BADM 372 Applied Analytics

BADM 372

2022-03-29

Contents

Chapter 1

About this course

This website serves as headquarters for **BADM 372 Applied Analytics**.

Content here will be updated with any changes made during the semester, so if at any point you are told there was a change in the schedule or an assignment, you can come here to get the updated version.

Also, this website has benefited greatly from lots of free, readily available resources posted on the web and we leverage these extensively. I would encourage you to review these resources in your analytics journey. Some that we specifically use with great frequency are these (**and I say a loud THANK YOU to the authors!**):

- R for Data Science
- An Introduction to Statistical Learning with Applications in R
- Data Science in a Box
- stackoverflow.com, for example

Chapter 2

Syllabus

Instructor: Tobin Turner

Office Hours: mutually convenient time arranged by email e-mail: jtturner@presby.edu

2.1 Course Objectives and Learning Outcomes

This course is designed to introduce to data science. Students will apply statistical knowledge and techniques to both business and non-business contexts.

At the end of this course students should be able to:

By the end of the course, you will be able to…

- gain insight from data
- gain insight from data, reproducibly
- gain insight from data, reproducibly, using modern programming tools and techniques
- gain insight from data, reproducibly and collaboratively, using modern programming tools and techniques
- gain insight from data, reproducibly (with literate programming and version control) and collaboratively, using modern programming tools and techniques
- communicate results effectively

This course will be focused on both understanding and applying key business analytical concepts. Although the text serves as a useful foundation for the concepts covered in the class, simple memorization of the material in the text will not be sufficient. Class participation, discussion, and application are critical.

2.2 Text and Resources

- This course website (primary resource)
- R for Data Science
- An Introduction to Statistical Learning with Applications in R
- Data Science in a Box
- stackoverflow.com, for example
- Other free, publicly available datasets and publications.

2.3 Performance Evaluation (Grading)

- Quizzes and Assignments - 40%
- Exam 1 - 20%
- Exam 2 - 20%
- Final Exam - 20%

2.3.1 Exams

Exams will cover assigned chapters in the textbook, other assigned readings, lectures, class exercises, class discussions, videos, and guest speakers. I will typically allocate time prior to each exam to clearly identify the body of knowledge each test will cover and to answer questions about the format and objectives of the exam.

2.3.2 Quizzes – DON'T MISS CLASS

- The average of all quizzes and assignments will comprise the Quizzes and Assignments - 40% portion of your final grade
- Quizzes and Assignments are designed to prepare you for your exams and to ensure you stay up with the course material
- **Missed Quizzes and Assignments cannot be made up later. Be present.**

Quizzes rule. **LISTEN. - Missed Quizzes and Assignments cannot be made up later. Be present.**

2.3.3 Final Average

- Final Average Grade
  - 90-100 A

- 88-89 B+
- 82-87 B+
- 80-81 B-
- 78-79 C+
- 72-77 C+
- 70-71 C-
- 60-69 D
- 59 and below F

2.4 Class Participation:

I will frequently give readings or assignments for you to complete prior to the next class meeting. I expect you to fully engage the material: answer questions, pose questions, provide insightful observations. Keep in mind that quality is an important component in "participation." Periodic cold calls will take place. I will also put students in the "hot seat" on occasion. In these class sessions, I may select a random group of students to lead us in the discussion and debate. Because the selection of participants will not be announced until class begins, everyone will be expected to prepare for the discussion. Reading the assigned chapters and articles are the best way to prepare for the discussion. If you have concerns about being called on in class, please see me to discuss. The purpose of the "hot seat" is not to stress or embarrass students, but to encourage students to actively engage the material.

2.5 Phones

Phones are not allowed to be used in class without the instructor's prior consent. If you have a need of a phone during class please let me know before class. Unauthorized use of electronic devices may result in the lowering of the grade or dismissal from the class. **I mean this.**

The phone thing? I mean this.

2.6 Attendance

You are expected to be regular and punctual in your class attendance. Students are responsible for all the material missed and homework assignments made. If class is missed, notes/homework should be obtained from another student. If I am more than 15 minutes late, class is considered cancelled. No more than 4 absences are allowed during a semester. Exceeding the absence policy may result in receiving an F for the course. The professors roll is the official roll and students not present when roll is taken will be counted as absent. If a student must miss an exam, she or he must work out an agreeable time with the instructor to take the test prior to the exam being given. If a student misses a test due to an emergency, the student must inform the instructor as soon as is possible. In special cases, the instructor may allow the student to take a make-up exam.

2.7 Accommodations

Presbyterian College is committed to providing reasonable accommodations for all students with documented disabilities. If you are seeking academic accommodations under the Americans with Disabilities Act, you must register with the Academic Success Office, located on 5th Avenue (beside Campus Police). To receive these accommodations, please obtain the proper Accommodations Approval Form from that office, and then meet with me at the beginning of the semester to discuss how we may deliver your approved accommodations. I especially encourage you to meet with me well in advance of the actual accommodations being provided, as it may not be feasible to offer immediate accommodations without sufficient advance notice (such as in the case of tests). I can assure you that all discussions will remain confidential. Disability Services information is located at this link http://bit.ly/PCdisabilityservices

Additionally, it is the student's responsibility to give the instructor one week's notice prior to each instance where accommodation will be required.

2.8 Honor Code and Plagiarism:

All assignments/exams must be your own work. Any copying or use of unauthorized assistance will be treated as a violation of PC's Honor Code. If you are unsure of what resources are allowed, please ask. Please note that all text longer than 7 words taken from ANY other source must be placed in quotations and cited. Also, summarizing ANY other source must also be cited. Using ANY other source and showing work to be your own is a violation of plagiarism and the honor code.

2.9 First-Generation Version:

I am a Presby First+ Advocate. I am here to support our current first-generation students. At Presbyterian College, first-generation students are those in which neither parent nor legal guardian graduated from a four-year higher education institution with a bachelor's degree. If you are a first-generation college student, please contact me. For more information about support for first-generation college students on our campus visit our Presby First+ webpage.

2.10 Continuing Advocate Version

I am a Presby First+ Advocate. I am committed to supporting first-generation students at Presbyterian College. At Presbyterian College, first-generation students are those in which neither parent nor legal guardian graduated from a four-year higher education institution with a bachelor's degree. If you are a first-generation college student, please contact me anytime or visit me during my office hours. For more information about support for first-generation college students on our campus visit our Presby First+ webpage.

Chapter 3

Our Class Rhythm

Monday: Wrap up previous topic and introduce what you've pre-read about. Chat. Play. Work some examples. Make sure the topics applies to real-life.

Wednesday: Work more examples. Chat as needed. **Live our best lives. :)**.

Friday: Apply what we've learned – demonstrate your mastery (typically in the form of a quiz, lab, or assignment). Rinse. Repeat.

Chapter 4

End in Mind

Dana Simmons: “Can you predict which students will enroll at PC?”

Christina Miller: ??? Well, can you? ???

Chapter 5

Schedule

This is a tentative schedule, and it will change. **BUT** I will do my very best to review this often so that we all stay on the same page and so that you may plan accordingly!

Spring 2022

Date	Topic
Monday, January 10, 2022	Intro and A1 review
Wednesday, January 12, 2022	Rmarkdown
Friday, January 14, 2022	Lab 1: Rmarkdown
Monday, January 17, 2022	MLK Holiday
Wednesday, January 19, 2022	ggplot
Friday, January 21, 2022	Lab 2: ggplot
Monday, January 24, 2022	EDA & ggplot
Wednesday, January 26, 2022	EDA & ggplot
Friday, January 28, 2022	Lab 3: EDA & ggplot
Monday, January 31, 2022	TIDY SPREAD AND GATHER (R4DS CH 9 DPLYR)
Wednesday, February 2, 2022	RELATIONAL DATA (R4DS CH 10 DPLYR)
Friday, February 4, 2022	QUIZ
Monday, February 7, 2022	STRINGS (R4DS CH 11 stringr)
Wednesday, February 9, 2022	STRINGS (R4DS CH 12 factors)
Friday, February 11, 2022	QUIZ
Monday, February 14, 2022	Dates and Times
Wednesday, February 16, 2022	Dates and Times
Friday, February 18, 2022	QUIZ
Monday, February 21, 2022	Functions
Wednesday, February 23, 2022	Functions

Date	Topic
Friday, February 25, 2022	QUIZ
Monday, February 28, 2022	Itertation
Wednesday, March 2, 2022	Itertation
Friday, March 4, 2022	QUIZ
Monday, March 7, 2022	LAUNCH PROJECT
Wednesday, March 9, 2022	INDEPENDENT PROJECT
Friday, March 11, 2022	INDEPENDENT PROJECT
Monday, March 14, 2022	SPRING BREAK
Wednesday, March 16, 2022	SPRING BREAK
Friday, March 18, 2022	SPRING BREAK
Monday, March 21, 2022	INDEPENDENT PROJECT
Wednesday, March 23, 2022	PRESENTATIONS
Friday, March 25, 2022	PRESENTATIONS
Monday, March 28, 2022	Model Builiding/ADVISING WEEK
Wednesday, March 30, 2022	Model Builiding/ADVISING WEEK
Friday, April 1, 2022	QUIZ
Monday, April 4, 2022	regrssion
Wednesday, April 6, 2022	stepwise addition/deletion
Friday, April 8, 2022	QUIZ
Monday, April 11, 2022	logistic regression
Wednesday, April 13, 2022	trees & forests
Friday, April 15, 2022	Easter Holidays
Monday, April 18, 2022	Easter Holidays
Wednesday, April 20, 2022	Model Builiding
Friday, April 22, 2022	QUIZ
Monday, April 25, 2022	PRESENTATIONS
Wednesday, April 27, 2022	PRESENTATIONS
Friday, April 29, 2022	LAST DAY
Monday, May 2, 2022	Final Exam 8:30 p.m. – F period

Chapter 6

Lab 1 Excercises

Let's make sure we feel good about BADM 371 material.

All open notes/internet/R4DS/etc., **but all work must be your own**.

Use the starwars data (dplyr package) to answer/do:

1. Who is the tallest individual? Shortest?
2. How many homeworlds are there?
3. Which homeworld has the most individuals? Fewest? Average # of idividuals per homeworld?
4. Make a plot of all individuals with mass on the x axis and height on the y axis.
5. Put a best fit line on this plot.
6. Who is the biggest outlier in this dataset?
7. Calculate BMI for all these individuals. What is the average BMI for all individuals?
8. What is the average BMI for each homeworld?
9. Which homeworlds have the greatest percentage of individuals with BMI's greater than the average you found in #8 above?
10. How many individuals have no missing data? Which variables have the most missing data?

Chapter 7

Lab 1 in Rmarkdown

7.1 R Markdown

```
library(dplyr)
```

1. Who is the tallest individual? Shortest?

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
#>    66.0   167.0   180.0   174.4   191.0   264.0       6
```

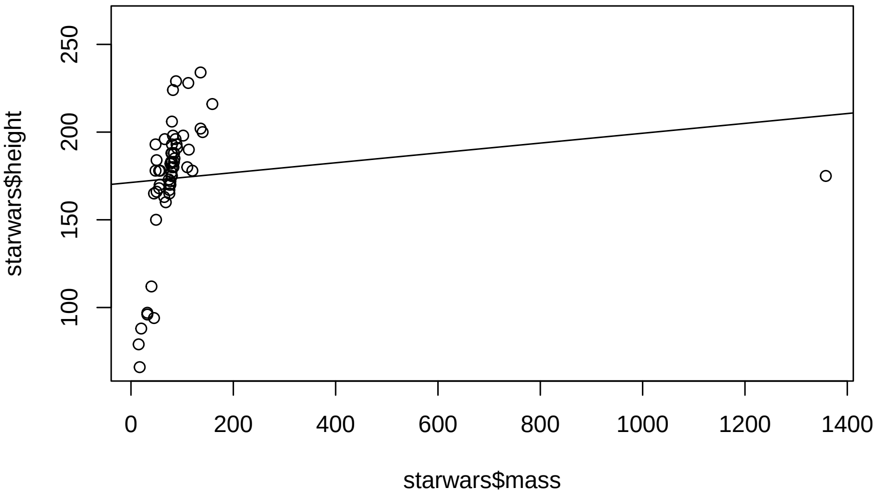
2. How many homeworlds are there?

```
#> # A tibble: 49 x 1
#>    homeworld
#>    <chr>
#>  1 Tatooine
#>  2 Naboo
#>  3 Alderaan
#>  4 Stewjon
#>  5 Eriadu
#>  6 Kashyyyk
#>  7 Corellia
#>  8 Rodia
#>  9 Nal Hutta
#> 10 Bestine IV
#> # ... with 39 more rows
```

3. Which homeworld has the most individuals? Fewest? Average # of individuals per homeworld?

```
#> # A tibble: 49 x 2
#>    homeworld     n
#>    <chr>     <int>
#>  1 Naboo        11
#>  2 Tatooine     10
#>  3 <NA>         10
#>  4 Alderaan      3
#>  5 Coruscant     3
#>  6 Kamino        3
#>  7 Corellia      2
#>  8 Kashyyyk      2
#>  9 Mirial        2
#> 10 Ryloth        2
#> # ... with 39 more rows
#> # A tibble: 49 x 2
#>    homeworld          n
#>    <chr>          <int>
#>  1 Aleen Minor        1
#>  2 Bespin             1
#>  3 Bestine IV         1
#>  4 Cato Neimoidia     1
#>  5 Cerea              1
#>  6 Champala           1
#>  7 Chandrila          1
#>  8 Concord Dawn       1
#>  9 Dathomir           1
#> 10 Dorin              1
#> # ... with 39 more rows
```

4-6. Make a plot of all individuals with mass on the x axis and height on the y axis. Put a best fit line on this plot. Who is the biggest outlier in this dataset?



```
#> # A tibble: 1 x 3
#>   name                   mass height
#>   <chr>                 <dbl>  <int>
#> 1 Jabba Desilijic Tiure  1358    175
```

7. Calculate BMI for all these individuals. What is the average BMI for all individuals?

   Via google: With the metric system, the formula for BMI is weight in kilograms divided by height in meters squared. Since height is commonly measured in centimeters, an alternate calculation formula, dividing the weight in kilograms by the height in centimeters squared, and then multiplying the result by 10,000, can be used

```
#> # A tibble: 59 x 4
#>    name                 BMI height  mass
#>    <chr>              <dbl>  <int> <dbl>
#>  1 Luke Skywalker      26.0    172    77
#>  2 C-3PO               26.9    167    75
#>  3 R2-D2               34.7     96    32
#>  4 Darth Vader         33.3    202   136
#>  5 Leia Organa         21.8    150    49
#>  6 Owen Lars           37.9    178   120
#>  7 Beru Whitesun lars  27.5    165    75
#>  8 R5-D4               34.0     97    32
#>  9 Biggs Darklighter   25.1    183    84
#> 10 Obi-Wan Kenobi      23.2    182    77
#> # ... with 49 more rows
#> # A tibble: 1 x 1
#>   `mean(BMI)`
#>         <dbl>
#> 1        32.0
```

8. What is the average BMI for each homeworld?

```
#> # A tibble: 40 x 2
#>    homeworld  avg.BMI
#>    <chr>        <dbl>
#>  1 Nal Hutta    443.
#>  2 Vulpter       50.9
#>  3 Kalee         34.1
#>  4 Bestine IV    34.0
#>  5 <NA>          32.6
#>  6 Malastare     31.9
#>  7 Trandosha     31.3
#>  8 Tatooine      29.3
#>  9 Sullust       26.6
#> 10 Dathomir      26.1
#> # ... with 30 more rows
```

9. Which homeworlds have the greatest percentage of individuals with BMI's greater than the average you found in #8 above? How many individuals have no missing data? Which variables have the most missing data?

```
#> # A tibble: 5 x 2
#>   homeworld  avg.BMI
#>   <chr>        <dbl>
#> 1 Nal Hutta    443.
#> 2 Vulpter       50.9
#> 3 Kalee         34.1
#> 4 Bestine IV    34.0
#> 5 <NA>          32.6
```

10. How many individuals have no missing data? Which variables have the most missing data?

    Via google: https://stackoverflow.com/questions/22353633/filter-for-complete-cases-in-data-frame-using-dplyr-case-wise-deletion

```
#> # A tibble: 29 x 14
#>    name         height  mass hair_color  skin_color eye_color
#>    <chr>         <int> <dbl> <chr>       <chr>      <chr>
#>  1 Luke Skywa~     172    77 blond       fair       blue
#>  2 Darth Vader     202   136 none        white      yellow
#>  3 Leia Organa     150    49 brown       light      brown
#>  4 Owen Lars       178   120 brown, grey light      blue
#>  5 Beru White~     165    75 brown       light      blue
#>  6 Biggs Dark~     183    84 black       light      brown
#>  7 Obi-Wan Ke~     182    77 auburn, wh~ fair       blue-gray
#>  8 Anakin Sky~     188    84 blond       fair       blue
#>  9 Chewbacca       228   112 brown       unknown    blue
#> 10 Han Solo        180    80 brown       fair       brown
#> # ... with 19 more rows, and 8 more variables:
#> #   birth_year <dbl>, sex <chr>, gender <chr>,
#> #   homeworld <chr>, species <chr>, films <list>,
#> #   vehicles <list>, starships <list>
#> Warning: `funs()` was deprecated in dplyr 0.8.0.
#> Please use a list of either functions or lambdas:
#>
#>   # Simple named list:
#>   list(mean = mean, median = median)
#>
#>   # Auto named with `tibble::lst()`:
#>   tibble::lst(mean, median)
#>
#>   # Using lambdas
#>   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
#> This warning is displayed once every 8 hours.
#> Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
#> # A tibble: 1 x 14
#>    name height  mass hair_color skin_color eye_color
#>   <int>  <int> <int>      <int>      <int>     <int>
#> 1     0      6    28          5          0         0
#> # ... with 8 more variables: birth_year <int>, sex <int>,
#> #   gender <int>, homeworld <int>, species <int>,
#> #   films <int>, vehicles <int>, starships <int>
```

Chapter 8

Lab 2: Pretty pictures!

> Please make sure you have read and understood R4DS Chapter on data visulization. Also check out the *Data Visualization with ggplot2 Cheat Sheet* from RStudio. and

> **Think DEEPLY:** Why is being able to generate good data vislization in R important *even* with awesome tools like PowerBI and Tableau around?

8.1 ggplot package and code

```
ggplot(data = ___, mapping = aes(x = ___)) +
  geom_histogram(binwidth = ___) +
  facet_wrap(~___)

Let's deconstruct this code:
- `ggplot()` is the function we are using to build our plot, in layers.
- In the first layer we always define the data frame as the first argument. Then, we define the m
- In the next layer we represent the data with **geom**etric shapes, in this case with a histogra
- In the final layer we facet the data by neighbourhood.
```

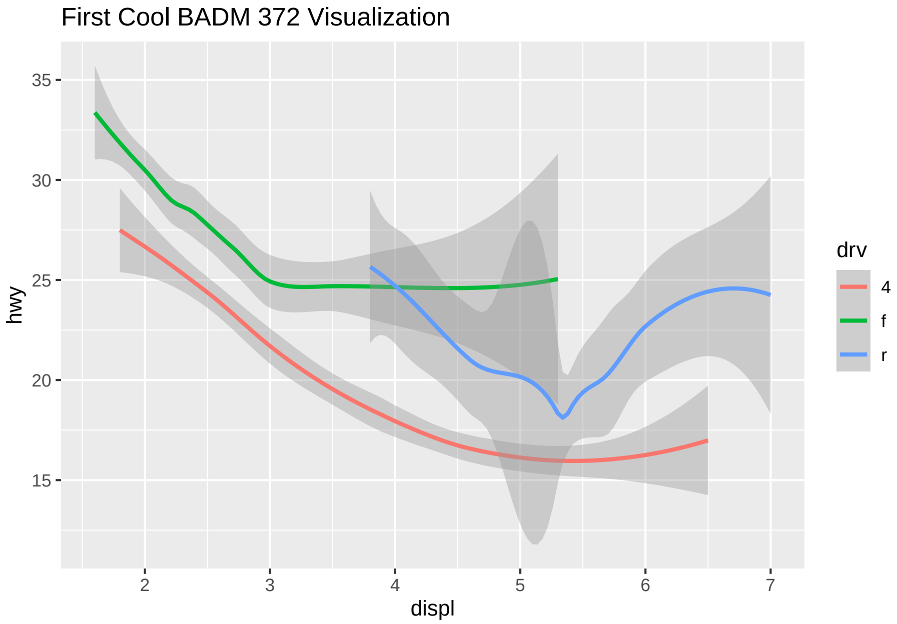
8.2 Packages

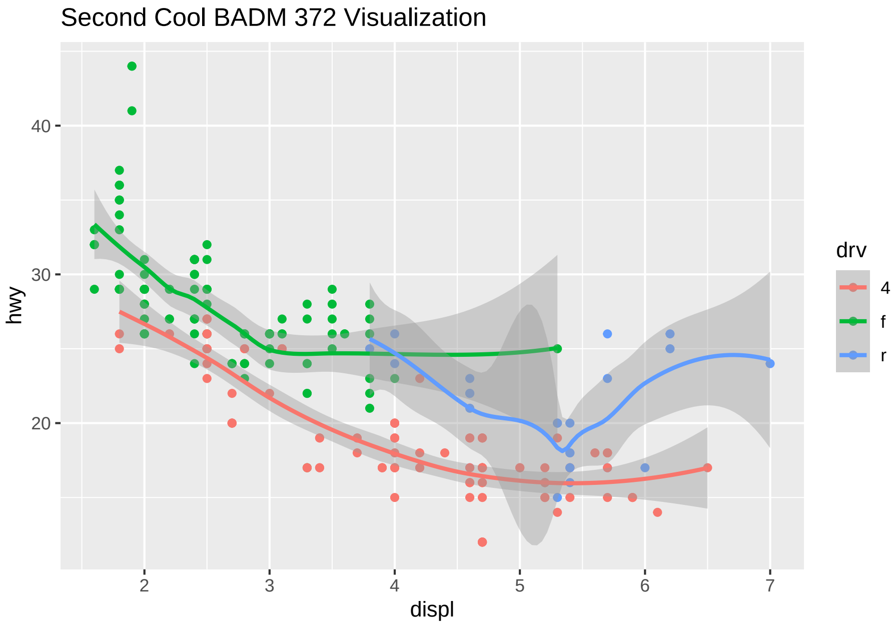
We'll use the **tidyverse** packages for this analysis, and the data is in the **dsbox** package. Run the following code in the Console to load these packages.

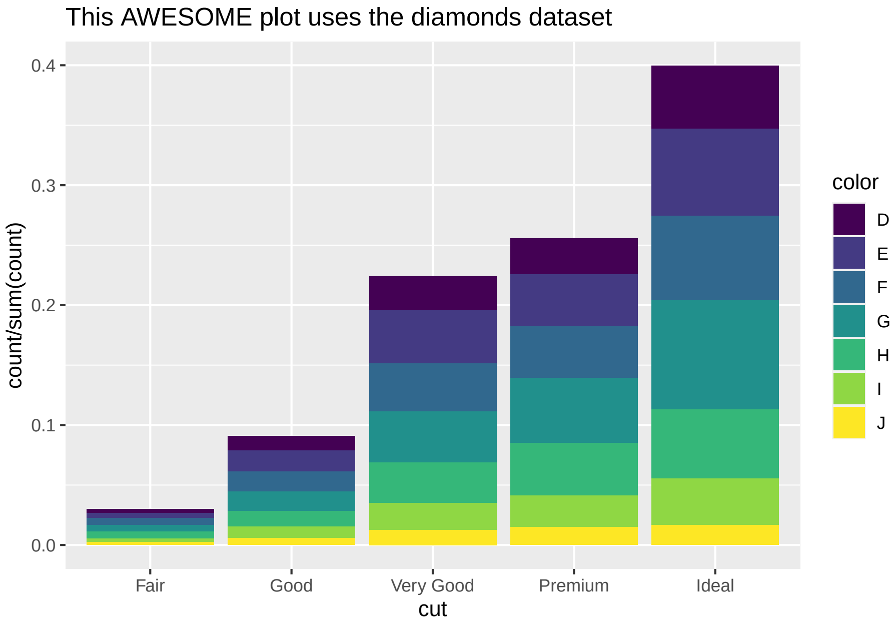
```
library(tidyverse)
library(dsbox)
```

8.3 Excercises

1. Create these figures using the data sets `mpg` or `diamonds` as needed:




Second Cool BADM 372 Visualization
hwy
40
30
20
displ
2
3
4
5
6
7
drv
4
f
r


This AWESOME plot uses the diamonds dataset
count/sum(count)
0.4
0.3
0.2
0.1
0.0
Fair
Good
Very Good
Premium
Ideal
cut
color
D
E
F
G
H
I
J

8.4 Airbnb listings in Edinburgh

This data comes from the `dsbox` package. Recent development in Edinburgh regarding the growth of Airbnb and its impact on the housing market means a better understanding of the Airbnb listings is needed. Using data provided by Airbnb, we can explore how Airbnb availability and prices vary by neighborhood.

The data come from the Kaggle database. It's been modified to better serve the goals of this exploration.

8.4.1 Learning goals

The goal of this assignment is not to conduct a thorough analysis of Airbnb listings in Edinburgh (yet?), but instead to give you a chance to practice your workflow, data visualization, and interpretation skills.

8.4.2 Data

2. The dataset you'll be using is called `edibnb` the data is in the **dsbox** package. Run `View(edibnb)` in your Console to view the data in the data viewer. What does each row in the dataset represent?

   Hint: The Markdown, ggplot2, and dplyr Quick Reference sheets has an example of inline R code that might be helpful. You can access it from the Help menu in RStudio.

3. How many observations (rows) does the dataset have? What interesting data is present? What was the purpose of this data being collected in the first place? Visit the kaggle site if needed.

Each column represents a variable. We can get a list of the variables in the data frame using the `names()` function. How else can we find out details of about these variables?

```
names(edibnb)
#>  [1] "id"                   "price"
#>  [3] "neighbourhood"        "accommodates"
#>  [5] "bathrooms"            "bedrooms"
#>  [7] "beds"                 "review_scores_rating"
#>  [9] "number_of_reviews"    "listing_url"
```

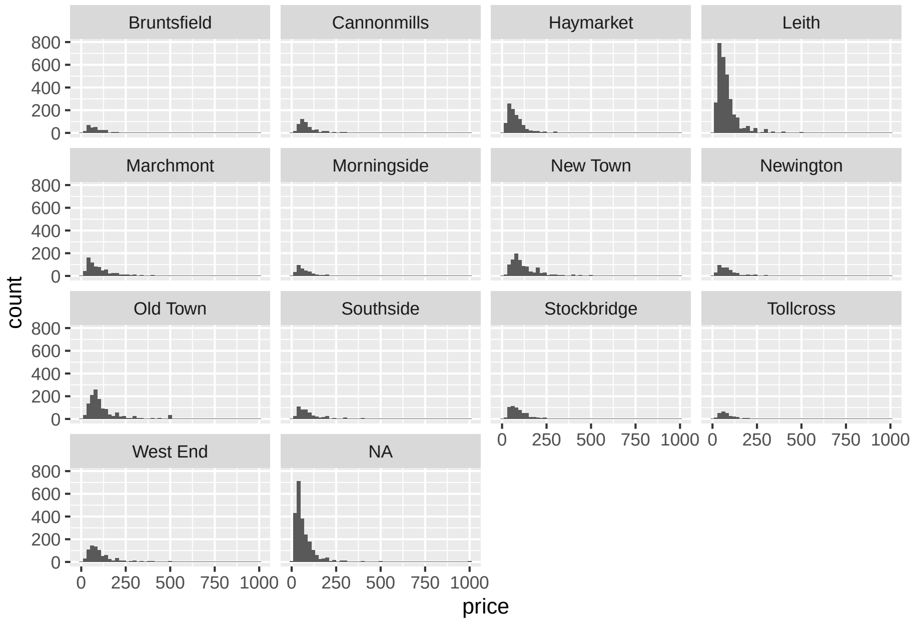
You can find descriptions of each of the variables in the help file for the dataset, which you can access by running `?edibnb` in your Console.

4. Create a faceted histogram where each facet represents a neighborhood and displays the distribution of Airbnb prices in that neighborhood. Your histogram may be similar (or better! than the example below.)

5. Create a faceted histogram where each facet represents a neighborhood and displays the distribution of Airbnb prices in that neighborhood. You histogram may be similar (or better! than the example below.)

   Note: The plot will give a warning about some observations with non-finite values for price being removed. Don't worry about the warning, it simply means that 199 listings in the data didn't have prices available, so they can't be plotted.

```
#> Warning: Removed 199 rows containing non-finite values
#> (stat_bin).
```



6. Create a similar visualization, this time showing the distribution of review scores (`review_scores_rating`) across neighborhoods. In your answer, include a brief interpretation of how Airbnb guests rate properties in general and how the neighborhoods compare to each other in terms of their ratings.

7. Create another informative visualization of your choosing. Be prepared to share it with the class – although the visualization should need no explaining!

8.5 Instructional staff employment trends

The next dataset is about instructional staff employee hiring trends between 1975 and 2011.

The dataset is called `instructors` found in `dsbox`. You can find descriptions of each of the variables in the help file for the dataset, which you can access by running `?instructors` in your Console.

The American Association of University Professors (AAUP) is a nonprofit membership association of faculty and other academic professionals. This report compiled by the AAUP shows trends in instructional staff employees between 1975 and 2011, and contains an image very similar to the one given below.

8.6 More Excercises

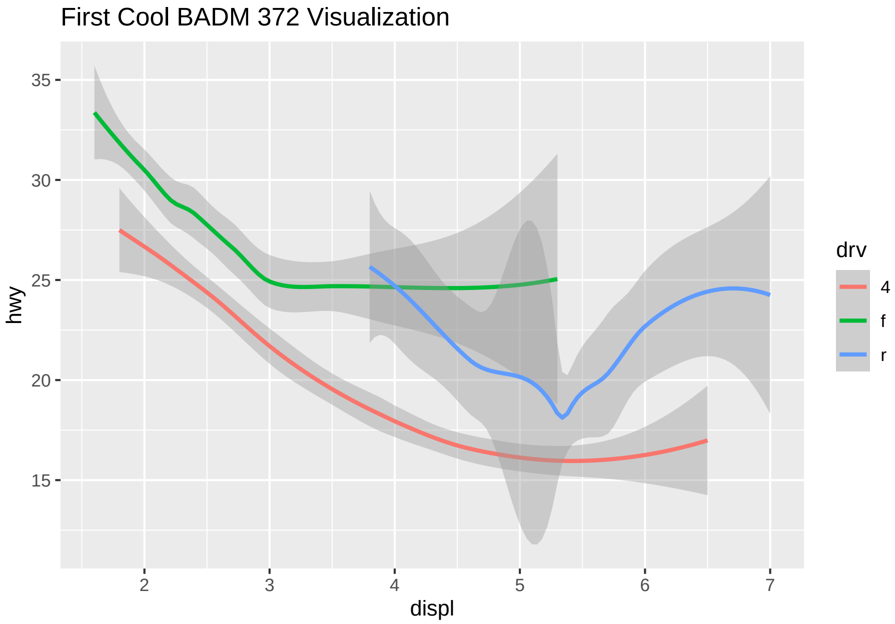
8. Recreate a graph similar to the one above.
9. Discuss how you would improve upon this visualization if the main objective was to communicate that the proportion of part-time faculty have gone up over time compared to other instructional staff types.Implement the improvements and provide your improved visualization as part of your answer. Also write a few sentences about why you chose to make these improvements and how they address the main goal stated above.

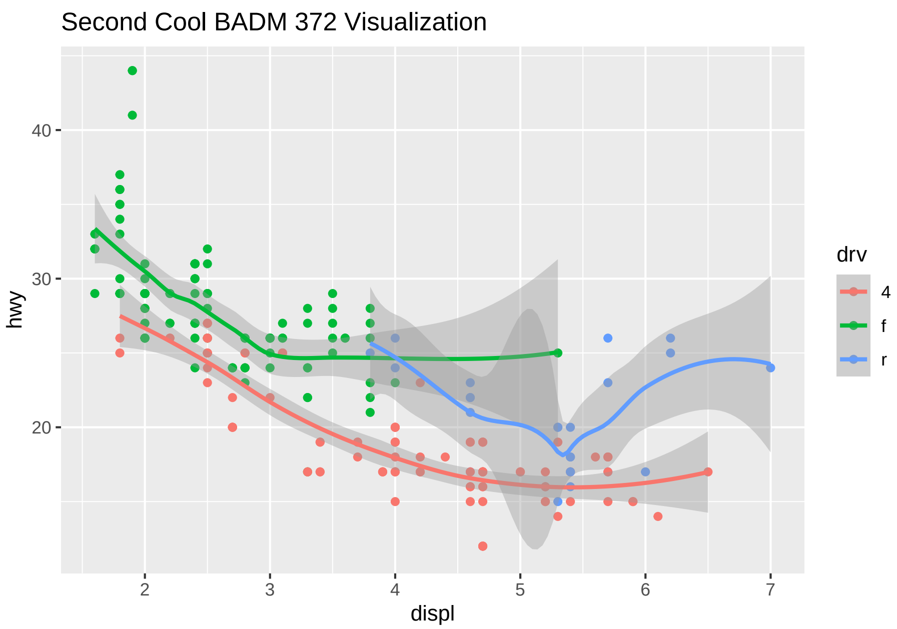
Chapter 9

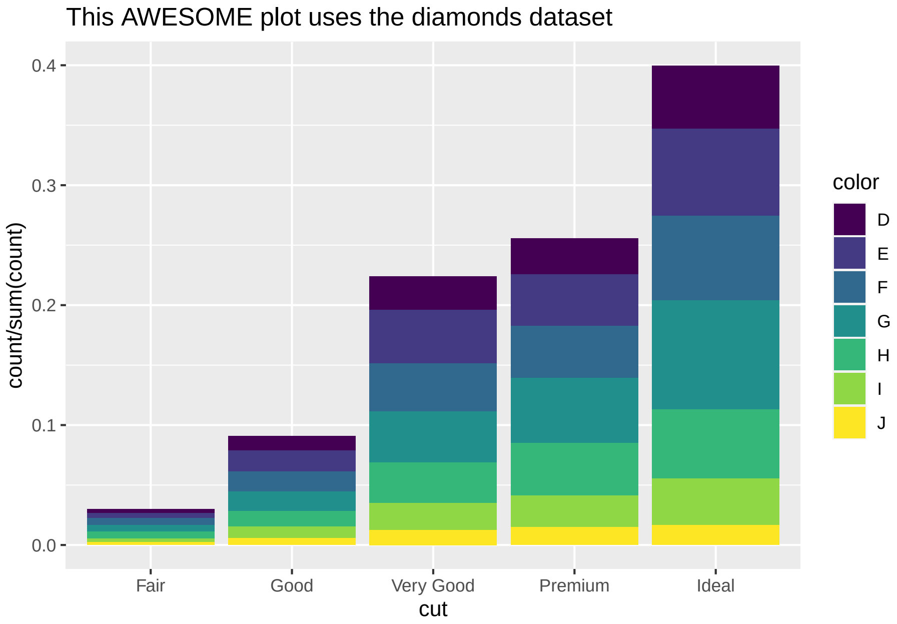
Lab 2 – ggplot without `dsbox`

9.1 Excercises using the data sets `mpg` or `diamonds`

1. Create these figures using the data sets `mpg` or `diamonds` as needed:




Second Cool BADM 372 Visualization
hwy
40
30
20
displ
2
3
4
5
6
7
drv
4
f
r

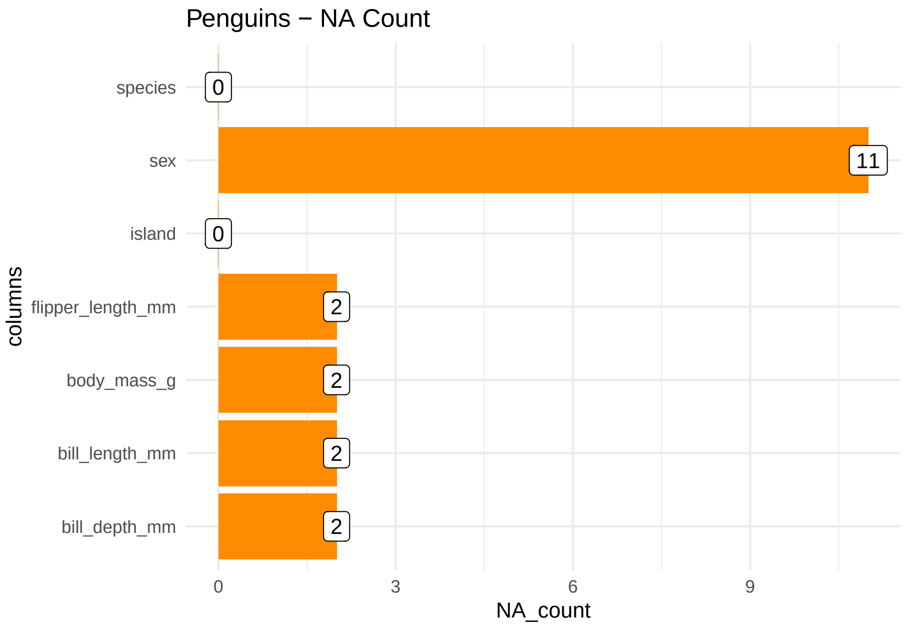

This AWESOME plot uses the diamonds dataset
count/sum(count)
0.4
0.3
0.2
0.1
0.0
Fair
Good
Very Good
Premium
Ideal
cut
color
D
E
F
G
H
I
J

9.2 palmerpenguins

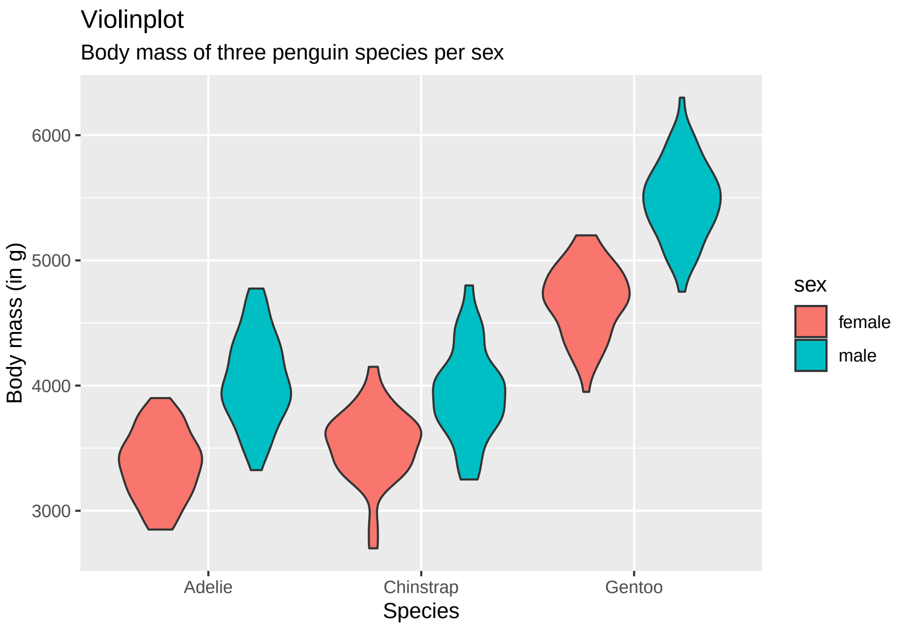
`palmerpenguins` is a realtively new package on CRAN, so you can install it from CRAN instead of Github.

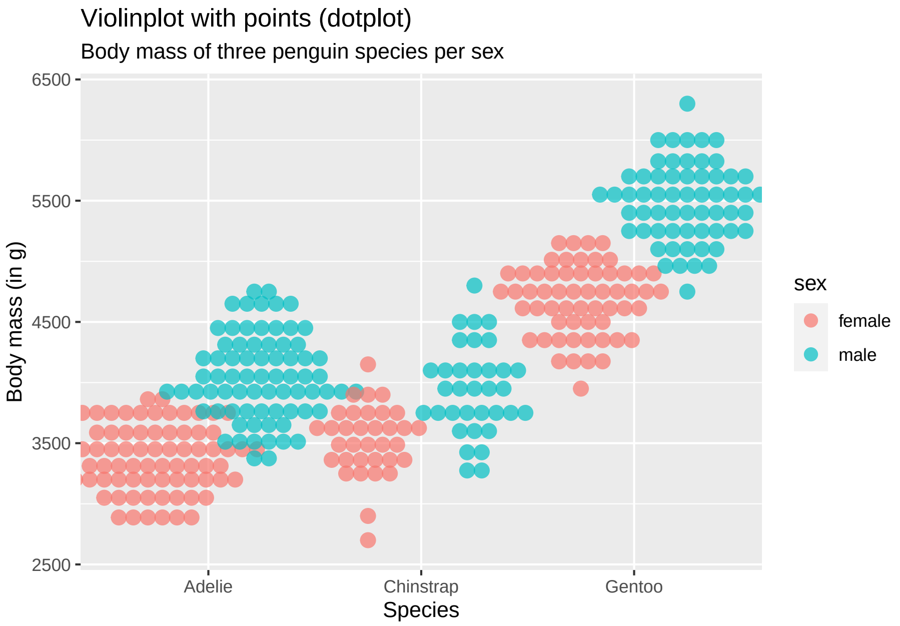
Install it like a normal package. After successful installation, you can find out that there are two datasets attached with the package – penguins and penguins_raw. You can check out their help page (?penguins_raw and ?penguins_raw) to understand more about respective datasets.

1. Please make a well-labeled, meangingful plot that show how many missing variables there are for each variable in the dataset. Your results shoud look something like this:



2. Make a plot showing a count of penguins of each species.
3. Create a plot that illustrates the relationship between flipper_length_mm and body_mass_g with respect to each species.
4. Create a plot that illustrates the relationship between flipper_length_mm and body_mass_g with respect to each species for each island.
5. Create a few plots of your own using new/interesting geoms and make sure the plots have meangiful, imprmfative labels, too. For possible examples:


Violinplot
Body mass of three penguin species per sex
Body mass (in g)
6000
5000
4000
3000
Adelie
Chinstrap
Gentoo
Species
sex
female
male


Violinplot with points (dotplot)
Body mass of three penguin species per sex
Body mass (in g)
6500
5500
4500
3500
2500
Adelie
Chinstrap
Gentoo
Species
sex
female
male

Chapter 10

Lab 3: coronavirus visualization, data wrangling, and dates

The package is available on GitHub here and is updated daily.

> I use the `coronavirus` package and use the `coronavirus::update_data()` function to keep the data current. This also has the dates preformatted which can be nice.

10.1 Let's look like Applied Analytics Superstars and make some neat visuals.

```
library(coronavirus)
library(dplyr)
library(ggplot2)
```

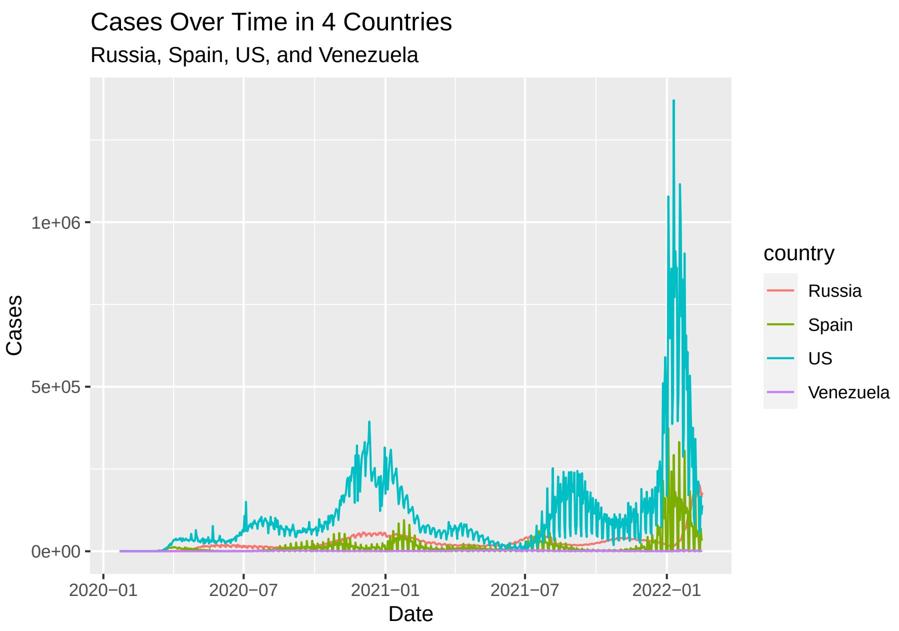
I'd recommend you always start by trying to understand a bit about the data.

```
head(coronavirus)
#>         date province country     lat      long      type
#> 1 2020-01-22  Alberta  Canada 53.9333 -116.5765 confirmed
#> 2 2020-01-23  Alberta  Canada 53.9333 -116.5765 confirmed
#> 3 2020-01-24  Alberta  Canada 53.9333 -116.5765 confirmed
#> 4 2020-01-25  Alberta  Canada 53.9333 -116.5765 confirmed
#> 5 2020-01-26  Alberta  Canada 53.9333 -116.5765 confirmed
#> 6 2020-01-27  Alberta  Canada 53.9333 -116.5765 confirmed
#>   cases   uid iso2 iso3 code3    combined_key population
#> 1     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 2     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 3     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 4     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 5     0 12401   CA  CAN   124 Alberta, Canada    4413146
#> 6     0 12401   CA  CAN   124 Alberta, Canada    4413146
#>   continent_name continent_code
#> 1  North America             NA
#> 2  North America             NA
#> 3  North America             NA
#> 4  North America             NA
#> 5  North America             NA
#> 6  North America             NA
```

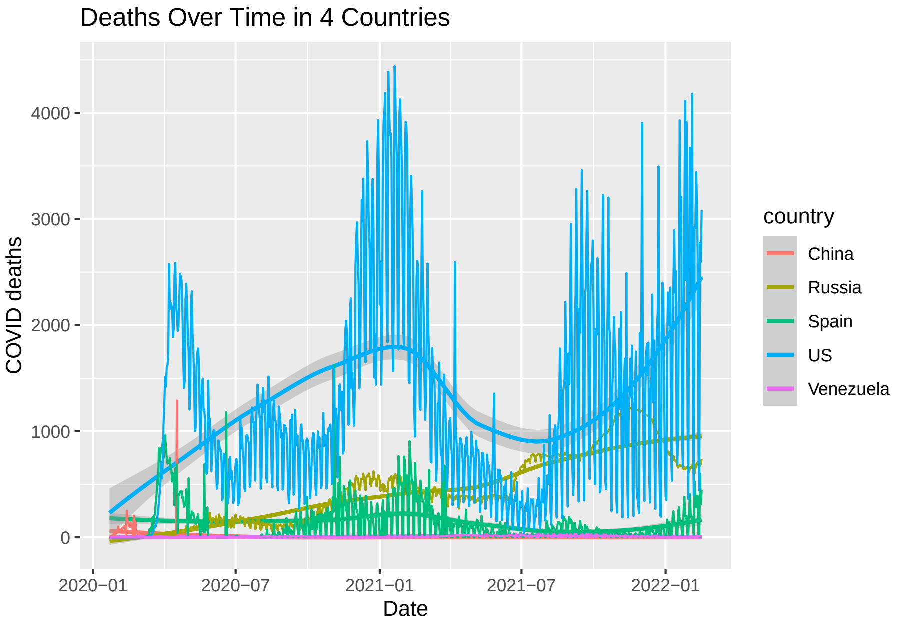
For example, what does this summary let us know?

```
summary(coronavirus$cases)
#>      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#> -30974748         0         0       669        29   1368563
```

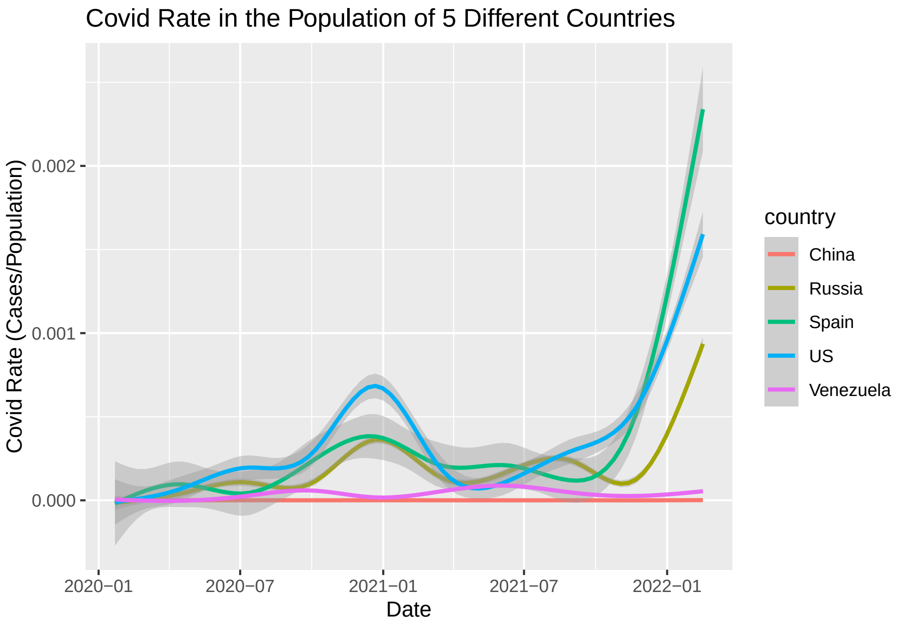
1. Can you create a visual showing the cases over time for Russia, Spain, US, and Venezuela? Also, why might `filter(cases >= 0)` be worth using?



2. Can you show deaths over time for Russia, Spain, US, and Venezuela? And can you play with your geoms and make something neat?

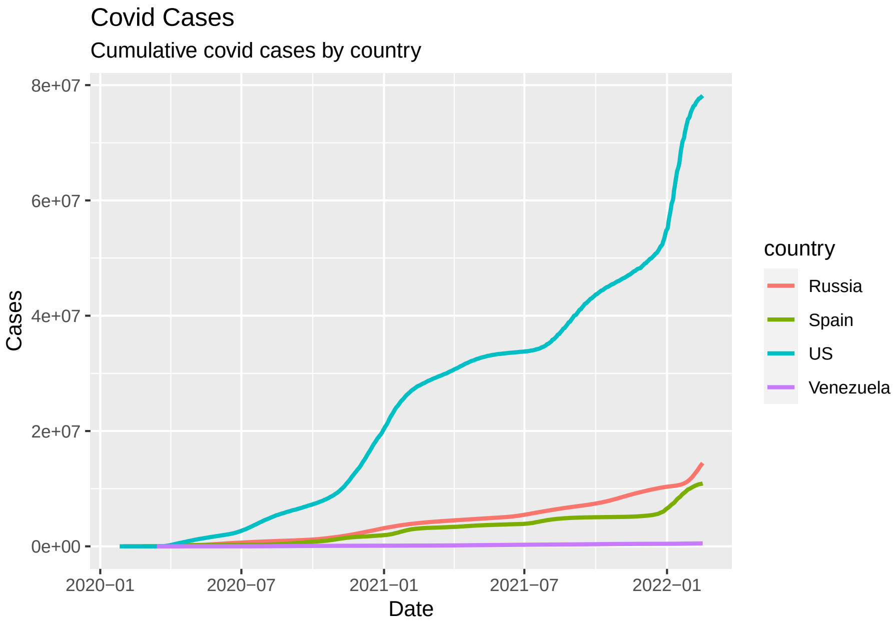


3. Now let's do a plot of COVID rate (# confirmed cases / population). Something like this.



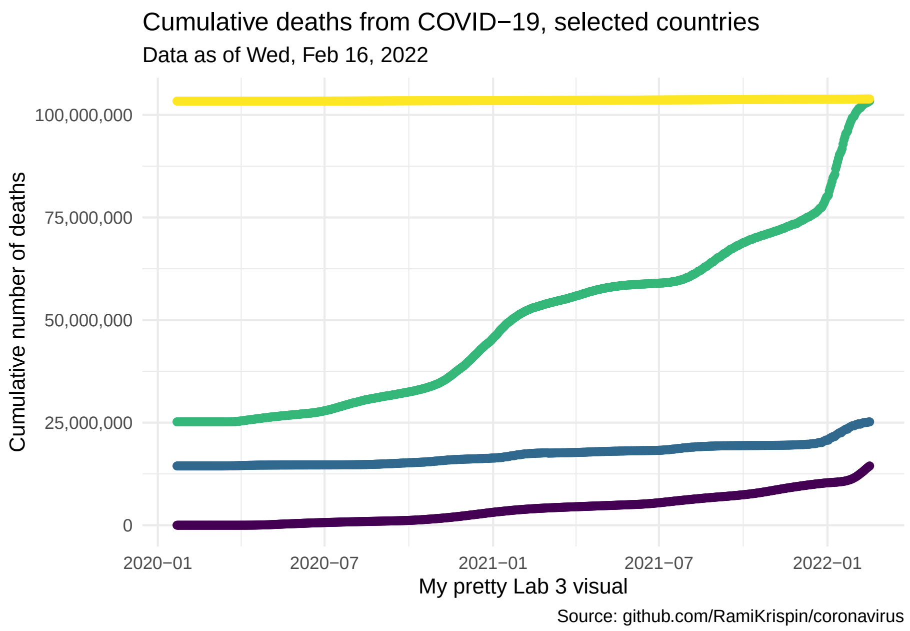
4. What is and **is not** useful about the previous illustration?

5. Make a chart with cumulative cases. Something like this:



6. With a little more time and a few extra packages, we **could** make a graph prettier. Try.

```
library(scales)
library(ggrepel)
library(glue)
library(lubridate)
```


Cumulative deaths from COVID−19, selected countries

Data as of Wed, Feb 16, 2022

Source: github.com/RamiKrispin/coronavirus

7. Now let's **really** have some fun. Let's illustrate death rates relative to confirmed cases. Why is this more challenging than anything we've done so far in this lab? We're going to have to make this data **tidy**.

One way to play this game.

Let's make a little table of just date, country, and deaths (with a meaningful variable name), and then count observations by coutry just to make sure eveything looks nice.

```
#>         date country deaths
#> 1 2020-01-22  Russia      0
#> 2 2020-01-23  Russia      0
#> 3 2020-01-24  Russia      0
#> 4 2020-01-25  Russia      0
#> 5 2020-01-26  Russia      0
#> 6 2020-01-27  Russia      0
#>     country   n
#> 1    Russia 757
#> 2     Spain 754
#> 3        US 757
#> 4 Venezuela 756
```

Let's make a little table of just confirmed cases.

```
#>         date country confirmed
#> 1 2020-01-22  Russia         0
#> 2 2020-01-23  Russia         0
#> 3 2020-01-24  Russia         0
#> 4 2020-01-25  Russia         0
#> 5 2020-01-26  Russia         0
#> 6 2020-01-27  Russia         0
#>     country   n
#> 1    Russia 757
#> 2     Spain 757
#> 3        US 757
#> 4 Venezuela 757
```

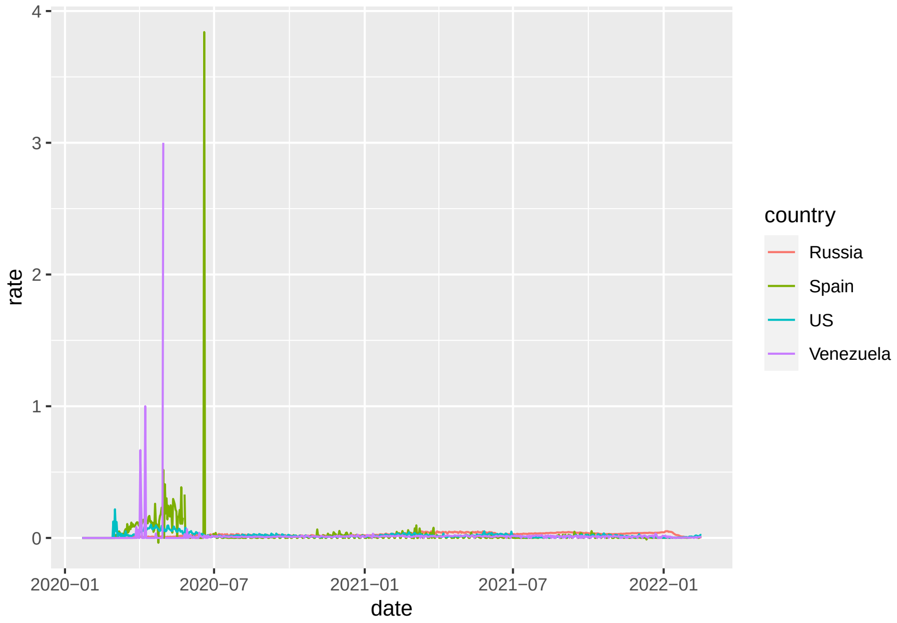
Let's join these together. I use `left_join`.

```
#>         date country deaths confirmed
#> 1 2020-01-22  Russia      0         0
#> 2 2020-01-23  Russia      0         0
#> 3 2020-01-24  Russia      0         0
#> 4 2020-01-25  Russia      0         0
#> 5 2020-01-26  Russia      0         0
#> 6 2020-01-27  Russia      0         0
#>     country   n
#> 1    Russia 757
#> 2     Spain 757
#> 3        US 757
#> 4 Venezuela 757
```

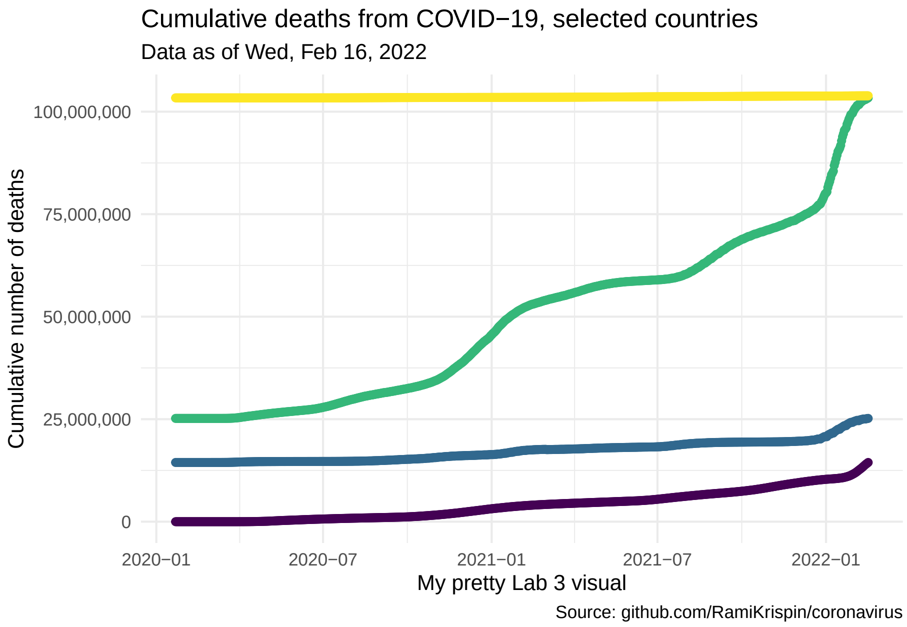
Let's add some cumulative statistics as well.

```
#>         date country deaths confirmed cumulative_cases
#> 1 2020-01-22  Russia      0         0                0
#> 2 2020-01-23  Russia      0         0                0
#> 3 2020-01-24  Russia      0         0                0
#> 4 2020-01-25  Russia      0         0                0
#> 5 2020-01-26  Russia      0         0                0
#> 6 2020-01-27  Russia      0         0                0
#>   cumulative_deaths rate
#> 1                 0    0
#> 2                 0    0
#> 3                 0    0
#> 4                 0    0
#> 5                 0    0
#> 6                 0    0
```

Now we can plot some more fun stuff.



```
#>       date              country              deaths       
#>  Min.   :2020-01-22   Length:3028        Min.   :   0.0  
#>  1st Qu.:2020-07-29   Class :character   1st Qu.:   5.0  
#>  Median :2021-02-03   Mode  :character   Median : 126.0  
#>  Mean   :2021-02-03                      Mean   : 452.7  
#>  3rd Qu.:2021-08-11                      3rd Qu.: 639.2  
#>  Max.   :2022-02-16                      Max.   :4442.0  
#>                                          NA's   :4       
#>    confirmed         cumulative_cases    cumulative_deaths
#>  Min.   : -74937.0   Min.   :        0   Min.   :     0   
#>  1st Qu.:    461.5   1st Qu.: 14445698   1st Qu.: 16977   
#>  Median :   7814.0   Median : 25190092   Median : 99431   
#>  Mean   :  34303.5   Mean   : 43523860   Mean   :135068   
#>  3rd Qu.:  28170.0   3rd Qu.:103362932   3rd Qu.:248203   
#>  Max.   :1368563.0   Max.   :103870974   Max.   :364273   
#>                                          NA's   :2147     
#>       rate          
#>  Min.   :-0.036576  
#>  1st Qu.: 0.004568  
#>  Median : 0.012750  
#>  Mean   : 0.021680  
#>  3rd Qu.: 0.023227  
#>  Max.   : 3.840391  
#>  NA's   :4
```


Cumulative deaths from COVID−19, selected countries
Data as of Wed, Feb 16, 2022
Cumulative number of deaths
100,000,000
75,000,000
50,000,000
25,000,000
0
2020−01
2020−07
2021−01
2021−07
2022−01
My pretty Lab 3 visual
Source: github.com/RamiKrispin/coronavirus

Chapter 11

Project (E1)

11.1 A project to call your own

Pick a dataset, any dataset…

…and do something with it. That is your first Analytics 2 project. Make us both proud, in a nutshell. More details below.

11.2 May be too long, but please do read

This project for this class will consist of analysis on a dataset of your own choosing. **Please make sure I am ok with your choice.** The dataset may already exist, or you may collect your own data using a survey or by conducting an experiment. You can choose the data based on your interests or based on work in other courses or research projects. The goal of this project is for you to demonstrate proficiency in the techniques we have covered in this class (and beyond, if you like) and apply them to a novel dataset in a meaningful way.

11.3 Data

In order for you to have the greatest chance of success with this project it is important that you choose a manageable dataset. This means that the data should be readily accessible and large enough that multiple relationships can be explored. As such, your dataset must have at least 50 observations and between 10 to 20 variables (exceptions can be made but you must speak with me first). The dataset's variables should include categorical variables, discrete numerical variables, and continuous numerical variables.

All analyses must be done in RStudio. If you are using a dataset that comes in a format that we haven't encountered in class, make sure that you are able to load it into RStudio as this can be tricky depending on the source. If you are having trouble ask for help before it is too late.

Reusing datasets from class: Do not reuse datasets used in examples / homework in the class.

11.4 Components

11.4.1 Project proposal

This is a draft of the introduction section of your project as well as a data analysis plan and your dataset. Each section should be no more than 1 page (excluding figures). You can check a print preview to confirm length.

> Your write up and all analysis including visuals must be done using R Markdown.

11.4.1.1 Section 1 - Introduction:

The introduction should introduce your general research question and your data (where it came from, how it was collected, what are the cases, what are the variables, etc.).

11.4.1.2 Section 2 - Data analysis plan:

The data analysis plan should include:

- The outcome (dependent, response, Y) and predictor (independent, explanatory, X) variables you will use to answer your question.
- The comparison groups you will use, if applicable.
- Very preliminary exploratory data analysis, including some summary statistics and visualizations, along with some explanation on how they help you learn more about your data. (You can add to these later as you work on your project..)
- The statistical method(s) that you believe will be useful in answering your question(s). (You can update these later as you work on your project.)
- Ideally you will use at least two out of these options: tree methods, linear regression, and classification (like logistic regression).
- What results from these specific statistical methods are needed to support your hypothesized answer?

11.4.1.3 Section 3 - Data:

In yuor write up, include enough details that I understand what your raw data looked like and included.

11.4.2 Project

11.4.2.1 Write up

After providing the description of your dataset and research question in the introduction use the remainder of your write up to showcase how you have arrived at an answer / answers to your question using any techniques we have learned in this class (and some beyond, if you're feeling adventurous). The goal is not to do an exhaustive data analysis i.e., do not calculate every statistic and procedure you have learned for every variable, but rather let me know that you are proficient at asking meaningful questions and answering them with results of data analysis, that you are proficient in using R, and that you are proficient at interpreting and presenting the results. Focus on methods that help you begin to answer your research questions. You do not have to apply every statistical procedure we learned. Also pay attention to your presentation. Neatness, coherency, and clarity will count.

Your write up must also include a one to two page conclusion and discussion. This will require a summary of what you have learned about your research question along with statistical arguments supporting your conclusions. Also critique your own methods and provide suggestions for improving your analysis. Issues pertaining to the reliability and validity of your data, and appropriateness of the statistical analysis should be discussed here. A paragraph on what you would do differently if you were able to start over with the project or what you would do next if you were going to continue work on the project should also be included.

> The project is very open ended. You should create some kind of compelling visualization(s) of this data in R.

There is no limit on what tools or packages you may use, but sticking to packages we learned in class (ISLR and R4DS) is required. You do not need to visualize all of the data at once. A single high quality visualization will receive a much higher grade than a large number of poor quality visualizations.

Before you finalize your write up, make sure your chunks are turned off with `echo = FALSE`. **Exception:** If you want to highlight something specific about a piece of code, you're welcomed to show that portion. [See below: I will also want a copy of the raw .Rmd file not just the html output.]

You can add sections as you see fit to the project but make sure you have a section called Introduction at the beginning and a section called Conclusion at the end. The rest is up to you!

11.4.2.2 Presentation

10 minutes maximum.

You can use any software you like for your final presentation, including R Markdown to create your slides. There isn't a limit to how many slides you can use, just a time limit (10 minutes total). Perhaps try `ioslides` or `beamer`. Your presentation should not just be an account of everything you tried ("then we did this, then we did this, etc."), instead it should convey what choices you made, and why, and what you found.

11.4.2.3 Delivarables

Your submission should include

- RMarkdown file (formatted to clearly present all of your code and results)
- HTML file
- Dataset(s) (in csv or RData format, in a `/data` folder)
- Presentation (if using Keynote/PowerPoint/Google Slides, export to PDF and put in repo, in a `/presentation` folder)

Style and format does count for this assignment, so please take the time to make sure everything looks good and your data and code are properly formated.

11.5 Grading

Total	100 pts
Introduction	20 pts
Data analysis plan	20 pts
Data Methods and code quality	50 pts
Organization	10 pts

Grading of the project will take into account the following:

- Content - What is the quality of research and/or policy question and relevancy of data to those questions?

- Correctness - Are statistical procedures carried out and explained correctly?
- Writing and Presentation - What is the quality of the statistical presentation, writing, and explanations?
- Creativity and Critical Thought - Is the project carefully thought out? Are the limitations carefully considered? Does it appear that time and effort went into the planning and implementation of the project?

A general breakdown of scoring is as follows:

- 90%-100% - Outstanding effort. Student understands how to apply all statistical concepts, can put the results into a cogent argument, can identify weaknesses in the argument, and can clearly communicate the results to others.
- 80%-89% - Good effort. Student understands most of the concepts, puts together an adequate argument, identifies some weaknesses of their argument, and communicates most results clearly to others.
- 70%-79% - Passing effort. Student has misunderstanding of concepts in several areas, has some trouble putting results together in a cogent argument, and communication of results is sometimes unclear.
- 60%-69% - Struggling effort. Student is making some effort, but has misunderstanding of many concepts and is unable to put together a cogent argument. Communication of results is unclear.
- Below 60% - Student is not making a sufficient effort.

Late penalty:

- Late, but within 24 hours of due date/time: -20% (only applies to written portion, there is no option to do your presentation later)
- Any later: no credit

Chapter 12

Functions

12.1 Writing Functions

12.1.1 Fahrenheit to Kelvin

$k = ((f - 32) * (5/9)) + 273.15$

```
((32 - 32) * (5 / 9)) + 273.15
#> [1] 273.15
((212 - 32) * (5 / 9)) + 273.15
#> [1] 373.15
((-42 - 32) * (5 / 9)) + 273.15
#> [1] 232.0389
```

```
f_k <- function(f_temp) {
    ((f_temp - 32) * (5 / 9)) + 273.15
}
```

```
f_k(32)
#> [1] 273.15
f_k(212)
#> [1] 373.15
f_k(-42)
#> [1] 232.0389
```

12.1.2 Kelvin to Celsius

```
k_c <- function(temp_k) {
    temp_c <- temp_k - 273.15
    return(temp_c)
}
```

```
k_c(0)
#> [1] -273.15
```

12.1.3 Fahrenheit to Celsius

```
f_c <- function(temp_f) {
    temp_k <- f_k(temp_f)
    temp_c <- k_c(temp_k)
    return(temp_c)
}
```

```
f_c(32)
#> [1] 0
f_c(212)
#> [1] 100
```

12.2 Testing Functions

```
library(testthat)
testthat::expect_equal(f_c(32), 0)
testthat::expect_equal(f_c(212), 100)
```

12.3 Exercise

1. What happens if you use `NA`, `Inf`, `-Inf` in your function?
2. What are some better names to give the functions we wrote?

- How would you name these functions in a package?

12.4 Checking values

Calculating weighted means

```
mean_wt <- function(x, w) {
  sum(x * w) / sum(w)
}
```

```
mean_wt(1:6, 1:6)
#> [1] 4.333333
```

If you expect the lengths to be the same, then you should test for it in the function

```
mean_wt(1:6, 1:3)
#> [1] 7.666667
```

```
mean_wt <- function(x, w) {
  if (length(x) != length(w)) {
    stop("`x` and `w` should be the same length")
  }
  sum(x * w) / sum(w)
}
```

```
mean_wt(1:6, 1:3)
#> Error in mean_wt(1:6, 1:3): `x` and `w` should be the same length
```

12.5 dot-dot-dot ...

Use it to pass on arguments to another function inside.

But you can also use it to force named arguments in your function.

```
sum_3 <- function(x, y, z) {
  return(x + y + z)
}
```

```
sum_3(1, 2, 3)
#> [1] 6
```

```
sum_3 <- function(x, y, ..., z) {
  return(x + y + z)
}
```

```
sum_3(1, 2, z = 3)
#> [1] 6
```

```
sum_3(1, 2, z = 3)
#> [1] 6
```

Chapter 13

Conditionals

13.1 if statements

```
# make a modification to this function
k_c <- function(temp_k) {
    if (temp_k < 0) {
        warning('you passed in a negative Kelvin number')
        # stop()
        return(NA)
    }
    temp_c <- temp_k - 273.15
    return(temp_c)
}
```

```
k_c(-9)
#> Warning in k_c(-9): you passed in a negative Kelvin number
#> [1] NA
```

Our current function does not deal with missing numbers

```
k_c(NA)
```

```
Error in if (temp_k < 0) { : missing value where TRUE/FALSE needed
```

```
k_c(0)
#> [1] -273.15
```

13.2 If else statements

```
k_c <- function(temp_k) {
    if (temp_k < 0) {
        warning('you passed in a negative Kelvin number')
        # stop()
        return(NA)
    } else {
        temp_c <- temp_k - 273.15
        return(temp_c)
    }
}
```

```
k_c(-9)
#> Warning in k_c(-9): you passed in a negative Kelvin number
#> [1] NA
```

Our current function does not deal with missing numbers

```
k_c(NA)
```

```
k_c(0)
#> [1] -273.15
```

13.3 Dealing with NA

Re-write our function to work with missing values.

Note you need to make the `NA` check first.

```
k_c <- function(temp_k) {
    if (is.na(temp_k)) {
        return(NA)
    } else if (temp_k < 0) {
        warning('you passed in a negative Kelvin number')
        # stop()
        return(NA)
    } else {
        temp_c <- temp_k - 273.15
        return(temp_c)
    }
}
```

```
k_c(-9)
#> Warning in k_c(-9): you passed in a negative Kelvin number
#> [1] NA
```

```
k_c(NA)
#> [1] NA
```

```
k_c(0)
#> [1] -273.15
```

```
if (c(TRUE, FALSE)) {}
#> Warning in if (c(TRUE, FALSE)) {: the condition has length >
#> 1 and only the first element will be used
#> NULL
```

```
if (NA) {}
#> Error in if (NA) {: missing value where TRUE/FALSE needed
```

use `&&` and `||` to short-circuit the boolean comparisons. This will also guarantee a value of length `1L`. `==` is also vectorized, should use `identical()` or `all.equal()`.

`identical` is very strict. Doesn't corece types.

```
identical(0L, 0)
#> [1] FALSE
```

`all.equal` has ability to set tolerances.

`all.equal`: compare R objects x and y testing 'near equality'. If they are different, comparison is still made to some extent, and a report of the differences is returned. Do not use all.equal directly in if expressions—either use isTRUE(all.equal(....)) or identical if appropriate.

```
all.equal(0L, 0)
#> [1] TRUE
```

```
if (isTRUE(all.equal(0L, 0))) {print("Hello")}
#> [1] "Hello"
```

13.4 Fizzbuzz

```r
fizzbuzz <- function(x) {
  # these two lines check that x is a valid input
  stopifnot(length(x) == 1)
  stopifnot(is.numeric(x))
  if (!(x %% 3) && !(x %% 5)) {
    "fizzbuzz"
  } else if (!(x %% 3)) {
    "fizz"
  } else if (!(x %% 5)) {
    "buzz"
  } else {
    # ensure that the function returns a character vector
    as.character(x)
  }
}
```

```r
fizzbuzz(6)
#> [1] "fizz"
```

Check modulo 3 only once

```r
fizzbuzz2 <- function(x) {
  # these two lines check that x is a valid input
  stopifnot(length(x) == 1)
  stopifnot(is.numeric(x))
  if (!(x %% 3)) {
    if (!(x %% 5)) {
      "fizzbuzz"
    } else {
      "fizz"
    }
  } else if (!(x %% 5)) {
    "buzz"
  } else {
    # ensure that the function returns a character vector
    as.character(x)
  }
}
```

```r
fizzbuzz(6)
#> [1] "fizz"
```

13.4.1 Vectorized conditionals

```
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following object is masked from 'package:testthat':
#>
#>     matches
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
fizzbuzz_vec <- function(x) {
  dplyr::case_when(
    !(x %% 3) & !(x %% 5) ~ "fizzbuzz",
    !(x %% 3) ~ "fizz",
    !(x %% 5) ~ "buzz",
    TRUE ~ as.character(x)
  )
}
```

```
fizzbuzz(1:10)
#> Error in fizzbuzz(1:10): length(x) == 1 is not TRUE
```

```
fizzbuzz_vec(1:10)
#>  [1] "1"    "2"    "fizz" "4"    "buzz" "fizz" "7"    "8"
#>  [9] "fizz" "buzz"
```

13.4.2 Multiple conditions

```
if (this) {
  # do that
} else if (that) {
  # do something else
} else {
  #
}
```

13.4.2.1 switch

```
calc_op <- function(x, y, op) {
  switch(op,
         plus = x + y,
         minus = x - y,
         times = x * y,
         divide = x / y,
         stop("Unknown op!")
  )
}
```

```
calc_op(10, 20, "times")
#> [1] 200
```

```
calc_op(10, 20, "divide")
#> [1] 0.5
```

13.4.2.2 cut

```
describe_temp <- function(temp) {
  if (temp <= 0) {
    "freezing"
  } else if (temp <= 10) {
    "cold"
  } else if (temp <= 20) {
    "cool"
  } else if (temp <= 30) {
    "warm"
  } else {
    "hot"
  }
}
```

```
describe_temp(16)
#> [1] "cool"
```

Current function can't handle vectors

```
describe_temp(c(16, 61))
#> Warning in if (temp <= 0) {: the condition has length > 1
#> and only the first element will be used
#> Warning in if (temp <= 10) {: the condition has length > 1
#> and only the first element will be used
#> Warning in if (temp <= 20) {: the condition has length > 1
#> and only the first element will be used
#> [1] "cool"
```

How cut works:

```
values <- -10:10
values
#>  [1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3
#> [15]   4   5   6   7   8   9  10
```

```
cut(values, c(-Inf, -5, -1, 1, 7, Inf))
#>  [1] (-Inf,-5] (-Inf,-5] (-Inf,-5] (-Inf,-5] (-Inf,-5]
#>  [6] (-Inf,-5] (-5,-1]   (-5,-1]   (-5,-1]   (-5,-1]
#> [11] (-1,1]    (-1,1]    (1,7]     (1,7]     (1,7]
#> [16] (1,7]     (1,7]     (1,7]     (7, Inf]  (7, Inf]
#> [21] (7, Inf]
#> Levels: (-Inf,-5] (-5,-1] (-1,1] (1,7] (7, Inf]
```

```
cut(values, c(-Inf, -5, -1, 1, 7, Inf), labels = LETTERS[1:5], right = TRUE)
#>  [1] A A A A A A B B B B C C D D D D D D E E E
#> Levels: A B C D E
```

```
cut(values, c(-Inf, -5, -1, 1, 7, Inf), labels = LETTERS[1:5], right = FALSE)
#>  [1] A A A A A B B B B C C D D D D D D E E E E
#> Levels: A B C D E
```

13.5 Exercise

1. Rewrite the function using `cut`

```
describe_temp <- function(temp) {
  if (temp <= 0) {
    "freezing"
  } else if (temp <= 10) {
    "cold"
  } else if (temp <= 20) {
    "cool"
  } else if (temp <= 30) {
    "warm"
  } else {
    "hot"
  }
}
```

2. How do you indicate `<` and `<=`?

Chapter 14

Linear Regression

Your resource for this is ISLR chapter 3: linear regression.

14.1 Exercises

1. Make sure you can define the terms below **outloud**, in your own words, so that they make sense both to you and to someone else (me?). Actually practice saying and defining these terms **outloud** until your answers make sense:

- least squares approach
- confidence interval
- p-value
- R^2
- Adjusted R^2
- qualitative predictor
- collinearity
- KNN
- Residual standard error
- F-statistic
- Explain the point of Figure 3.1

2. In `m1`, below, which variables are significant predictors of Balance? How do you know?

```
library("ISLR")
data(Credit)
attach(Credit)
```

```
head(Credit)
#>   ID  Income Limit Rating Cards Age Education Gender
#> 1  1  14.891  3606    283     2  34        11   Male
#> 2  2 106.025  6645    483     3  82        15 Female
#> 3  3 104.593  7075    514     4  71        11   Male
#> 4  4 148.924  9504    681     3  36        11 Female
#> 5  5  55.882  4897    357     2  68        16   Male
#> 6  6  80.180  8047    569     4  77        10   Male
#>   Student Married Ethnicity Balance
#> 1      No     Yes Caucasian     333
#> 2     Yes     Yes     Asian     903
#> 3      No      No     Asian     580
#> 4      No      No     Asian     964
#> 5      No     Yes Caucasian     331
#> 6      No      No Caucasian    1151
m1 <- lm(Balance ~ Age + Income + Education, data = Credit)
summary(m1)
#> 
#> Call:
#> lm(formula = Balance ~ Age + Income + Education, data = Credit)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -867.14 -343.14  -49.44  316.55 1080.56 
#> 
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)    
#> (Intercept) 348.8115   112.6895   3.095  0.00211 ** 
#> Age          -2.1863     1.2004  -1.821  0.06930 .  
#> Income        6.2380     0.5877  10.614  < 2e-16 ***
#> Education     0.8058     6.5254   0.123  0.90179    
#> ---
#> Signif. codes:  
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 407.2 on 396 degrees of freedom
#> Multiple R-squared:  0.2215, Adjusted R-squared:  0.2156 
#> F-statistic: 37.56 on 3 and 396 DF,  p-value: < 2.2e-16
```

3. How “good” is the model created in `m1`? How do you know?
4. Add more `Credit` variables to model `m1`. Can you find two other variables that have extremely high collinearity? What are they? How do you know that they have high collinearity? Why does this make sense given what each of the variables mean?
5. Based on the model below, what would you predict the balance to be for

an individual who is 40, has an income of $100,000, 16 years of education, is Asian and not a student?

```
m2 <- lm(Balance ~ Age + Income + Education + Ethnicity + Student, data = Credit)
summary(m2)
#> 
#> Call:
#> lm(formula = Balance ~ Age + Income + Education + Ethnicity + 
#>     Student, data = Credit)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -818.77 -322.14  -54.52  315.67  781.45 
#> 
#> Coefficients:
#>                    Estimate Std. Error t value Pr(>|t|)    
#> (Intercept)        336.6241   115.6311   2.911  0.00381 ** 
#> Age                 -1.9756     1.1595  -1.704  0.08922 .  
#> Income               6.1491     0.5666  10.853  < 2e-16 ***
#> Education           -1.7606     6.3060  -0.279  0.78024    
#> EthnicityAsian     -14.2547    55.5240  -0.257  0.79752    
#> EthnicityCaucasian   8.8839    48.3276   0.184  0.85424    
#> StudentYes         382.0498    65.6854   5.816 1.25e-08 ***
#> ---
#> Signif. codes:  
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 392.2 on 393 degrees of freedom
#> Multiple R-squared:  0.2833, Adjusted R-squared:  0.2723 
#> F-statistic: 25.89 on 6 and 393 DF,  p-value: < 2.2e-16
```

6. Interpret this model and its output, especially the coefficients `Income:Education   0.3149`:

```
m3 <- lm(Balance ~ Income*Education, data = Credit)
summary(m3)
#> 
#> Call:
#> lm(formula = Balance ~ Income * Education, data = Credit)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -858.07 -349.99  -56.12  304.51 1083.93 
#> 
#> Coefficients:
```

```
#>                  Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      435.4599   147.1000   2.960  0.00326 **
#> Income             1.8168     2.4727   0.735  0.46294
#> Education        -13.9887    10.5931  -1.321  0.18741
#> Income:Education   0.3149     0.1788   1.761  0.07902 .
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 407.3 on 396 degrees of freedom
#> Multiple R-squared:  0.2211, Adjusted R-squared:  0.2152
#> F-statistic: 37.47 on 3 and 396 DF,  p-value: < 2.2e-16
```

Chapter 15

Tree-Based Methods

Your resource for this is ISLR Chapter 8: Tree-Based Methods.

Make sure you can exlain the terms/ideas/figures below **outloud**, in your own words, so that they make sense both to you and to someone else (me?). Actually practice exlaining the terms/ideas/figures **outloud** until your answers make sense:

- Regression vs. classification trees
- Understand Figure 8.1, Figure 8.2, Algorithm 8.1, Figure 8.3, Figure 8.4, Figure 8.5, Figure 8.6, Figure 8.7
- Bea ble to explain the=’s and -’s of trees (see section 8.1.4)
- Understand “combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss in interpretation.”
- top-down, greedy approach (aka recursive binary splitting)
- tree pruning and subtrees
- cost complexity pruning (aka weakest link pruning)
- bagging
- random forests
- boosting
- Bayesian additive regression trees