Problem Set 2

Data Wrangling

[YOUR NAME]

Due Date: 2024-07-09

Getting Set Up

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...`. Accept defaults and save this file as `[YOUR NAME]_ps2.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[YOUR NAME]_ps2.Rmd` file. Then change the `author: [Your Name]` on line 2 to your name.

We will be using two different files. First is the `MI2020_ExitPoll.rds` data from the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/data/MI2020_ExitPoll.rds). Second is the `nba_players_2018.Rds` data, which is also on the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/data/nba_players_2018.Rds)

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is worth 10 total points, plus **two** extra credit questions, each worth **two** points. The point values for each question are indicated in brackets below. To receive full credit, you must have the correct code. In addition, some questions ask you to provide a written response in addition to the code.

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, email the knitted output to Eun Ji Kim (kej990804@snu.ac.kr (mailto:kej990804@snu.ac.kr)) **as a PDF** by the start of class on Tuesday, July 9th. If you need help converting to a PDF, see this tutorial (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Psets/ISP_pset_0_HELPER.pdf).

Good luck!

*Copy the link to ChatGPT you used here: _____

Question 0

Require `tidyverse` and load the `MI2020_ExitPoll.Rds` data to an object called `MI_raw`. ALSO load a new package called `labelled`, which will allow us to read the labels for our variables. Remember, if you don't have this package yet, you need to use `install.packages("labelled")` in the `Console` window.

```
require()
```

```
## Loading required package:
```

```
require()
```

```
## Loading required package:
```

```
MI_raw <- read_rds("")
```

```
## Error in read_rds(""): could not find function "read_rds"
```

Question 1 [1 point]

How many units of observation are there in the raw dataset? How many variables are there?

> Write answer here.

Now create a new object called `MI_clean` *that contains only the following variables:*

- AGE10
- SEX
- PARTYID
- EDUC18
- PRSMI20
- QLT20
- LGBT
- BRNAGAIN
- LATINOS
- QRACEAI
- WEIGHT

```
MI_clean <- MI_raw %>%
  select() # Insert variables here
```

```
## Error in MI_raw %>% select(): could not find function "%>%"
```

How many units of observation are there in the new dataset? How many variables are there?

> Write answer here.

Question 2 [1 point]

How many **unit non-responses** *are there in the* `PRSMI20` *variable? What is the numeric code for unit non-response in this data? Remember, unit non-response refers to people who were asked a question but refused to answer.*

```
MI_clean %>%
  count() # Insert variable here
```

```
## Error in MI_clean %>% count(): could not find function "%>%"
```

Write answer here.

Question 3 [1 point]

Let's create a new variable called `preschoice` *that converts* `PRSMI20` *to a simpler categorical variable that indicates whether the respondent voted for Trump, Biden, or neither. To do this, use* `mutate()` *and* `ifelse()` *to replace the numeric values of* `PRSMI20` *with their text labels. Remember to replace the unit non-response code with* `NA`*.*

```
MI_clean <- MI_clean %>%
  mutate(preschoice = ifelse(, # Test for Biden
                             , # If TRUE, type "Biden"
                             ifelse(, # Test for Trump if it is not Biden
                                    , # If TRUE, type "Trump"
                                    ifelse(, # Test for unit non-response if it is neith
er Biden nor Trump
                                           , # If TRUE, type NA
                                           )))) # Otherwise, type "Neither"
```

```
## Error in MI_clean %>% mutate(preschoice = ifelse(, , ifelse(, , ifelse(, : could not
find function "%>%"
```

```
# Use this code to confirm you did it correctly
MI_clean %>%
  count(PRSMI20,preschoice)
```

```
## Error in MI_clean %>% count(PRSMI20, preschoice): could not find function "%>%"
```

Now `count()` *the number of respondents who reported voting for each candidate using the* `preschoice` *variable. How many respondents didn't vote for either Biden or Trump in 2020?*

```
MI_clean %>%
  count() # Insert variable here
```

```
## Error in MI_clean %>% count(): could not find function "%>%"
```

Write answer here.

Question 4 [1 point]

Now do the same for the `SEX` variable and the `LGBT` variable. Name the the text version `Gender` for `SEX` and `Lgbt_clean` for `LGBT`. Remember to replace the unit non-response code with `NA`, and be aware that different variables use different codes!

```
# Follow these steps for each variable: SEX
# 1. count() to see the unit non-response code (if it exists)
MI_clean %>%
  count() # Insert variable here
```

```
## Error in MI_clean %>% count(): could not find function "%>%"
```

```
# 2. Create the cleaned variable
MI_clean <- MI_clean %>%
  mutate(Gender = ifelse(,  # Test for value 1
                         ,  # If TRUE, type ...
                         )) # If FALSE, type ...
```

```
## Error in MI_clean %>% mutate(Gender = ifelse(, , )): could not find function "%>%"
```

```
# Follow these steps for each variable: LGBT
# 1. count() to see the unit non-response code (if it exists)
MI_clean %>%
  count() # Insert variable here
```

```
## Error in MI_clean %>% count(): could not find function "%>%"
```

```
# 2. Create the cleaned variable
MI_clean <- MI_clean %>%
  mutate(Lgbt_clean = ifelse(, # Test for value 1
                             , # If TRUE, type ...
                             ifelse(, # If FALSE, test for unit non-response
                                    , # If TRUE, type NA
                                    ))) # Otherwise, type ...
```

```
## Error in MI_clean %>% mutate(Lgbt_clean = ifelse(, , ifelse(, , ))): could not find f
unction "%>%"
```

Question 5 [1 point]

What proportion of women supported Trump? What proportion of LGBTQ-identifying respondents supported Trump?

```
# Apply the following for each variable
# Method 1: more lines of code
MI_clean %>%
  count() %>% # Count by Gender and preschoice
  group_by() %>% # Group by Gender
  mutate(totn = ) %>% # Calculate the total respondents by gender
  mutate(proportion = ) # Calculate the proportion
```

```
## Error in MI_clean %>% count() %>% group_by() %>% mutate(totn = ) %>% mutate(proportio
n = ): could not find function "%>%"
```

```
# Method 2: prop.table()
MI_clean %>%
  count() %>% # Count by Gender and preschoice
  group_by() %>% # Group by Gender
  mutate(proportion = ) # Calculate the proportion using prop.table()
```

```
## Error in MI_clean %>% count() %>% group_by() %>% mutate(proportion = ): could not fin
d function "%>%"
```

> Write answer here.

Extra Credit 1 [2 points]

Calculate the proportion of women who supported Trump by age-group to determine which age-group had the highest Trump support among women. You will need to clean the AGE10 variable before completing this problem, just like we did with the PRSMI20, SEX, and LGBT variables. Call the new variable "Age". HINT: to make your life easier (and not write a 10-level nested ifelse() function), try asking ChatGPT for help with this prompt: "I have a labelled variable in R that I want to convert to text. How can I do this?"

```
# INSERT CODE HERE
```

> Write answer here.

Question 6 [1 point]

Now let's load a different dataset to practice univariate visualization. Open `nba_players_2018.Rds` *from the github page and save it to a new object called* `nba` *. This dataset contains information on basketball players in the NBA from the 2018-2019 season. The codebook for it can be found in homework 3, which is also on github (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Homeworks/ISP_hw_3.pdf).*

```
nba <- read_rds("") # Insert link here
```

```
## Error in read_rds(""): could not find function "read_rds"
```

We are interested in the relationship between the player's age (`agePlayer`) and the amount of points they score (`pts`). Please answer the following research question and provide a theory supporting your answer: "Do older NBA players score more points than younger players?"

> Write answer here.

Question 7 [2 points]

Based on your answer above, what is the outcome / dependent / Y variable and what is the explanatory / independent / X variable? Why?

> Write answer here.

Create a univariate visualization of both the X and Y variables. Choose the best `geom_...()` based on the variable type, and make sure to label your plots!

```
# X variable
nba %>%
  ggplot(aes(x = ...)) + # Put the X variable on the x-axis
  geom_...() + # Choose the best visualization given the type of variable
  labs(x = "", # Provide helpful labels
       y = "",
       title = "")
```

```
## Error in nba %>% ggplot(aes(x = ...)): could not find function "%>%"
```

```
# Y variable
nba %>%
  ggplot(aes(x = ...)) + # Put the Y variable on the x-axis
  geom_...() + # Choose the best visualization given the type of variable
  labs(x = "", # Provide helpful labels
       y = "",
       title = "")
```

```
## Error in nba %>% ggplot(aes(x = ...)): could not find function "%>%"
```

Question 8 [2 points]

Now analyze the data by creating a multivariate visualization that shows the relationship between age and points.

```
nba %>%
  ggplot(aes(x = ...,    # Put the X variable on the x-axis
             y = ...)) + # Put the Y variable on the y-axis
  geom_...() + # Choose the best visualization given the type of variable
  labs(x = "", # Provide helpful labels
       y = "",
       title = "")
```

```
## Error in nba %>% ggplot(aes(x = ..., y = ...)): could not find function "%>%"
```

Based on your analysis, does the data support or reject your hypothesis from Question 6?

> Write answer here.

Extra Credit 2 [2 points]

Let's look for evidence of a "curvelinear" relationship between player age and points scored. To do so, first calculate the average points scored by age. Then plot this relationship using a multivariate visualization. Add a line of best fit with `geom_smooth()` *but DON'T use* `method = "lm"` *. What do you conclude? Why?*

```
# Insert code here.
```

> Write answer here.