXFzTKgFfQ_xDjTAcvPyQb8JhSLGvsEHJ6o/edit?usp=sh)
[2. Course Website](https://sahirbhatnagar.com/EPIB607/)
[3. Teaching Philosophy](https://sahirbhatnagar.com/EPIB607/course-information.html#teaching-strategy)
[4. Video: What is Statistics?](https://www.learner.org/courses/againstalldds/unitpages/unit01.html)
[5. Introduction to DALITE](https://mydalite.org/en/live/signup/form/NTc4)
[6. Review of A1 and DALITE Q1](https://sahirbhatnagar.com/EPIB607/assignments.html)
[7. Live Poll](http://etc.ch/tfZA)
[8. Terms and Concepts group exercise](https://sahirbhatnagar.com/EPIB607/terms-and-concepts.html)

5.2 Week 2

5.2.1 Monday September 10

[1. McGill VPN](http://kb.mcgill.ca/kb/?ArticleId=1212&source=article&c=12&cid=2#tab:homeTab:crumb:8:artId:1212:src:article)
[2.1 Data graphics Slides](https://docs.google.com/presentation/d/1wXgcTzcRKl_LeGRfNZjWWPkjwJSTIZSXBCI-fFuLEaE/edit?usp=sharing)
[2.2 Data graphics exercise](https://sahirbhatnagar.com/EPIB607/data-graphics.html)
[3. Introduction to the 'mosaic' package](https://cran.r-project.org/package=mosaic)
[4. R Markdown Presentation](https://prezi.com/dvmgx17e_was/reproducible/?utm_campaign=share&utm_medium=copy)

5.2.2 Wednesday September 12

[1. DALITE Q1 - Histograms, Medians and Means due by 5pm](https://github.com/sahirbhatnagar/EPIB607/raw/master/dalite/001_hist_m)
--

5.2.3 Thursday September 13

1. Histograms, Medians and Means

5.3 Week 3

5.3.1 Monday September 17

-
1. Boxplots and Standard Deviation
 2. Parameters and Statistics
 3. Sampling Distributions and Central Limit Theorem
-

5.3.2 Thursday September 20

[Assignment 1 - Setting up the computing environment due by 5pm](<https://github.com/sahirbhatnagar/EPIB607/raw/master/assignment1>)

1. Bootstrap

5.3.3 Friday September 21

Assignment 2 - Histograms, Medians, Means, Boxplots and Standard Deviation due by 11:59pm

5.4 Week 4

5.4.1 Monday September 24

5.4.2 Thursday September 27

-
1. The Bootstrap
-

5.4.3 Friday September 28

Assignment 3 due by 11:59pm

5.5 Week 5

5.5.1 Monday October 1

5.5.2 Thursday October 4

5.5.3 Friday October 5

Assignment 4 due by 11:59pm

5.6 Week 6

5.6.1 Monday October 8

Thanksgiving. No class.

5.6.2 Thursday October 11

—
—

5.6.3 Friday October 12

Assignment 5 due by 11:59pm

5.7 Week 7

5.7.1 Monday October 15

—
—

5.7.2 Thursday October 18

—
—

5.7.3 Friday October 19

Assignment 6 due by 11:59pm

5.8 Week 8

5.8.1 Monday October 22

—
—

5.8.2 Thursday October 25

5.9 Week 9**5.9.1 Monday October 29**

Midterm

5.9.2 Thursday November 1

5.9.3 Friday November 2

Assignment 7 due by 11:59pm

5.10 Week 10**5.10.1 Monday November 5**

5.10.2 Thursday November 8

5.10.3 Friday November 9

Assignment 8 due by 11:59pm

5.11 Week 11**5.11.1 Monday November 12**

5.11.2 Thursday November 15

—
—

5.11.3 Friday November 16

Assignment 9 due by 11:59pm

5.12 Week 12

5.12.1 Monday November 19

—
—

5.12.2 Thursday November 22

—
—

5.12.3 Friday November 23

Assignment 10 due by 11:59pm

5.13 Week 13

5.13.1 Monday November 26

—
—

5.13.2 Thursday November 29

—
—

5.14 Week 14

5.14.1 Monday December 3

—
—

5.14.2 Tuesday December 4

Final Exam (Tentative).

5.14.3 Friday December 7

Group project due by 11:59pm.

Chapter 6

Slides

1. [Introduction to EPIB 607](#)

Chapter 7

Assignments

Assignment	Topic
1	[Setting up the computing environment](https://github.com/sahirbhatnagar/EPIB607/raw/master/assignments/a1/a1-setup)
2	Histograms, Means, Medians, Boxplots, Standard Deviation
3	Sampling Distributions and Central Limit Theorem
4	The Bootstrap
5	Confidence Intervals
6	Tests of Significance and p-values
7	One-Sample Inference
8	Two-Sample Inference
9	Regression
10	Nonparametric Statistics

Chapter 8

DALITE

1. [Q1 \(due September 12, 2018\)](#)

Chapter 9

Terms and Concepts

In groups of approximately 10, pick an article from the list below and fill out this [Google Spreadsheet](#)

The first sheet contains an example based on this article: [The efficacy of calorie labelling formats on pre-packaged foods: An experimental study among adolescents and young adults in Canada](#)

-
1. [Blood Lead Levels of Children in Flint, Michigan: 2006-2016](#)
 2. [Brief Standing Desk Intervention to Reduce Sedentary Behavior at a Physical Activity Conference in 2016](#)
 3. [Lunch is ready ... but not healthy: An analysis of lunches served in childcare centres in two Canadian provinces](#)
 4. [Fluoride exposure and reported learning disability diagnosis among Canadian children: Implications for community water fluoridation](#)
 5. [Folate Nutrition Status in Mothers of the Boston Birth Cohort, Sample of a US Urban Low-Income Population](#)
 6. [State Indoor Tanning Laws and Prevalence of Indoor Tanning Among US High School Students, 2009–2015](#)
 7. [Outdoor time, physical activity and sedentary time among young children: The 2012–2013 Canadian Health Measures Survey](#)
 8. [Physicochemical properties and phenolic content of honey from different floral origins and from rural versus urban landscapes](#)
 9. [A family-centered lifestyle intervention for obese six- to eight-year-old children: Results from a one-year randomized controlled trial conducted in Montreal, Canada](#)

Part II

Part I

Chapter 10

Data Graphics

Data graphics can be understood in terms of five basic elements ([Baumer et al., 2017](#)):

1. visual cues
2. coordinate system
3. scale
4. context
5. facets and layers

10.1 Visual Cues

Visual cues are graphical elements that draw the eye to what you want your audience to focus upon. They are the fundamental building blocks of data graphics, and the choice of which visual cues to use to represent which quantities is the central question for the data graphic composer. Nathan Yau identifies nine distinct visual cues, for which we also list whether that cue is used to encode a numerical or categorical quantity:

1. Position (numerical): where in relation to other things?
2. Length (numerical): how big (in one dimension)?
3. Angle (numerical): how wide? parallel to something else?
4. Direction (numerical) at what slope? In a time series, going up or down?
5. Shape (categorical) belonging to which group?
6. Area (numerical) how big (in two dimensions)?
7. Volume (numerical) how big (in three dimensions)?
8. Shade (either) to what extent? how severely?
9. Color (either) to what extent? how severely? Beware of red/green color blindness

10.2 Coordinate systems

How are the data points organized? While any number of coordinate systems are possible, three are most common.

10.2.1 Cartesian

This is the familiar (x, y) -rectangular coordinate system with two perpendicular axes.

10.2.2 Polar

The radial analog of the Cartesian system with points identified by their radius ρ and angle θ .

10.2.3 Geographic

This is the increasingly important system in which we have locations on the curved surface of the Earth, but we are trying to represent these locations in a flat two-dimensional plane.

10.3 Scale

Scales translate values into visual cues. The choice of scale is often crucial. The central question is how does distance in the data graphic translate into meaningful differences in quantity? Each coordinate axis can have its own scale, for which we have three different choices:

10.3.1 Numeric

A numeric quantity is most commonly set on a linear, logarithmic, or percent- age scale. Note that a logarithmic scale does not have the property that, say, a one-centimeter difference in position corresponds to an equal difference in quantity anywhere on the scale.

10.3.2 Categorical

A categorical variable may have no ordering (e.g., Democrat, Republican, or Independent), or it may be ordinal (e.g., never, former, or current smoker).

10.3.3 Time

Time is a numeric quantity that has some special properties. First, because of the calendar, it can be demarcated by a series of different units (e.g., year, month, day, etc.). Second, it can be considered periodically (or cyclically) as a “wrap-around” scale. Time is also so commonly used and misused that it warrants careful consideration.

10.4 Context

The purpose of data graphics is to help the viewer make meaningful comparisons, but a bad data graphic can do just the opposite: It can instead focus the viewer’s attention on meaningless artifacts, or ignore crucial pieces of relevant but external knowledge. Context can be added to data graphics in the form of titles or subtitles that explain what is being shown, axis labels that make it clear how units and scale are depicted, or reference points or lines that contribute relevant external information. While one should avoid cluttering up a data graphic with excessive annotations, it is necessary to provide proper context.

10.5 Facets and layers

One of the fundamental challenges of creating data graphics is condensing multivariate information into a two-dimensional image. While three-dimensional images are occasionally useful, they are often more confusing than anything else. Instead, here are three common ways of incorporating more variables into a two-dimensional data graphic.

10.5.1 Facets

A single data graphic can be composed of several small multiples of the same basic plot, with one (discrete) variable changing in each of the small sub-images.

10.5.2 Layers

It is sometimes appropriate to draw a new layer on top of an existing data graphic. This new layer can provide context or comparison, but there is a limit to how many layers humans can reliably parse.

10.5.3 Animation

If time is the additional variable, then an animation can sometimes effectively convey changes in that variable. Of course, this doesn't work on the printed page, and makes it impossible for the user to see all the data at once.

10.6 Color

Approximately 8 percent of the population—most of whom are men—have some form of color blindness. Most commonly, this renders them incapable of seeing colors accurately, most notably of distinguishing between red and green. Compounding the problem, many of these people do not know that they are color-blind. Thus, for professional graphics it is worth thinking carefully about which colors to use. The NFL famously failed to account for this in a 2015 game in which the Buffalo Bills wore all-red jerseys and the New York Jets wore all-green, leaving colorblind fans unable to distinguish one team from the other!

Thankfully, we have been freed from the burden of having to create such intelligent palettes by the research of Cynthia Brewer, creator of the [ColorBrewer website](#) (and [R package](#)). Brewer has created colorblind-safe palettes in a variety of hues for three different types of numeric data in a single variable:

10.6.1 Sequential

The ordering of the data has only one direction. Positive integers are sequential because they can only go up: they can't go past 0. (Thus, if 0 is encoded as white, then any darker shade of gray indicates a larger number.)

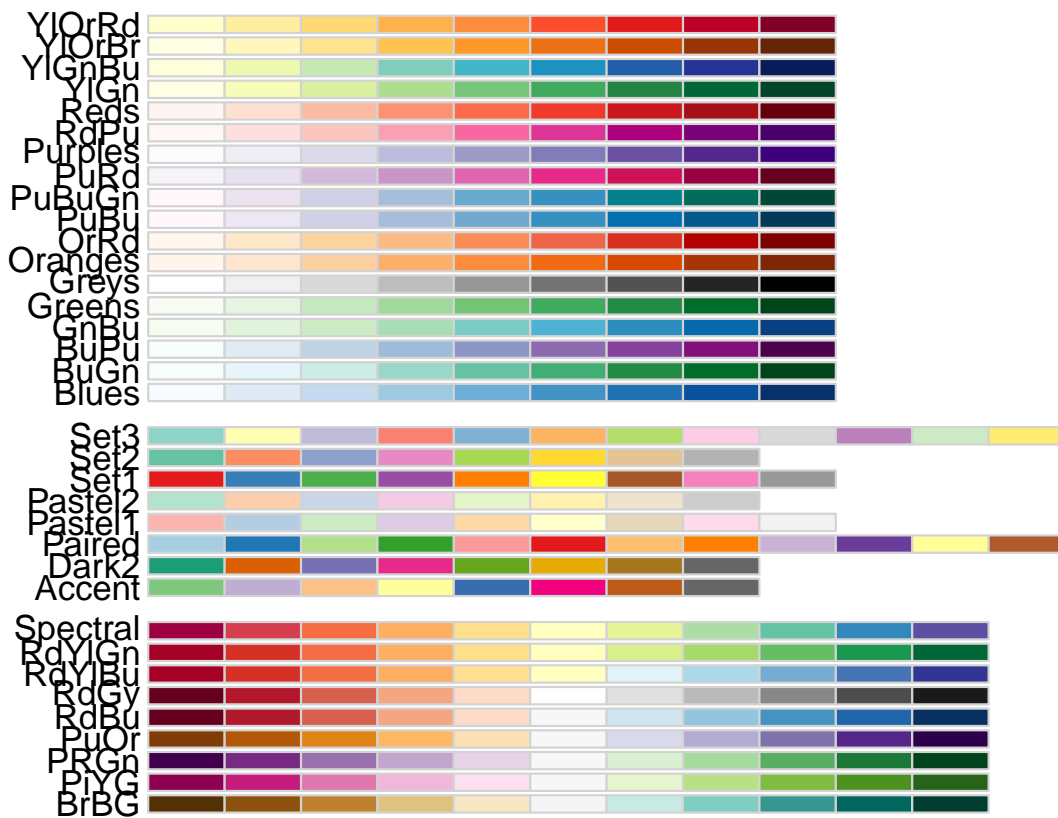
10.6.2 Diverging

The ordering of the data has two directions. In an election forecast, we commonly see states colored based on how they are expected to vote for the president. Since red is associated with Republicans and blue with Democrats, states that are solidly red or blue are on opposite ends of the scale. But “swing states” that could go either way may appear purple, white, or some other neutral color that is “between” red and blue.

10.6.3 Qualitative

There is no ordering of the data, and we simply need color to differentiate different categories.

```
pacman::p_load(RColorBrewer)
RColorBrewer::display.brewer.all()
```



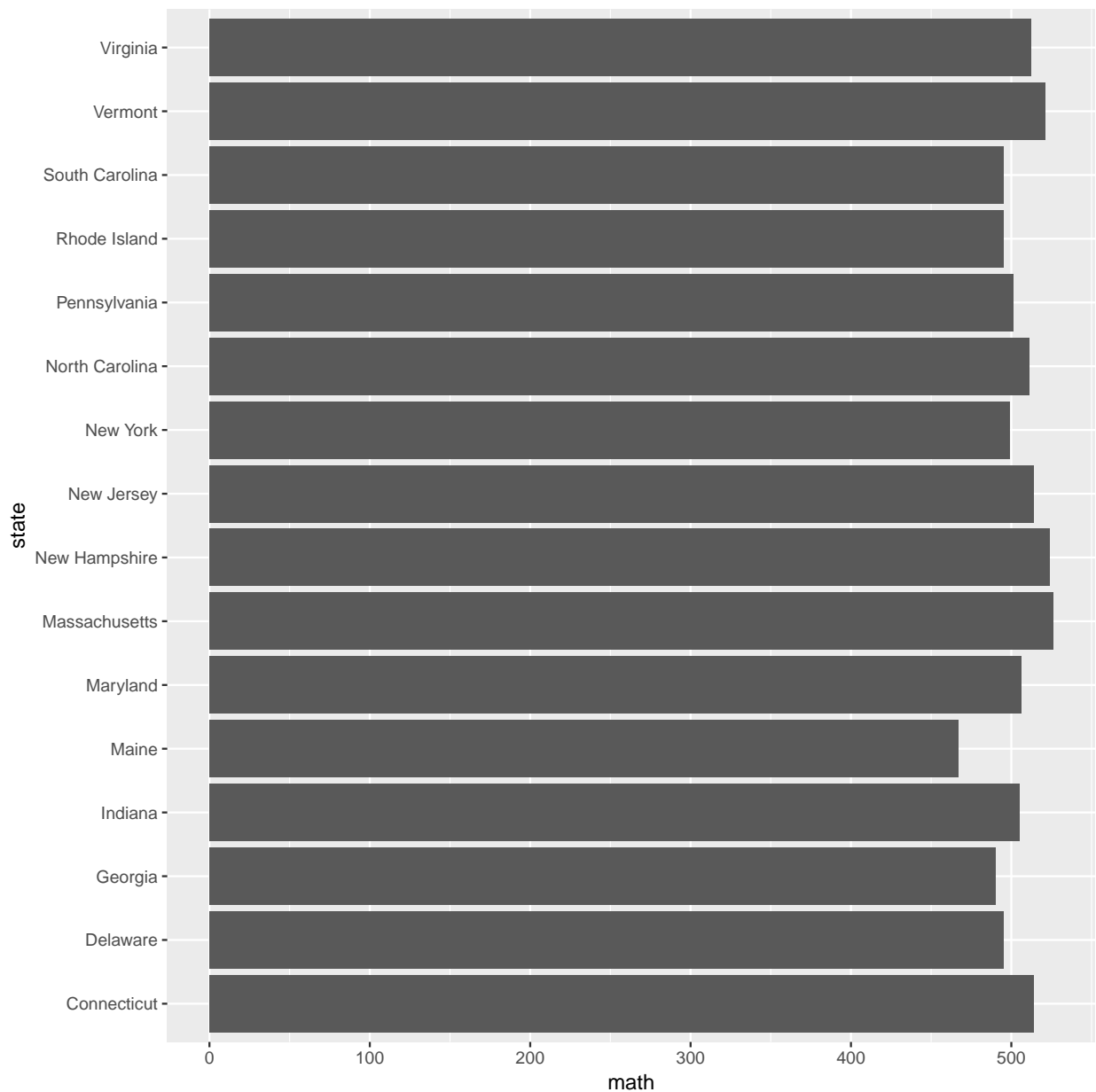
Also see the [viridis R package](#).

10.7 Examples

10.7.1 SAT Scores

The bar graph below displays the average score on the math portion of the 2010 SAT (with possible scores ranging from 200 to 800) among states for whom at least two-thirds of the students took the SAT.

```
pacman::p_load(mdsr)
pacman::p_load(mosaic)
gf_colh(state ~ math, data = subset(SAT_2010, sat_pct > 66))
```



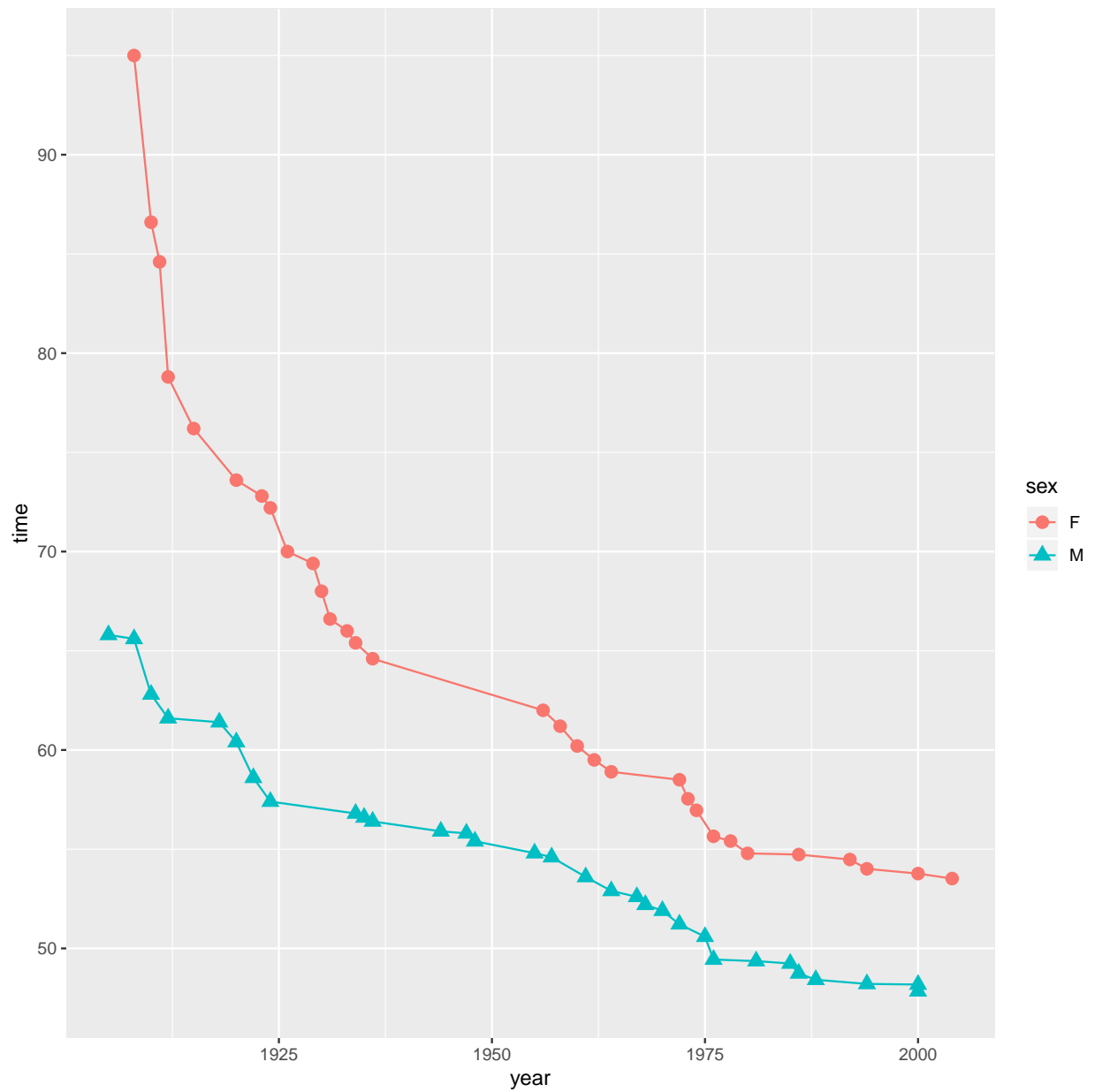
This plot uses the visual cue of position to represent the math SAT score on the vertical axis with a linear scale. The categorical variable of state is arrayed on the horizontal axis. Although the states are ordered alphabetically, it would not

be appropriate to consider the state variable to be ordinal, since the ordering is not meaningful in the context of math SAT scores. The coordinate system is Cartesian, although as noted previously, the horizontal coordinate is meaningless. Context is provided by the axis labels and title.

10.7.2 Swimming records

Next, we consider a time series that shows the progression of the world record times in the 100-meter freestyle swimming event for men and women. The Figure below displays the times as a function of the year in which the new record was set. At some level this is simply a scatterplot that uses position on both the vertical and horizontal axes to indicate swimming time and chronological time, respectively, in a Cartesian plane. The numeric scale on the vertical axis is linear, in units of seconds, while the scale on the horizontal axis is also linear, measured in years. But there is more going on here. Color is being used as a visual cue to distinguish the categorical variable sex. Furthermore, since the points are connected by lines, direction is being used to indicate the progression of the record times. (In this case, the records can only get faster, so the direction is always down.) One might even argue that angle is being used to compare the descent of the world records across time and/or gender. In fact, in this case shape is also being used to distinguish sex.

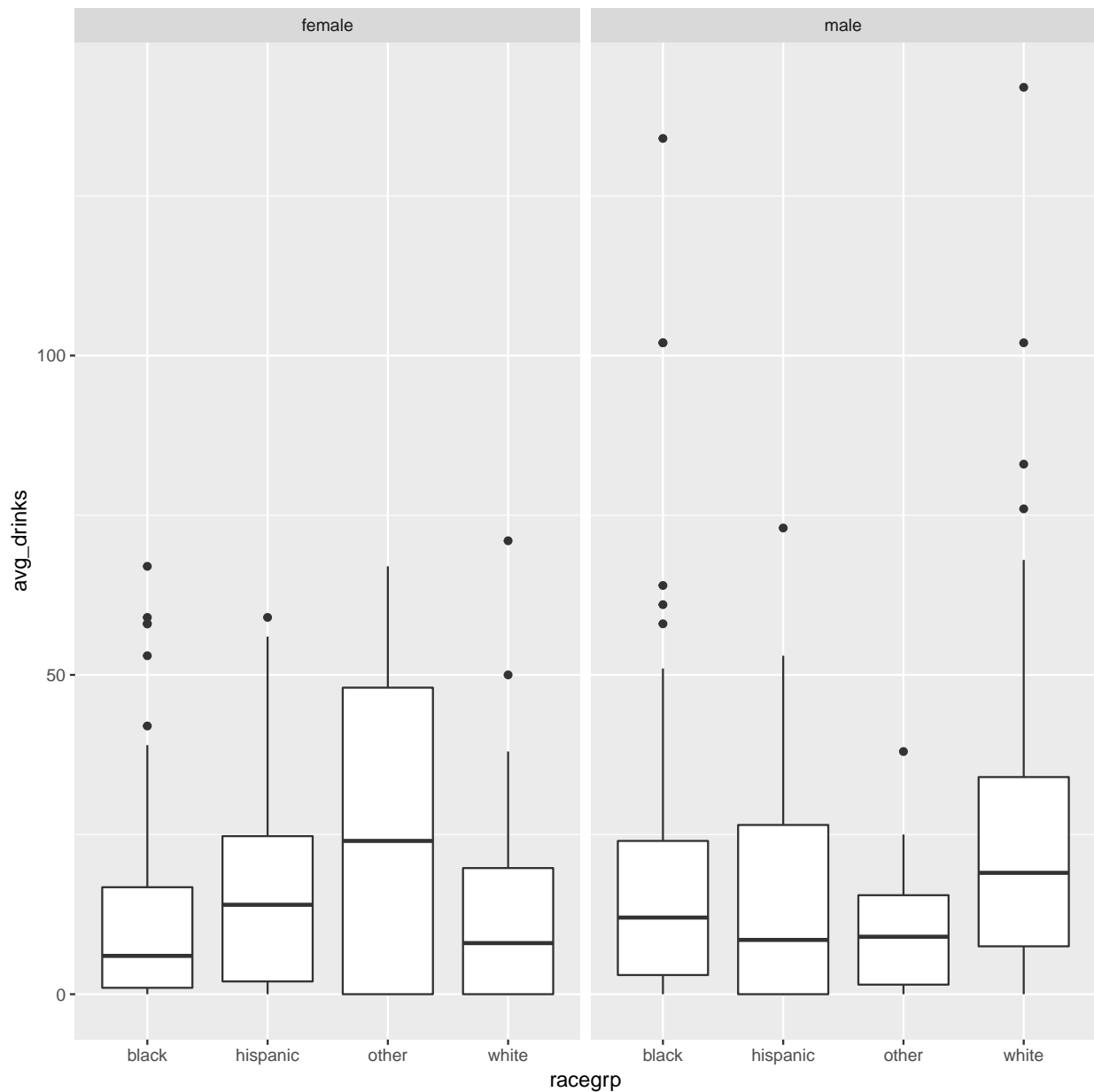
```
pacman::p_load(mosaicData)
gf_point(time ~ year , data = SwimRecords, color = ~ sex, shape = ~ sex, size = 3) %>%
  gf_line(time ~ year , data = SwimRecords, color = ~ sex)
```



10.7.3 HELP study

The HELP study was a clinical trial for adult inpatients recruited from a detoxification unit. Patients with no primary care physician were randomized to receive a multidisciplinary assessment and a brief motivational intervention or usual care, with the goal of linking them to primary medical care.

```
gf_boxplot(avg_drinks ~ racegrp | sex, data = HELPrct)
```



10.8 Exercises

10.8.1 Exercise 1

For this exercise, refer to the article assigned to your team in the [Terms and Concepts](#) exercise.

1. Identify the aesthetics and scale(s). If your article did not contain a graphic, pick a Table in your paper and think about what graphic could have been used instead. See [Fundamentals of Data Visualization](#) and [Top 50 ggplot2 Visualizations](#) for examples of graphics.

2. How many variables are depicted in the graphic? Explicitly link each variable to an aesthetic that you listed above.
3. Critique this data graphic using the taxonomy described in this chapter.

10.8.2 Exercise 2

Vox published a list of [Charts that explain food in America](#). There are 40 maps, charts, and graphs that show where our food and drink comes from and how we eat it.

Pick your best and least favorite graphic. One representative from each group will present in 1 minute or less their rationale for the groups choices.

Appendix A

Vectorization, *apply and for loops

This section will cover the basics of vectorizations, the `*apply` family of functions and `for` loops.

A.1 Vectorization

Almost everything in R is a vector. A scalar is really a vector of length 1 and a `data.frame` is a collection of vectors. An nice feature of R is its vectorized capabilities. Vectorization indicates that a function operates on a whole vector of values at the same time and not just on a single value¹. If you have ever taken a basic linear algebra course, this concept will be familiar to you.

Take for example two vectors:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

The corresponding R code is given by:

```
a <- c(1,2,3)
b <- c(1,2,3)
a+b
#> [1] 2 4 6
```

Many of the base functions in R are already vectorized. Here are some common examples:

```
# generate a sequence of numbers from 1 to 10
(a <- 1:10)
#> [1] 1 2 3 4 5 6 7 8 9 10

# sum the numbers from 1 to 10
sum(a)
#> [1] 55
```

¹<http://www.dummies.com/how-to/content/how-to-vectorize-your-functions-in-r.html>

```
# calculate sums of each column
colSums(iris[, -5])
#> Sepal.Length Sepal.Width Petal.Length Petal.Width
#>      876.5      458.6      563.7      179.9
```

Exercise: What happens when you sum two vectors of different lengths?

A.2 Family of *apply functions

- `apply`, `lapply` and `sapply` are some of the most commonly used class of functions in R
- *apply functions are not necessarily faster than loops, but can be easier to read (and vice versa)
- `apply` is used when you need to perform an operation on every row or column of a matrix or data.frame
- `lapply` and `sapply` differ in the format of the output. The former returns a list while the latter returns a vector
- There are other *apply functions such as `tapply`, `vapply` and `mapply` with similar functionality and purpose

A.2.1 Loops vs. Apply

```
# Getting the row means of two columns
# Generate data
N <- 10000
x1 <- runif(N)
x2 <- runif(N)
d <- as.data.frame(cbind(x1, x2))
head(d)
#>      x1      x2
#> 1 0.57632 0.9615
#> 2 0.56474 0.1950
#> 3 0.07399 0.3001
#> 4 0.45387 0.3823
#> 5 0.37328 0.1197
#> 6 0.33132 0.9891

# Loop:
# create a vector to store the results in
rowMeanFor <- vector("double", N)

for (i in seq_len(N)) {
  rowMeanFor[[i]] <- mean(c(d[i, 1], d[i, 2]))
}

# Apply:
rowMeanApply <- apply(d, 1, mean)

# are the results equal
all.equal(rowMeanFor, rowMeanApply)
#> [1] TRUE
```

A.2.2 Descriptive Statistics using *apply

```
data(women)
# data structure
str(women)
#> 'data.frame': 15 obs. of 2 variables:
#> $ height: num 58 59 60 61 62 63 64 65 66 67 ...
#> $ weight: num 115 117 120 123 126 129 132 135 139 142 ...

# calculate the mean for each column
apply(women, 2, mean)
#> height weight
#> 65.0 136.7

# apply 'fivenum' function to each column
vapply(women, fivenum, c("Min." = 0, "1st Qu." = 0, "Median" = 0,
                        "3rd Qu." = 0, "Max." = 0))
#>      height weight
#> Min.      58.0 115.0
#> 1st Qu.    61.5 124.5
#> Median     65.0 135.0
#> 3rd Qu.    68.5 148.0
#> Max.      72.0 164.0
```

A.2.3 Creating new columns using sapply

You can apply a *user defined function* to columns or the entire data frame:

```
# the output of sapply is a vector
# the 's' in sapply stands for 'simplified' apply
mtcars$gear2 <- sapply(mtcars$gear,
                      function(i) if (i==4) "alot" else "some")

head(mtcars)[,c("gear", "gear2")]
#>      gear gear2
#> Mazda RX4      4 alot
#> Mazda RX4 Wag  4 alot
#> Datsun 710      4 alot
#> Hornet 4 Drive  3 some
#> Hornet Sportabout 3 some
#> Valiant        3 some
```

A.2.4 Applying functions to subsets using tapply

```
# Fisher's famous dataset
data(iris)
```

```
str(iris)
#> 'data.frame':   150 obs. of  5 variables:
#>  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
#>  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
#>  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
#>  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
#>  $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

# mean sepal length by species
tapply(iris$Sepal.Length, iris$Species, mean)
#>      setosa versicolor virginica
#>      5.006      5.936      6.588
```

A.2.5 Nested for loops using mapply

mapply is my favorite base R function and here are some reasons why:

- Using mapply is equivalent to writing nested for loops except that it is 100% more human readable and less prone to errors
- It is an effective way of conducting simulations because it iterates of many arguments

Let's say you want to generate random samples from a normal distribution with varying means and standard deviations. Of course the brute force way would be to write out the command once, copy paste as many times as you want, and then manually change the arguments for mean and sd in the rnorm function as so:

```
v1 <- rnorm(100, mean = 5, sd = 1)
v2 <- rnorm(100, mean = 10, sd = 5)
v3 <- rnorm(100, mean = -3, sd = 10)
```

This isn't too bad for three vectors. But what if you want to generate many more combinations of means and sds ? Furthermore, how can you keep track of the parameters you used? Now lets consider the mapply function:

```
means <- c(5,10,-3) ; sds <- c(1,5,10)

# MoreArgs is a list of arguments that dont change
randomNormals <- mapply(rnorm, mean = means, sd = sds,
                        MoreArgs = list(n = 100))

head(randomNormals)
#>      [,1] [,2] [,3]
#> [1,] 3.836 8.771 5.144
#> [2,] 4.525 9.376 -2.280
#> [3,] 4.072 13.144 2.940
#> [4,] 4.737 18.210 -13.118
#> [5,] 5.690 22.951 -7.008
#> [6,] 4.826 7.615 -15.323
```

The following diagram (from [r4ds](#)) describes exactly what is going on in the above function call to mapply:

Advantages:

1. Result is automatically stored in a matrix

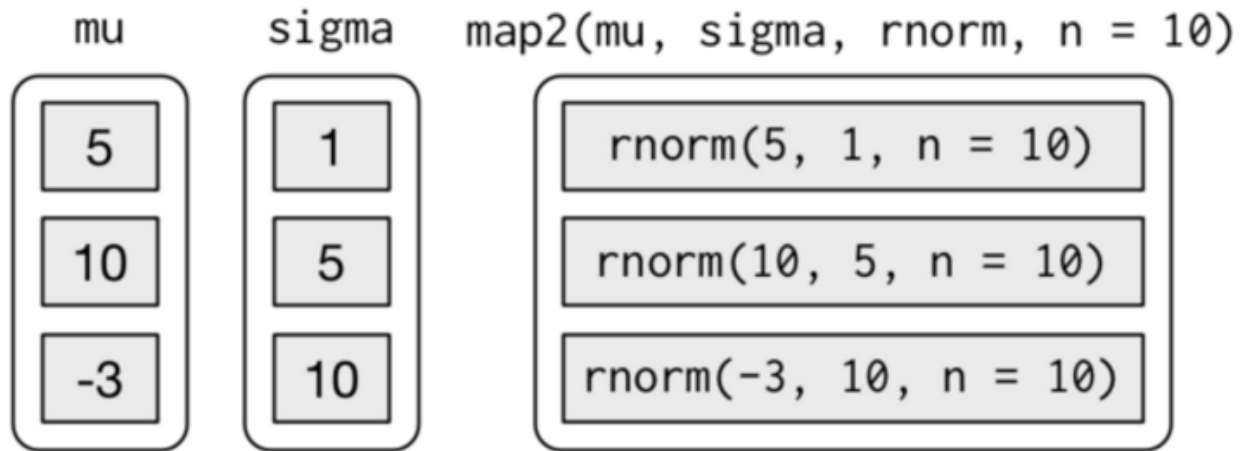


Figure A.1

2. The parameters are also saved in R objects so that they can be easily manipulated and/or recovered

Consider a more complex scenario where you want to consider many possible combinations of means and sds. We take advantage of the `expand.grid` function to create a `data.frame` of simulation parameters:

```
simParams <- expand.grid(means = 1:10,
                        sds = 1:10)

randomNormals <- mapply(rnorm, mean = simParams$means,
                        sd = simParams$sds,
                        MoreArgs = list(n = 100))

dim(randomNormals)
#> [1] 100 100
```

A.3 Creating dynamic documents with `mapply`

`mapply` together with the `rmarkdown` package (Allaire et al., 2018) can be very useful to create dynamic documents for exploratory analysis. We illustrate this using the Motor Trend Car Road Tests data which comes pre-loaded in R.

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

Copy the code below in a file called `mapplyRmarkdown.Rmd`:

Copy the code below in a file called `boxplotTemplate`:

Appendix B

Appendix B

Bibliography

- Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W. (2018). *rmarkdown: Dynamic Documents for R*. R package version 1.10.
- Baumer, B. S., Kaplan, D. T., and Horton, N. J. (2017). *Modern Data Science with R*. Chapman and Hall/CRC Press: Boca Raton.
- Bhatnagar, S., Lasry, N., Desmarais, M., and Charles, E. (2016). Dalite: Asynchronous peer instruction for moocs. In *European Conference on Technology Enhanced Learning*, pages 505–508. Springer.