Practice Set 5

Kauila Tengan

Due by 10pm ET on Friday

Practice Set Information

During the week, you will get further practice with the material by working through the Practice Set, a set of problems designed to give you practice beyond the examples produced in the text.

You may work through these problems with peers, but all work must be completed by you (see the Honor Code in the syllabus) and you must indicate who you worked with below.

Even then, the best approach here is to try the problems on your own before discussing them with peers, and then write your final solutions yourself.

GitHub Workflow

1. Before editing this file, verify you are working on the copy saved in *your* repo for the course (check the filepath and the project name in the top right corner).
2. Before editing this file, make an initial commit of the file to your repo to add your copy of the problem set.
3. Change your name at the top of the file and get started!
4. You should *save, knit, and commit* the .Rmd file each time you've finished a question, if not more often. You should also *push* your commits back onto GitHub occasionally (you can do this after each commit).
5. When you think you are done with the assignment, save the pdf as "*Name_thisfilename_date*.pdf" before committing and pushing (this is generally good practice but also helps me in those times where I need to download all student homework files).

Gradescope Upload

For each question (e.g., 3.1), allocate all pages associated with the specific question. If your work for a question runs onto a page that you did not select, you may not get credit for the work. If you do not allocate *any* pages when you upload your pdf, you may get a zero for the assignment.

You can resubmit your work as many times as you want before the deadline, so you should not wait until the last minute to submit some version of your work. Unexpected delays/crises that occur on the day the assignment is due do not warrant extensions (please submit whatever you have done to receive partial credit).

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (notes, textbook, etc) 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)

- SDS fellow, Zack for problem 1 and 2

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

-

Problem 1 **Justices of the Supreme Court of the United States**

1.1 Confirm that the following Wikipedia page allows automated scraping: https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States

```
url <- "https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States"

paths_allowed(url)
```

```
[1] TRUE
```

1.2 Go to the List of Justices of the Supreme Court of the United States. Create a new R script called "scrape-justices.R", and scrape the table of justices. Use `janitor::clean_names()` to tidy the names of the columns (do not do any additional wrangling beyond this for now), then write the final data frame to a csv file called "justices.csv" in the "data" folder using the `write_csv()` function. Commit and push both files in addition to this Rmd file when ready.

```
url <- "https://en.wikipedia.org/wiki/List_of_justices_of_the_Supreme_Court_of_the_United_States"

tables <- url %>%
  read_html() %>%
  html_elements("table")
#pluck the html table
scj <- html_table(tables[[2]], fill = TRUE) %>%
  #clean the names
  janitor::clean_names()
#create an object for outpath
# outpath <- "/Users/tpkt/Desktop/College2021-2022/Stats231/Personal Repo/Stat-231-KT/ps5/data"
#write the csv file pulling scj table and paste0 combines outpath with writecsv
# write_csv(scj, paste0(outpath, "/justices.csv"))
write_csv(scj, "data/justices.csv")
```

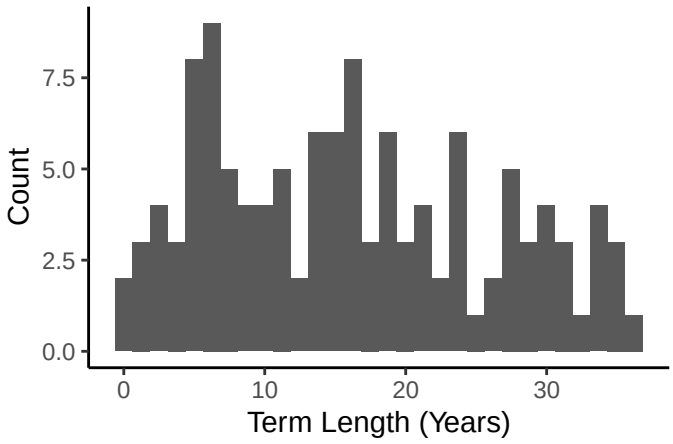
1.3 Load "justices.csv" into this file using the `read_csv()` function. Then, modify the code below as needed and run the code to create the variable `tenure_length` (a numeric variable containing each justice's time spent on the bench). Create a visualization to show the distribution of tenure length of U.S. Supreme Court judges. Interpret the plot.

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

# Run after loading your justices.csv file
justices <- justices %>%
  # Remove extra line that comes in at end of table
```

```
  filter(justice != "Justice") %>%
  # Some justices served <1 year; add "0 years," to make separating easier
  mutate(tenure_length = case_when(str_detect(tenure_length_d, "year") ~ tenure_length_d,
                                   TRUE ~ paste0("0 years, ", tenure_length_d))) %>%
  separate(tenure_length, into = c("years_char", "days_char"),
           sep = ", ") %>%
  mutate(tenure_years = parse_number(years_char) + (parse_number(days_char)/365)) %>%
  # Make date_confirmed_vote into date variable
  separate(date_confirmed_vote, into = c("date_confirmed", "vote"),
           sep = "\\(") %>%
  mutate(tenure_length = tenure_years) %>%
  mutate(date_confirmed = lubridate::mdy(date_confirmed))

ggplot(justices, aes(x = tenure_length)) +
  geom_histogram() +
  labs(x = "Term Length (Years)",
       y = "Count")
```



Problem 2 **Brainy Quotes** One theme of college (and life) is resilience. The code in the `scrape-resilience` chunk below scrapes a set of quotes returned from a search for "resilience" on BrainyQuote.com. The code in `display-quote` randomly selects one of those quotes 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).

```
quotes_url <- "https://www.brainyquote.com/topics/resilience-quotes"
robotstxt::paths_allowed(quotes_url)

quotes_html <- read_html(quotes_url)

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

quotes <- quotes[which(quotes != " ")]

people <- quotes_html %>%
  html_elements(".oncl_a") %>%
  html_text()

# put in data frame with two variables (person and quote)
resilience_quotes <- tibble(person = people, quote = quotes) %>%
  mutate(quote = str_remove_all(quote, "\n\n"),
         # Prep quotes for markdown display when `results = "asis"`
         display = paste0('> *"', as.character(quote), '"* --' , as.character(person)))

write_csv(resilience_quotes, "data/resilience_quotes.csv")
```

```
resilience_quotes <- read_csv("data/resilience_quotes.csv")

slice_sample(resilience_quotes, n = 1) %>%
  pull(display) %>%
  cat()
```

> *"Because, you know, resilience - if you think of it in terms of the Gold Rush, then you'd be pretty depressed right now because the last nugget of gold would be gone. But the good thing is, with innovation, there isn't a last nugget. Every new thing creates two new questions and two new opportunities."* –Jeff Bezos

2.1 Go to BrainyQuote.com and search a different topic or 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 to a csv called "quotes.csv" in the "data" folder. Be sure to push your R and csv files to your GitHub repo.

```
quotes_url <- "https://www.brainyquote.com/topics/funny-quotes_17"
robotstxt::paths_allowed(quotes_url)

quotes_html <- read_html(quotes_url)

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

people <- quotes_html %>%
  html_elements(".oncl_a") %>%
  html_text()

funny_quotes <- tibble(person = people, quote = quotes) %>%
  mutate(quote = str_remove_all(quote, "\n\n"),
         # Prep quotes for markdown display when `results = "asis"`
         display = paste0('> *"', as.character(quote), '"* --' , as.character(person)))

# outpath <- "/Users/tpkt/Desktop/College2021-2022/Stats231/Personal Repo/Stat-231-KT/ps5/data"
# write_csv(funny_quotes, paste0(outpath, "/quotes.csv"))
write_csv(funny_quotes, "data/quotes.csv")
```

2.2 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 `n = 3` in the `slice_sample()` function).

```
funny_quotes <- read_csv("data/quotes.csv")

slice_sample(funny_quotes, n = 3) %>%
  pull(display) %>%
  cat()
```

> *"It was funny actually because that was still during the time we were dating. He would get all these calls because supposedly before we broke up, we had already broken up in the trades, in the rags or whatever."* –Rosario Dawson > *"I can't even look at daily comic strips. And I hate sitcoms because they don't seem like real people to me: they're props that often say horrible things to each other, which I don't find funny. I have to feel like they're real people."* –Roz Chast > *"Early on, many years ago when we started 'Avatar,' the executive that we were working with said to make the sad scenes sadder, the funny scenes funnier, the scary scenes scarier. That was kind of permission to do what we felt comfortable with."* –Bryan Konietzko