Practice2

Sebastian Montesinos

Due by midnight, Friday, Feb. 25

Practice2

Reminder: Practice assignments may be completed working with other individuals.

Reading

The associated reading for the week is Chapter 4, Chapter 5, Chapter 6 (skip 6.4), and Sections 8.3 and 8.4.

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (anything besides our textbook, course materials in the repo, labs, R help menu) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

-

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

-

1 - Hardest Concept

We've covered many different data wrangling concepts and associated verbs during this unit. This problem will help you identify ways to get support on concepts you find challenging, beyond what we have in class and in the textbook.

> part a - What concept or data wrangling verb did you find most challenging to work with during this unit?

Solution: I still find the combination of group_by() with summarise() to be hard to conceptualize.

> part b - Look in our Resources folder at the tidyr and data-transformation cheat sheets. Can you find information related to your selected concept or verb? If so, what sheet is it in? What if any insights do you get from the cheatsheet?

(If you picked a concept or verb not on these cheatsheets, try to find it on a different one, or ask me where it is likely to be. These are just the two most common cheatsheets to reference for these chapters.)

Solution: In the data-transformation pdf there is a description of both the summarise and group_by function. The pdf has a visual showing how a data table gets transformed after using both functions (and the combination of both function). It also shows the basic syntax one would use in order to use the functions. Visually seeing how group_by allows each one of the computation summarise does to be done separately for different variables was helpful in visually conceptually how the functions work.

> part c - Most of the packages we use have vignettes that have been created for them. Vignettes are designed to show how functions are used. Identify either a function related to your concept or your selected verb (which is a function), and find what package it is in. Then look for a package vignette. What package did you look for a vignette for? Is your concept or verb illustrated in the vignette?

(Searching with Google or within R are possible.)

Solution: I used for vignettes for the dplyr package. I found that there is a vignette for the 'grouped data' associated with this package that covers how group_by works in a variety of examples. The vignette also describes how to use group_by in conjunction with other verbs like summarise.

> part d - Many people blog examples of different R functions. Search for an R example of your concept or verb using Google. Look over the search results and identify one that demonstrates correct use of the concept or verb. List the URL.

Solution: Here is the URL for the blogpost I found: ["https://cmdlinetips.com/2020/08/dplyr-groupby-and-summarize-group-by-one-or-more-variables/"]

2 - MDSR 5.2

Use the `Batting`, `Pitching`, and `Master` tables in the *Lahman* package to answer the following questions. Remember that you are responsible for loading packages in the setup chunk.

> part a - List the name of every player in baseball history who has accumulated at least 300 home runs (HR) and at least 300 stolen bases (SB). You can find the first and last name of the player in the `Master` data frame. Join this to your result along with the total home runs and total bases stolen for each of these elite players.

Solution:

```
top300 <- Batting %>%
  left_join(Master, by = c("playerID")) %>%
  select(nameFirst, playerID, nameLast, HR, SB) %>%
  group_by(playerID, nameFirst, nameLast) %>%
  summarize(HR = sum(HR), SB = sum(SB)) %>%
  filter(HR > 300 & SB > 300) %>%
  subset(select = -playerID) %>%
  rename("First Name" = nameFirst,
         "Last Name" = nameLast,
         "Home Runs" = HR,
         "Stolen Bases" = SB) %>%
  kable(booktabs = TRUE)
```

```
## `summarise()` has grouped output by 'playerID', 'nameFirst'. You can override
## using the `.groups` argument.
```

```
top300
```

First Name	Last Name	Home Runs	Stolen Bases
Carlos	Beltran	435	312
Barry	Bonds	762	514
Bobby	Bonds	332	461
Andre	Dawson	438	314
Steve	Finley	304	320
Willie	Mays	660	338
Alex	Rodriguez	696	329
Reggie	Sanders	305	304

> part b - Similarly, list the names every pitcher in baseball history who has accumulated at least 300 wins (W) and at least 3,000 strikeouts (SO).

Solution:

```
top300b <- Pitching %>%
  left_join(Master, by = c("playerID")) %>%
  select(nameFirst, playerID, nameLast, SO, W) %>%
  group_by(playerID, nameFirst, nameLast) %>%
  summarize(SO = sum(SO), W = sum(W)) %>%
```

```
  filter(SO > 3000 & W > 300) %>%
  subset(select = -playerID) %>%
  rename("First Name" = nameFirst,
         "Last Name" = nameLast,
         "Wins" = W,
         "Strikeouts" = SO) %>%
  kable(booktabs = TRUE)
```

```
## `summarise()` has grouped output by 'playerID', 'nameFirst'. You can override
## using the `.groups` argument.
```

```
top300b
```

First Name	Last Name	Strikeouts	Wins
Steve	Carlton	4136	329
Roger	Clemens	4672	354
Randy	Johnson	4875	303
Walter	Johnson	3509	417
Greg	Maddux	3371	355
Phil	Niekro	3342	318
Gaylord	Perry	3534	314
Nolan	Ryan	5714	324
Tom	Seaver	3640	311
Don	Sutton	3574	324

part c - Finally, list the name and year of every player who has hit at least 50 home runs in a single season. Which player had the lowest batting average in that season?

Note: Batting average is calculated as the number of hits (H) divided by the number of at bats (AB).

Solution: Pete Alonso's 2019 season involved the lowest batting average of players who hit at least 50 home runs in a single season.

```
homeruns <- Batting %>%
  left_join(Master, by = c("playerID")) %>%
  select(nameFirst, yearID, nameLast, playerID, HR, H, AB) %>%
  filter(HR > 50) %>%
  group_by(yearID, nameFirst, nameLast) %>%
  summarize(battingaverage = H/AB) %>%
  arrange(battingaverage) %>%
  rename("First Name" = nameFirst,
         "Last Name" = nameLast,
         "Batting Average" = battingaverage) %>%
  kable(booktabs = TRUE)
```

```
## `summarise()` has grouped output by 'yearID', 'nameFirst'. You can override
## using the `.groups` argument.
```

```
homeruns
```

yearID	First Name	Last Name	Batting Average
2019	Pete	Alonso	0.2596315
2010	Jose	Bautista	0.2601054
2005	Andruw	Jones	0.2627986
1961	Roger	Maris	0.2694915
1990	Cecil	Fielder	0.2774869
1999	Mark	McGwire	0.2783109
2017	Giancarlo	Stanton	0.2814070
2017	Aaron	Judge	0.2841328
1998	Ken	Griffey	0.2843602
2013	Chris	Davis	0.2859589
2006	David	Ortiz	0.2867384
1999	Sammy	Sosa	0.2880000
1998	Mark	McGwire	0.2986248
2002	Alex	Rodriguez	0.2996795
1947	Johnny	Mize	0.3020478
2002	Jim	Thome	0.3041667
1997	Ken	Griffey	0.3042763
1998	Sammy	Sosa	0.3079316
1949	Ralph	Kiner	0.3096539
1996	Mark	McGwire	0.3120567
2006	Ryan	Howard	0.3132530
1947	Ralph	Kiner	0.3132743
2007	Alex	Rodriguez	0.3138937
1938	Hank	Greenberg	0.3147482
1961	Mickey	Mantle	0.3171206
1965	Willie	Mays	0.3172043
2001	Alex	Rodriguez	0.3180380
1955	Willie	Mays	0.3189655
1977	George	Foster	0.3203252
1928	Babe	Ruth	0.3227612
2001	Luis	Gonzalez	0.3251232
2001	Sammy	Sosa	0.3275563
2001	Barry	Bonds	0.3277311
1956	Mickey	Mantle	0.3527205
1927	Babe	Ruth	0.3555556
1930	Hack	Wilson	0.3555556
1932	Jimmie	Foxx	0.3641026
1920	Babe	Ruth	0.3763676
1921	Babe	Ruth	0.3777778

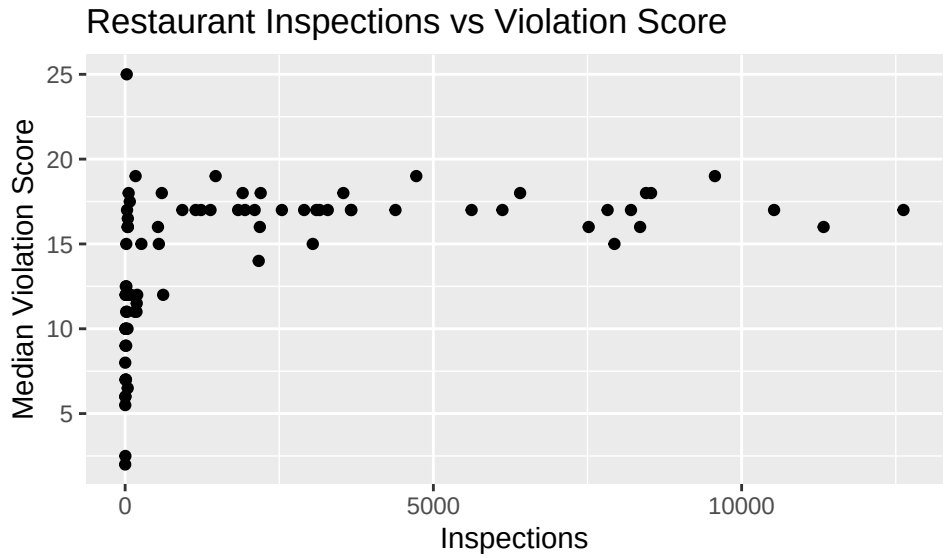
3 - MDSR 4.11 (modified)

The `Violations` data set in the **mdsr** package contains information regarding the outcome of health inspections of restaurants in New York City. Note that higher inspection scores indicate worse violations: "restaurants with an inspection score between 0 and 13 points earn an A, those with 14 to 27 points receive a B and those with 28 or more a C" (nyc.gov).

> part a - Use these data to calculate the median violation score by zip code for zip codes in Manhattan. What pattern, if any, do you see between the number of inspections and the median score? Generate a visualization to support your response.

Solution: Restaurants with more than 100-200 inspections appear to have higher median violation scores than restaurants with less inspections, but beyond that point there does not appear to be a relationship between inspections and median violation score.

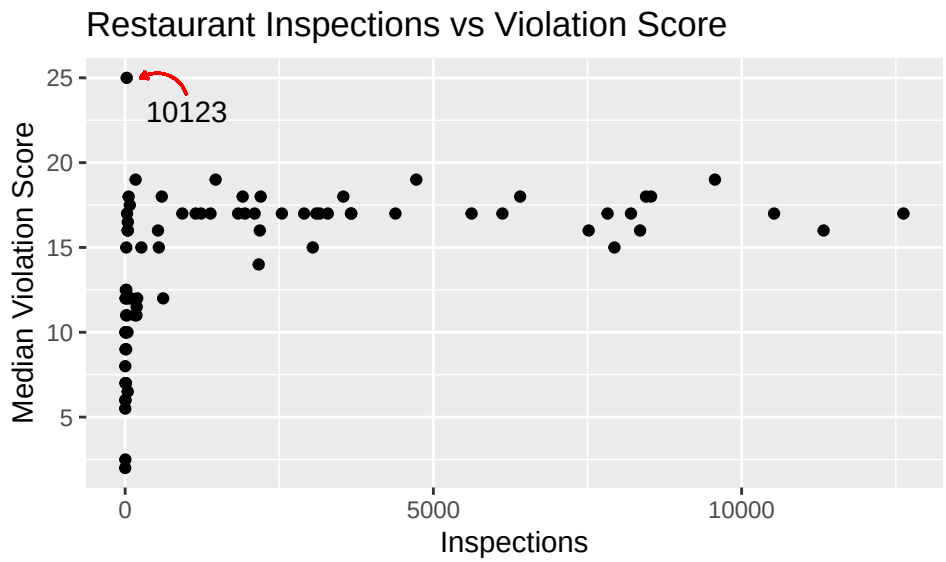
```
median_violation <- Violations %>%
  select(boro, zipcode, score) %>%
  na.omit() %>%
  filter(boro == "MANHATTAN") %>%
  group_by(zipcode) %>%
  summarize(medianviolations = median(score),
            inspections = n())


p <- ggplot(data = median_violation, aes(x = inspections, y =medianviolations)) +
  geom_point() +
  labs(title = "Restaurant Inspections vs Violation Score",
       y = "Median Violation Score",
       x = "Inspections")
p
```



> part b - In your visualization above, there are several potential outliers but there is one zipcode in particular that does not seem to fall along the general trend. Add text to the outlier identifying what zipcode it is, and add an arrow pointing from the text to the observation. Note: first, you may want to `filter()` to identify the zipcode (so you know what text to add to the plot).

Solution:

```
findviolation <- median_violation %>%
  filter(medianviolations > 20)
p <- ggplot(data = median_violation, aes(x = inspections, y =medianviolations)) +
  geom_point() +
  annotate("text", x = 1000, y = 23, label = "10123") +
  geom_curve(aes(x = 1000, y = 24, xend = 250, yend = 25),
             arrow = arrow(length = unit(.02, "npc")),
             color = "red") +
  labs(title = "Restaurant Inspections vs Violation Score",
       y = "Median Violation Score",
       x = "Inspections")
p
```



4 - MDSR 6.5

Generate the code to convert the data frame from the starting point to the results.

Figures available in text online in Section 6.6.

The starting data frame is provided. Hint (from text): Use *pivot_longer()* in conjunction with *pivot_wider()*.

```
OrigData <- data.frame(grp = c("A","A","B", "B")
                       , sex = c("F", "M", "F", "M")
                       , meanL = c(0.22, 0.47, 0.33, 0.55)
                       , sdL = c(0.11, 0.33, 0.11, 0.31)
                       , meanR = c(0.34, 0.57, 0.40, 0.65)
                       , sdR = c(0.08, 0.33, 0.07, 0.27))
```

Solution:

```
newData <- OrigData %>%
pivot_longer(cols = meanL:sdR,
             names_to = "class",
             values_to = "values")  %>%
pivot_wider(
  names_from = c(class, sex),
  names_glue = ("{sex}.{class}"),
  values_from = c(values))
head(newData)
```

```
## # A tibble: 2 x 9
##   grp   F.meanL F.sdL F.meanR F.sdR M.meanL M.sdL M.meanR M.sdR
##   <chr>   <dbl> <dbl>   <dbl> <dbl>   <dbl> <dbl>   <dbl> <dbl>
## 1 A        0.22  0.11    0.34  0.08    0.47  0.33    0.57  0.33
## 2 B        0.33  0.11    0.4   0.07    0.55  0.31    0.65  0.27
```

5 - Combining your Wrangling and Visualization Skills

When we looked at our first UN votes visual, some wrangling was required to get the data into a format appropriate for the visual. Now that we've examined both visualization and wrangling, you can combine the skills too! (And you did a little of this above).

We will be looking at a data set on high school students in Portugal. We have information on their performance in a Math course and a Portugeuse course (think of this as your natural language course, i.e. English for English speakers, etc.), as well as a host of demographic variables. Detailed information about the data set is provided on the following pages - you should look it over as you tackle this problem. (Feel free to remove the info when knitting to the final version of your assignment.)

We want to visualize the relationship between final Math and final Portugeuse grade for students who were in both courses. In addition, we want to be sure all students in the visual were under 20 years old, and had fewer than 10 absences in either course (not total). We also want to factor in weekend alcohol use and travel time as reported in the Math data set in our examination of the relationship, treating these as appropriate group variables (categorical). (Students filled out the survey twice and not all responses match between them, even for the same student.)

1. Wrangle the data you need into an appropriate format, and save it as a new data set with the variables you need for your visual.

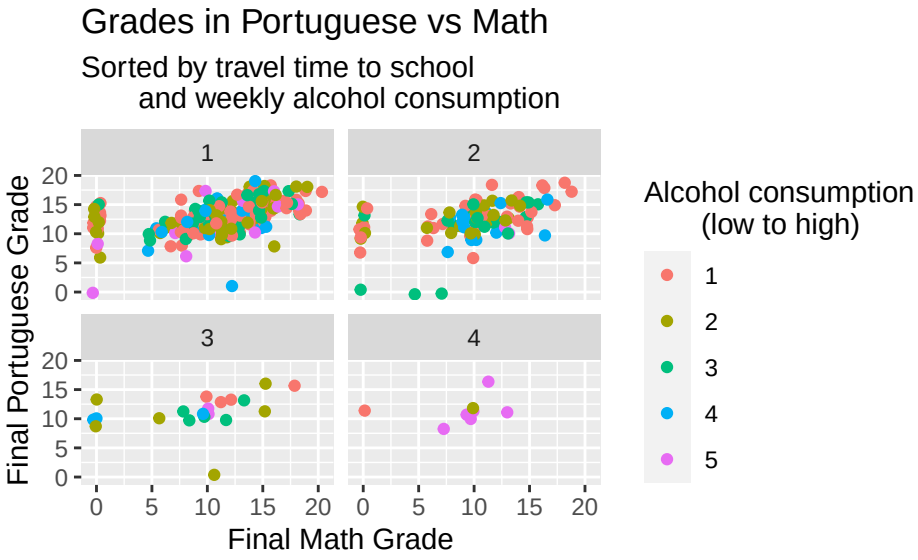
Solution:

```
oldgrades <- math_data %>%
  inner_join(port_data,
             c("school", "sex", "age",
               "address", "famsize","Pstatus",
               "Medu","Fedu","Mjob",
               "Fjob","reason","nursery",
               "internet"))

newgrades <- oldgrades %>%
  filter(age < 20) %>%
  filter(absences.x < 10 | absences.y < 10) %>%
  select(G3.x,
         G3.y, traveltime.x,
         Walc.x) %>%
  mutate(alcohol_consumption = as.factor(Walc.x)) %>%
  mutate(travel_time = as.factor(traveltime.x))
```

2. Then generate an appropriate visual. Make sure your graphic has appropriate labels, legends (as needed), and a title.

Solution:

```
p <- ggplot(data = newgrades, aes(x = G3.x, y = G3.y, color = alcohol_consumption)) +
  facet_wrap(~travel_time, nrow = 2) +
  geom_jitter() +
  labs(title = "x", x = "Final Math Grade", y = "Final Portuguese Grade") +
  labs(title = "Grades in Portuguese vs Math",
       subtitle = "Sorted by travel time to school
       and weekly alcohol consumption", color = "Alcohol consumption
       (low to high)")
p
```



3. Finally, in a few sentences, describe what you find.

Solution: There is a positive relationship between math grades and Portuguese grades in general. There appears to be a much weaker relationship between the two final grades when travel time is highest. Alcohol consumption does not appear to have a very strong effect on final grades.

Data Set Information for Problem 5

The data set is from a paper called "Using Data Mining To Predict Secondary School Student Alcohol Consumption" by Fabio Pagnotta and Hossain Mohammad Amran of the Department of Computer Science, University of Camerino, and the data set is hosted online in UCI's machine learning repository.

The information below was copied from the provided codebook online.

Attributes for both student-mat.csv (Math course) and student-por.csv (Portuguese language course) datasets:
1. school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. sex - student's sex (binary: 'F' - female or 'M' - male)
3. age - student's age (numeric: from 15 to 22)
4. address - student's home address type (binary: 'U' - urban or 'R' - rural)
5. famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7. Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
9. Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12. guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13. traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. failures - number of past class failures (numeric: n if 1<=n<3, else 4)
16. schoolsup - extra educational support (binary: yes or no)
17. famsup - family educational support (binary: yes or no)
18. paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. activities - extra-curricular activities (binary: yes or no)
20. nursery - attended nursery school (binary: yes or no)
21. higher - wants to take higher education (binary: yes or no)
22. internet - Internet access at home (binary: yes or no)
23. romantic - with a romantic relationship (binary: yes or no)
24. famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25. freetime - free time after school (numeric: from 1 - very low to 5 - very high)
26. goout - going out with friends (numeric: from 1 - very low to 5 - very high)
27. Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28. Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29. health - current health status (numeric: from 1 - very bad to 5 - very good)
30. absences - number of school absences (numeric: from 0 to 93)

Finally, the grades are related with the course subject, Math or Portuguese:
31. G1 - first period grade (numeric: from 0 to 20)
32. G2 - second period grade (numeric: from 0 to 20)
33. G3 - final grade (numeric: from 0 to 20, output target)
Thus, these variables appear in each data set, but have different meaning in each.

The data was provided as two different .csv files online. I obtained some errors trying to work with them, so ended up saving them as .txt files on my website. Many of the students were in both courses, but not all.