STAT 231: Problem Set 5B

Matthew Perkins

due by 10 PM on Friday, March 26

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

```r
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.

```r
## Add your code that is in justices.R to this code chunk.
## KEEP the "eval=FALSE" option in this code chunk option, as you do NOT want to
## evaluate it (which would re-scrape the website every time you knit this file).
## (We do want the grader to be able to see the code, though, as you'll only
## submit this one knit PDF to Gradescope.)

url <-
"https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States"
tables <- url %>%
  read_html() %>%
  html_nodes("table")
length(tables)
justices <- tables %>%
  purrr::pluck(2) %>%
  html_table(fill = TRUE)
justices
write_csv(justices, "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: The plot shows that justices have served a wide range of tenures on the Supreme Court, ranging from ~0-40 years. The most common tenure lengths are 0-10 and 10-20 years which have a nearly equal number of justices. 20-30 year tenures are less common, and 30-40 year tenures are fairly rare.

```
justices <- read_csv("justices.csv")
```

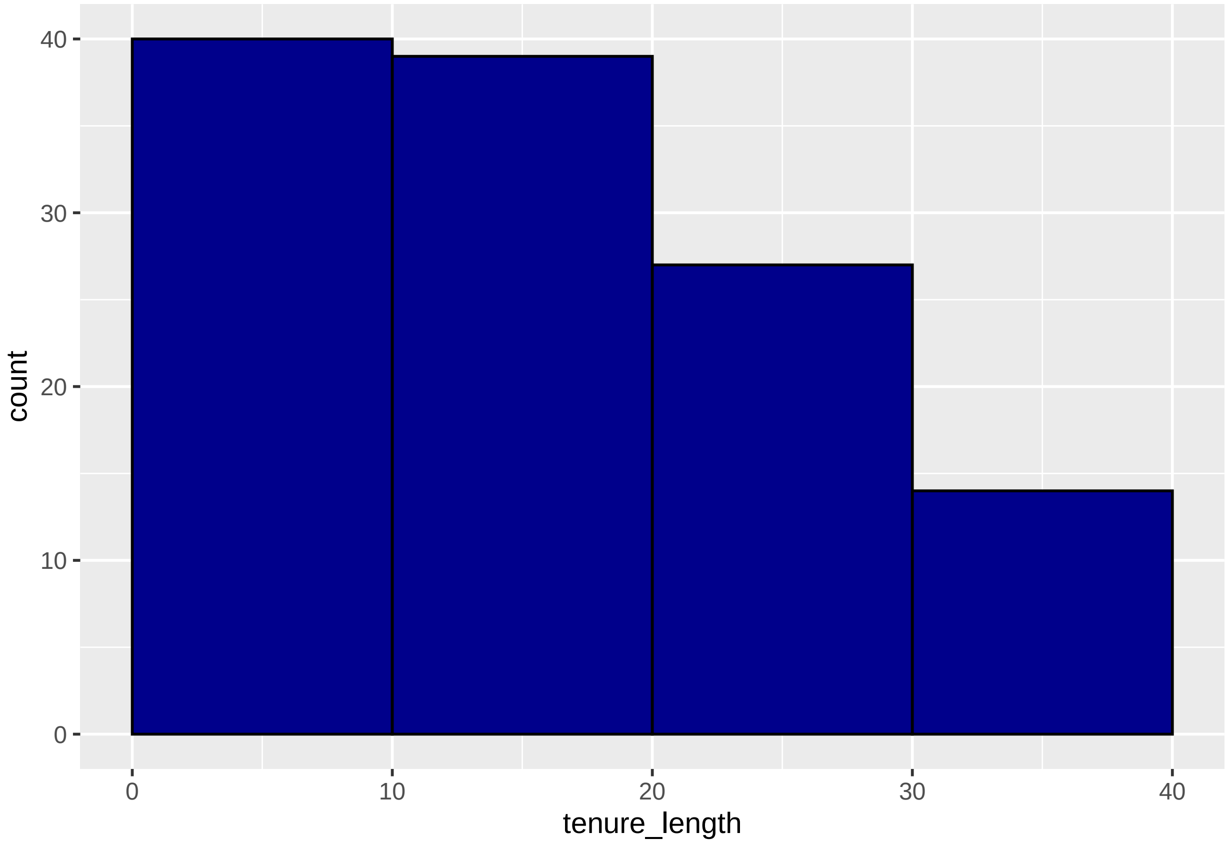
```
## Warning: Duplicated column names deduplicated: 'Justice' => 'Justice_1' [2],
## 'Justice' => 'Justice_2' [3]
```

```
##
## -- 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))

justices3 <- justices2 %>%
  select(tenure_length)
```

```
ggplot(data = justices3, mapping = aes(x=tenure_length)) +
geom_histogram(binwidth = 10 ,boundary=0, color = "black", fill = "darkblue")
```



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(quote = quotes, stringsAsFactors = FALSE) %>%
  filter(quote != "\n") %>%
  mutate(person = person
         , together = paste('"', as.character(quote), '" --'
                            , as.character(person), sep=""))
```

```
quotes_dat <- read_csv("http://kcorreia.people.amherst.edu/S2021/resilience_quotes.csv")
```

```
##
## -- 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),]

cat(quote_for_the_day$together)
```

```
## "As my kids grow up, I think a lot about the lessons and
values I want to impart to them. More than any particular
skill or even financial support, I believe perseverance and
resilience will serve them best, regardless of what
curveball life inevitably throws them." --Sal Khan
```

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.

```r
## Add your code that is in justices.R to this code chunk.
## KEEP the "eval=FALSE" option in this code chunk option, as you do NOT want to
## evaluate it (which would re-scrape the website every time you knit this file).
## (We do want the grader to be able to see the code, though, as you'll only
## submit this one knit PDF to Gradescope.)

quotes_funny_html <- read_html("https://www.brainyquote.com/topics/funny-quotes")

quotes_funny <- quotes_funny_html %>%
  html_nodes(".oncl_q") %>%
  html_text()

person_funny <- quotes_funny_html %>%
  html_nodes(".oncl_a") %>%
  html_text()

# put in data frame with two variables (person and quote)
quotes_funny_dat <- data.frame(quote = quotes_funny, stringsAsFactors = FALSE) %>%
  filter(quote != "\n") %>%
  mutate(person = person_funny
         , together = paste('"', as.character(quote), '" --'
                          , as.character(person), sep=""))

write_csv(quotes_funny_dat, "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).

```
funnyQuotes <- read_csv("quotes.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   quote = col_character(),
##   person = col_character(),
##   together = col_character()
## )
```

```
quotes_random <- funnyQuotes[sample(1:nrow(funnyQuotes), size = 3),]

cat(quotes_random$together)
```

```
## "People who think they know everything are a great
annoyance to those of us who do." --Isaac Asimov
"Everything I like is either illegal, immoral or
fattening." --Alexander Woollcott "Tragedy is when I cut my
finger. Comedy is when you fall into an open sewer and
die." --Mel Brooks
```