Problem Set 3

Insert Name

Stat 108, Week 4

Collaborators

I collaborated with… (list names of collaborators here).

```
# Put all necessary libraries here
# We got you started!
library(tidyverse)
```

Due: Wednesday, March 1st at 10:00pm

Goals of this problem set

1. Practice more data wrangling with `dplyr` and `tidyr`.
2. Explore and study the behavior of various `R` objects.
3. Practice subsetting `R` objects.

Problems

Problem 1

As we discussed in lecture, if you concatenate two atomic vectors of different classes, `R` will change the class of one of the vectors. For example, when you concatenate a numeric vector and a logical vector, you get a numeric.

```
# Example of concatenating numeric and logical
x <- c(1, 5, TRUE)
class(x)
```

```
## [1] "numeric"
```

In this problem, we want you to determine `R`'s implicit type conversion hierarchy for the following classes: factor, character, numeric, logical. Make sure to provide examples (like the one above) that illustrate the hierarchy and to write out the rules `R` follows when combining these vector types. (There might be some strange behavior.)

Problem 2

Here are some R objects. In this problem we are going to ask for particular subsets and want you to use `[]` and/or `[[]]` to obtain these pieces.

```
a <- c(1, 6, 8, 3) > 5

b <- c("hi", "howdy")

X <- data.frame(a, b = LETTERS[1:4], c = rep(1, 4))
```

```
ALL <- list(a = a, b = b, X = X)

a
```

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

```
b
```

```
## [1] "hi"    "howdy"
```

```
X
```

```
##       a b c
## 1 FALSE A 1
## 2  TRUE B 1
## 3  TRUE C 1
## 4 FALSE D 1
```

```
ALL
```

```
## $a
## [1] FALSE  TRUE  TRUE FALSE
##
## $b
## [1] "hi"    "howdy"
##
## $X
##       a b c
## 1 FALSE A 1
## 2  TRUE B 1
## 3  TRUE C 1
## 4 FALSE D 1
```

a. Provide the first and fourth entries of `a`.

b. Provide the third row of `X`.

c. Provide two ways to print `"howdy"`.

d. Provide the second column of the third entry in `ALL`.

Problem 3

Let’s return to the IMDB example we discussed in class and practice joining a few more datasets pulled from that database.

a. Do some data joins on the following datasets so that you produce a dataset that has the title and kind for any movie/show/game with “harvard” or “harvard-university” as a keyword.

Notes:

1. The `dplyr` function `distinct()` might be helpful for removing duplicate rows.
2. Although it is called the “Internet *Movie* Database”, it contains information on 7 types of shows/games.

```
# Dataset that contains the id for all movie types on IMDB
kinds <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/kinds

# Dataset that contains the keyword, movie id and kind id for
# movies that had Harvard or MIT keywords
movie_ids <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/mo
```

```r
# Dataset that contains the title and movie id for all movies
# that had Harvard keywords
harvard_titles <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/da
```

b. What proportion of titles with Harvard keywords are actually movies?

c. I *starred* in a movie in my youth but IMDB thinks I was in 3 movies. The movie I was in was directed by my high school friend, Scott Beck. Use the appropriate join on the following datasets to determine the name of the movie I was actually in.

```r
# Dataset of all movies associated with me on IMDB
kelly_mcconville <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/


# Dataset of all movies associated with Scott Beck on IMDB (from 2018 or earlier)
scott_beck <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/s
```

Problem 4

In this problem, we want you to wrangle/clean up some data and then compare your "cleaned data" with a peer to see how your final versions vary. You will use the following three datasets which contain information on the trees from a few parks in Portland, OR.

```r
# Data on trees in a few parks in Portland
treez <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/treez
treez_park <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/t
treez_loc <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/tr
```

a. Join `treez`, `treez_park`, and `treez_loc` to create one data frame where:

- Each row represents one tree (and there are no duplicates) from the following parks: Mt Tabor Park, Laurelhurst Park, Columbia Park
- All missing values (including suspicious values) are appropriately coded as `NA`.
- Each variable is stored with the most appropriate `class`.
- Categories of categorical variables are appropriated encoded.
- And, any other relevant cleaning is performed.

Note: It might take a little sleuthing to figure out which variables are your keys and what makes these datasets messy.

b. Export your dataset to a csv file using `write_csv()`.

```r
# We recommend leaving in eval = FALSE
write_csv(name_of_dataset, file = "your_file_name.csv")
```

c. Find a classmate and share your cleaned datasets with each other. Import their data below and also provide their name. (Feel free to share your data with multiple people but you only need to load one classmate's dataset.)

```r
# Import their dataset
```

d. Compare your dataset and their dataset then answer the following questions:

- Do your datasets have the same number of rows? Same number of columns?

- Use `setequal()` to determine if they are exactly the same.
- How are they different?

e. A goal of this exercise is to experience both the **subjectivity** and **iterative nature** of data cleaning. Any time we clean data, we are making choices and often we don't catch all the bugs in our data the first (or second) time around.

Based on your exploration of a classmate's cleaned dataset, do you think your dataset needs further wrangling? If not, explain your reasoning. If so, do that additional wrangling now.

f. Now that you are an expert on these data, make a data visualization that tells a story (provide appropriate context on the graph) and then briefly explain the story shown in the graph.

Problem 5

We are surprised at how many of the FiveThirtyEight datasets are "untidy". For this problem, you will explore the following untidy dataset from the article "What Do Men Think It Means To Be A Man?". Specifically, these data are responses to the question "Do you think that society puts pressure on men in a way that is unhealthy or bad for them?"

```r
masculine_data <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/da
```

a. How is `masculine_data` not currently in a tidy format? In your answer, make sure to reference one of the rules of tidy data.

b. Why might someone store data in this untidy format?

c. Create a tidy version of `masculine_data`. While we realize you could do this manually with `data.frame()` because the dataset is so small, please use `tidyr/dplyr` in your answer.

```r
# Hint: Reload the data but include a useful argument in `read_csv()` to ignore some of the data
```

d. Use your transformed `data.frame` to create a segmented bar graph, similar to the one in the article.