STAT 231: Problem Set 5B

Jack Dove

due by 5 PM on Friday, October 2

This homework assignment is designed to help you futher ingest, practice, and expand upon the material covered in class over the past week(s). You are encouraged to work with other students, but all code and text must be written by you, and you must indicate below who you discussed the assignment with (if anyone).

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps5B.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps5B.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

If you discussed this assignment with any of your peers, please list who here:

ANSWER:

1. Justices of the Supreme Court of the United States

a. Confirm (using an R command) that the following Wikipdeia page allows automated scraping: https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States

```
paths_allowed("https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States
```

```
##  en.wikipedia.org
```

```
## [1] TRUE
```

b. Go to the List of Justices of the Supreme Court of the United States and scrape the table for the Justices. Write, test, and save your code in an R script called `scrape_justices.R`, and write the data frame out to a csv file called `justices.csv` using the `write_csv` function.

Be sure to push your .R and .csv files to your GitHub repo.

```
## Add your code that is in justices.R to this code chunk.  KEEP the "eval=FALSE" option in this code c
url <- "https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States"

justices <- url %>%
  read_html() %>%
  html_node(xpath = '//*[@id="mw-content-text"]/div[1]/table[2]') %>%
  html_table(fill = TRUE)

write_csv(justices, path = "justices.csv")
```

c. Load `justices.csv` into this file using the `read_csv` function. Then, run the code given below to create the variable `tenure_length` (a numeric variable containing each justice's tenure on the bench).

Create a visualization to show the distribution of tenure length of U.S. Supreme Court judges. Interpret the plot.

ANSWER: Most justices served 5-15 year terms, but the distribution of justice tenures is fairly widespread, ranging between 0-40 years.

```
justices <- read_csv("~/Desktop/Data Science/Stat231JackDove/Labs/justices.csv")
```

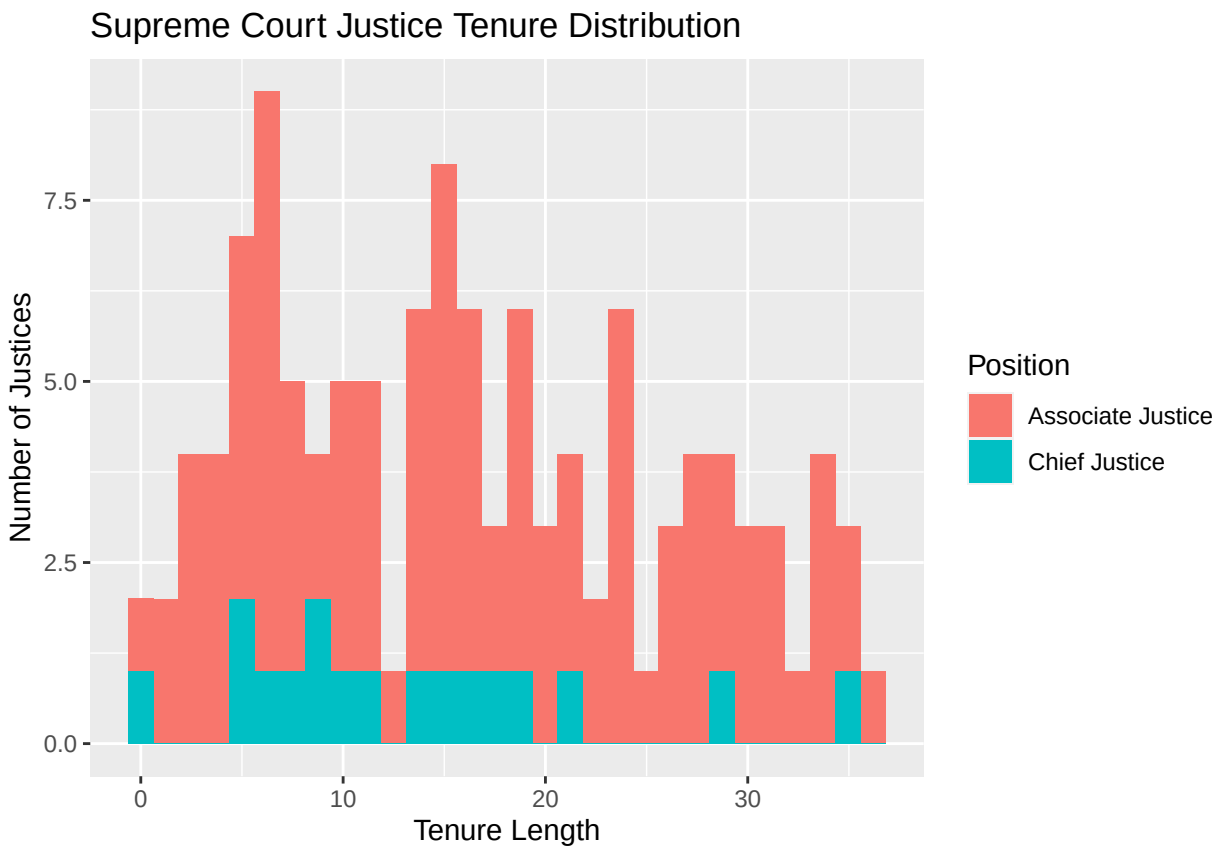
```
## Warning: Duplicated column names deduplicated: 'Justice' => 'Justice_1' [2],
## 'Justice' => 'Justice_2' [3]
```

```
## Parsed with column specification:
## cols(
##   Justice = col_character(),
##   Justice_1 = col_character(),
##   Justice_2 = col_character(),
##   `State[c]` = col_character(),
##   Position = col_character(),
##   Succeeded = col_character(),
##   `Date confirmed(Vote)` = col_character(),
##   Tenure = col_character(),
##   `Tenure length[d]` = col_character(),
##   `Nominated by` = col_character()
## )
```

```
justices2 <- justices %>%
  clean_names() %>%
  # remove extra line that comes in at end of table
  filter(justice != "Justice") %>%
  # some justices served less than 1 year, adjust their length so can
  # separate correctly
  mutate(tenure_length_temp = case_when(str_detect(tenure_length_d, "year") ~ tenure_length_d
                                        , TRUE ~ paste0("0 years, ", tenure_length_d))) %>%
  separate(tenure_length_temp, into = c("years_char", "days_char"), sep = ","
           , remove = FALSE) %>%
  mutate(tenure_length = parse_number(years_char) + (parse_number(days_char)/365)) %>%
  # create date confirmed as date variable
  separate(date_confirmed_vote, into = c("date_confirmed_vote", "extra")
           , sep = "\\(") %>%
  mutate(date_confirmed = lubridate::mdy(date_confirmed_vote))

plot <- justices2 %>%
  ggplot(aes(x=tenure_length, fill=position)) + geom_histogram() +
  labs(x="Tenure Length", y = "Number of Justices",
       title = "Supreme Court Justice Tenure Distribution", legend="Position") +
  scale_fill_discrete(name="Position",
                      breaks=c("AssociateJustice", "ChiefJustice"),
                      labels=c("Associate Justice", "Chief Justice"))
plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2. Brainy Quotes

a. Confirm (using an R command) that automated scraping of the Brainy Quote webpage (https://www.brainyquote.com/) is allowed.

```
paths_allowed("https://www.brainyquote.com/")
```

```
##  www.brainyquote.com
```

```
## [1] TRUE
```

b. Life can get frustrating at times. Like when we're trying to Zoom and our internet cuts out. Or when we can't figure out why R's throwing an error when we try to clone a GitHub repo in RStudio. Or, when COVID-19 upends life as we knew it. In these times, it can't hurt to be reminded of the power of persistence, resilience and optimism.

The code in the first R code chunk below scrapes the first 40 quotes returned from a search for "resilience" on BrainyQuote.com. (Do NOT remove the "eval = FALSE" option from that code chunk; you do not want it to evaluate it, i.e. scrape the site, every time you knit this file.)

The code in the second R code chunk below randomly selects a quote and prints it. When you're feeling frustrated, run that code chunk to randomly generate a quote to lift you up (or just make you laugh at the uselessness of the quote; some of them are pretty pathetic . . .).

Note that CSS selector gadget was used to identify the key words to specify in the `html_nodes` function (i.e. ".oncl_q" and ".oncl_a"). These key words will vary depending on what webpage and what particular objects from that webpage you're trying to scrape.

```
quotes_html <- read_html("https://www.brainyquote.com/topics/resilience-quotes")

quotes <- quotes_html %>%
  html_nodes(".oncl_q") %>%
  html_text()

person <- quotes_html %>%
  html_nodes(".oncl_a") %>%
  html_text()

# put in data frame with two variables (person and quote)
quotes_dat <- data.frame(person = person, quote = quotes
                         , stringsAsFactors = FALSE) %>%
  mutate(together = paste('"', as.character(quote), '" --'
                          , as.character(person), sep=""))
```

```
quotes_dat <- read_csv("http://kcorreia.people.amherst.edu/F2021/resilience_quotes.csv")
```

```
## Parsed with column specification:
## cols(
##   person = col_character(),
##   quote = col_character(),
##   together = col_character()
## )
```

```
quote_for_the_day <- quotes_dat[sample(1:nrow(quotes_dat), size = 1),]

quote_for_the_day$together
```

```
## [1] "\"When fear rushed in, I learned how to hear my
heart racing but refused to allow my feelings to sway me.
That resilience came from my family. It flowed through our
bloodline.\" --Coretta Scott King"
```

Go to BrainyQuote.com and search a different topic (or search an Author) that interests you. Scrape the webpage returned from your search following the same code given above. Save your code in an R script

called `scrape_quotes.R`, and write the data frame out to a csv file called `quotes.csv` using the `write_csv` function.

Be sure to push your .R and .csv files to your GitHub repo.

```
## Add your code that is in quotes.R to this code chunk.  KEEP the "eval=FALSE" option in this code chu
quotes_html <- read_html("https://www.brainyquote.com/topics/sports-quotes")

quotes <- quotes_html %>%
  html_nodes(".oncl_q") %>%
  html_text()

person <- quotes_html %>%
  html_nodes(".oncl_a") %>%
  html_text()

# put in data frame with two variables (person and quote)
quotes_dat <- data.frame(person = person, quote = quotes
                         , stringsAsFactors = FALSE) %>%
  mutate(together = paste('"', as.character(quote), '" --'
                          , as.character(person), sep=""))

write_csv(quotes_dat, path = "quotes.csv")
```

c. Load `quotes.csv` into this file using the `read_csv` function. Write code to select *three* of the quotes at random and print them (i.e., set `size = 3` in the `sample` function).

```
quotes <- read_csv("~/Desktop/Data Science/Stat231JackDove/Labs/quotes.csv")
```

```
## Parsed with column specification:
## cols(
##   person = col_character(),
##   quote = col_character(),
##   together = col_character()
## )
```

```
quotes_for_the_day <- quotes[sample(1:nrow(quotes), size = 3),]

quotes_for_the_day$together
```

```
## [1] "\"The game of golf would lose a great deal if
croquet mallets and billiard cues were allowed on the
putting green.\" --Ernest Hemingway"
## [2] "\"Winning is habit. Unfortunately, so is losing.\"
--Vince Lombardi"
## [3] "\"Gold medals aren't really made of gold. They're
made of sweat, determination, and a hard-to-find alloy
called guts.\" --Dan Gable"
```