Problem Set 2

Data Wrangling

[YOUR NAME]

Due Date: 2023-02-03

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 `[LAST NAME]_ps2.Rmd` to your `code` folder.

Copy and paste the contents of this file into your `[LAST NAME]_ps2.Rmd` file. Then change the `author: [YOUR NAME]` (line 4) to your name.

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 three extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must both have the correct code **and include a comment describing what each line does**. 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, compiled the completed problem set and upload the PDF file to Brightspace by midnight on 2023/02/03.

Good luck!

Question 0

Require `tidyverse` and load the `MI2020_ExitPoll.Rds` data to an object called `MI_raw` . (Tip: use the `read_rds()` function with the link to the raw data.)

```
require(tidyverse)
```

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

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
MI_raw <- read_rds('../data/MI2020_ExitPoll.rds') #https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/3_Data_Wrangling/data/MI2020_ExitPoll.rds?raw=true')
```

Question 1 [1 point]

How many voters were from Wayne County?

```
MI_raw %>%
  count(County) %>% # Count the number of respondents per county
  filter(County == 'WAYNE') # Subset to Wayne county (note capitalization)
```

```
## # A tibble: 1 × 2
##   County     n
##   <chr>  <int>
## 1 WAYNE    102
```

> There were 102 voters from Wayne County.

Question 2 [1 point]

Who did the majority of surveyed voters support in the 2020 presidential election?

```
MI_raw %>%
  count(PRSMI20) %>% # Count the number of respondents who supported each candidate
  mutate(share = n / sum(n)) %>% # Calculate this number as the proportion of all respondents
  arrange(desc(share)) # Arrange in decsending order
```

```
## # A tibble: 6 × 3
##   PRSMI20                                        n   share
##   <hvn_lbl_>                                 <int>   <dbl>
## 1 1 [Joe Biden, the Democrat]                  723 0.587
## 2 2 [Donald Trump, the Republican]             459 0.373
## 3 9 [Another candidate]                         25 0.0203
## 4 8 [Refused]                                   14 0.0114
## 5 0 (NA) [Will/Did not vote for president]       6 0.00487
## 6 7 [Undecided/Don’t know]                       4 0.00325
```

> The majority of surveyed voters (58.7%) supported Joe Biden in the 2020 presidential election.

Question 3 [1 point + 1 EC]

What proportion of women supported Trump? What proportion of men supported Biden? EC: Answer using `group_by()`.

```
MI_raw %>%
  filter(SEX == 2) %>% # Subset to women (check the numeric code!)
  count(PRSMI20) %>% # Count the number of women who supported each candidate
  mutate(share = n / sum(n)) %>% # Calculate the share of women who supported each candidate
  filter(PRSMI20 == 2) # Subset to those who supported Trump (check the numeric code!)
```

```
## # A tibble: 1 × 3
##   PRSMI20                             n share
##   <hvn_lbl_>                      <int> <dbl>
## 1 2 [Donald Trump, the Republican]   212 0.325
```

```
MI_raw %>%
  filter(SEX == 1) %>% # Subset to men (check the numeric code!)
  count(PRSMI20) %>% # Count the number of men who supported each candidate
  mutate(share = n / sum(n)) %>% # Calculate the share of men who supported each candidate
  filter(PRSMI20 == 1) # Subset to those who supported Biden (check the numeric code!)
```

```
## # A tibble: 1 × 3
##   PRSMI20                         n share
##   <hvn_lbl_>                  <int> <dbl>
## 1 1 [Joe Biden, the Democrat]   304 0.525
```

```
# Extra Credit: Alternative approach using group_by() and summarise()
MI_raw %>%
  group_by(SEX) %>%
  summarise(pctTrump = mean(PRSMI20 == 2),
            pctBiden = mean(PRSMI20 == 1))
```

```
## # A tibble: 2 × 3
##   SEX        pctTrump pctBiden
##   <hvn_lbl_>    <dbl>    <dbl>
## 1 1 [Male]      0.427    0.525
## 2 2 [Female]    0.325    0.643
```

32.5% of women supported Trump. 52.5% of men supported Biden.

Question 4 [1 point]

Create a new object called `MI_clean` that contains only the following variables: - AGE10 - SEX - PARTYID - EDUC18 - PRMSI20 - QLT20 - LGBT - BRNAGAIN - LATINOS - QRACEAI - WEIGHT and then list which of these variables contain missing data recorded as `NA`. How many respondents were not asked certain questions?

```
MI_clean <- MI_raw %>%
  select(AGE10,SEX,PARTYID,EDUC18,PRSMI20,QLT20,LGBT,BRNAGAIN,LATINOS,QRACEAI,WEIGHT) # Select t
he requested variables

summary(MI_clean) # Identify which have missing data recorded as NA
```

```
##      AGE10             SEX          PARTYID          EDUC18         PRSMI20
##  Min.   : 1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :0.00
##  1st Qu.: 6.000   1st Qu.:1.00   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.00
##  Median : 8.000   Median :2.00   Median :2.000   Median :3.000   Median :1.00
##  Mean   : 8.476   Mean   :1.53   Mean   :2.236   Mean   :3.288   Mean   :1.63
##  3rd Qu.: 9.000   3rd Qu.:2.00   3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:2.00
##  Max.   :99.000   Max.   :2.00   Max.   :9.000   Max.   :9.000   Max.   :9.00
##
##      QLT20            LGBT          BRNAGAIN        LATINOS
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000
##  Median :3.000   Median :2.000   Median :2.000   Median :2.000
##  Mean   :2.956   Mean   :2.224   Mean   :1.907   Mean   :2.175
##  3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
##  NA's   :616     NA's   :615     NA's   :615
##     QRACEAI          WEIGHT
##  Min.   :1.000   Min.   :0.1003
##  1st Qu.:1.000   1st Qu.:0.3775
##  Median :1.000   Median :0.8020
##  Mean   :1.572   Mean   :1.0000
##  3rd Qu.:1.000   3rd Qu.:1.4498
##  Max.   :9.000   Max.   :5.0853
##
```

`QLT20`, `LGBT`, and `BRNAGAIN` have missing values stored as `NA`. 616 respondents were not asked `QLT20`, and 615 were not asked either `LGBT` or `BRNAGAIN`.

Question 5 [1 point]

Are there **unit non-response** data in the `AGE10` variable? If so, how are they recorded? What about the `PARTYID` variable?

```
MI_clean %>%
  count(AGE10)
```

```
## # A tibble: 11 × 2
##    AGE10                          n
##    <hvn_lbl_>                <int>
##  1  1 [18 and 24,]               33
##  2  2 [25 and 29,]               28
##  3  3 [30 and 34,]               42
##  4  4 [35 and 39,]               46
##  5  5 [40 and 44,]               78
##  6  6 [45 and 49,]               83
##  7  7 [50 and 59,]              274
##  8  8 [60 and 64,]              143
##  9  9 [65 and 74,]              290
## 10 10 [75 or over?]             199
## 11 99 [[DON'T READ] Refused]    15
```

```
MI_clean %>%
  count(PARTYID)
```

```
## # A tibble: 5 × 2
##   PARTYID                                n
##   <hvn_lbl_>                        <int>
## 1 1 [Democrat]                        425
## 2 2 [Republican]                      280
## 3 3 [Independent]                     416
## 4 4 [Something else]                   94
## 5 9 [[DON'T READ] Don’t know/refused]    16
```

The unit non-response data in the `AGE10` variable is recorded with the number `99`. Missing data in the `PARTYID` variable is recorded with the number `9`.

Question 6 [1 point]

Let’s create a new variable called `preschoice` that converts `PRSMI20` to a character. To do this, install the `haven` package if you haven’t already, then use the `as_factor()` function from the `haven` package combined with the `as.character()` function from base `R`. Now `count()` the number of respondents who reported voting for each candidate. Do you get the same number as in Question 2?

```
MI_clean <- MI_clean %>%
  mutate(preschoice = as.character(haven::as_factor(PRSMI20)))

MI_clean %>%
  count(preschoice)
```

```
## # A tibble: 6 × 2
##   preschoice                          n
##   <chr>                           <int>
## 1 Another candidate                  25
## 2 Donald Trump, the Republican      459
## 3 Joe Biden, the Democrat           723
## 4 Refused                            14
## 5 Undecided/Don't know                4
## 6 Will/Did not vote for president     6
```

Question 7 [1 point]

Now do the same for the `QLT20` variable, the `AGE10` variable, and the `LGBT` variable. For each variable, make the character version `Qlty` for `QLT20`, `Age` for `AGE10`, and `Lgbt_clean` for `LGBT`.

```
# QLT20
MI_clean <- MI_clean %>%
  mutate(Qlty = as.character(haven::as_factor(QLT20)),
         Age = as.character(haven::as_factor(AGE10)),
         Lgbt_clean = as.character(haven::as_factor(LGBT)))
```

Question 8 [1 point]

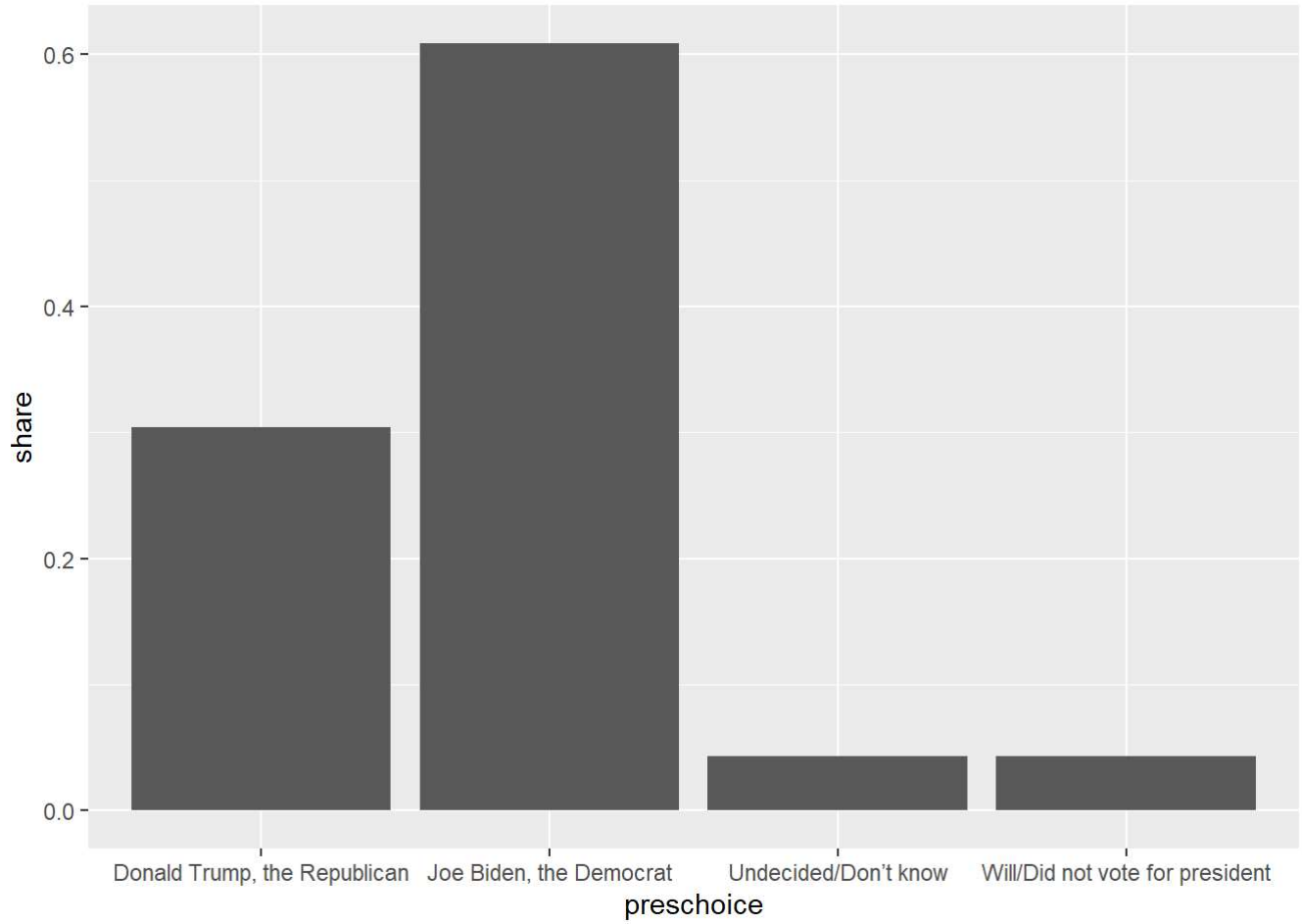
For each of these new variables, replace the **unit non-response** label with `NA`. EC: use a `grepl()` function with an `ifelse()` statement for more efficient code.

```
MI_clean <- MI_clean %>%
  mutate(Qlty = ifelse(grepl("DON'T READ",Qlty),NA,Qlty),
         Lgbt_clean = ifelse(grepl("DON'T READ",Lgbt_clean),NA,Lgbt_clean),
         Age = ifelse(grepl("DON'T READ",Age),NA,Age))
```

Question 9 [1 point + 1 EC]

What proportion of LGBT-identifying voters supported Trump? EC: Plot this answer.

```
MI_clean %>%
  filter(Lgbt_clean == 'Yes') %>%
  count(preschoice) %>%
  mutate(share = n / sum(n)) %>%
  ggplot(aes(x = preschoice,y = share)) +
  geom_bar(stat = 'identity')
```

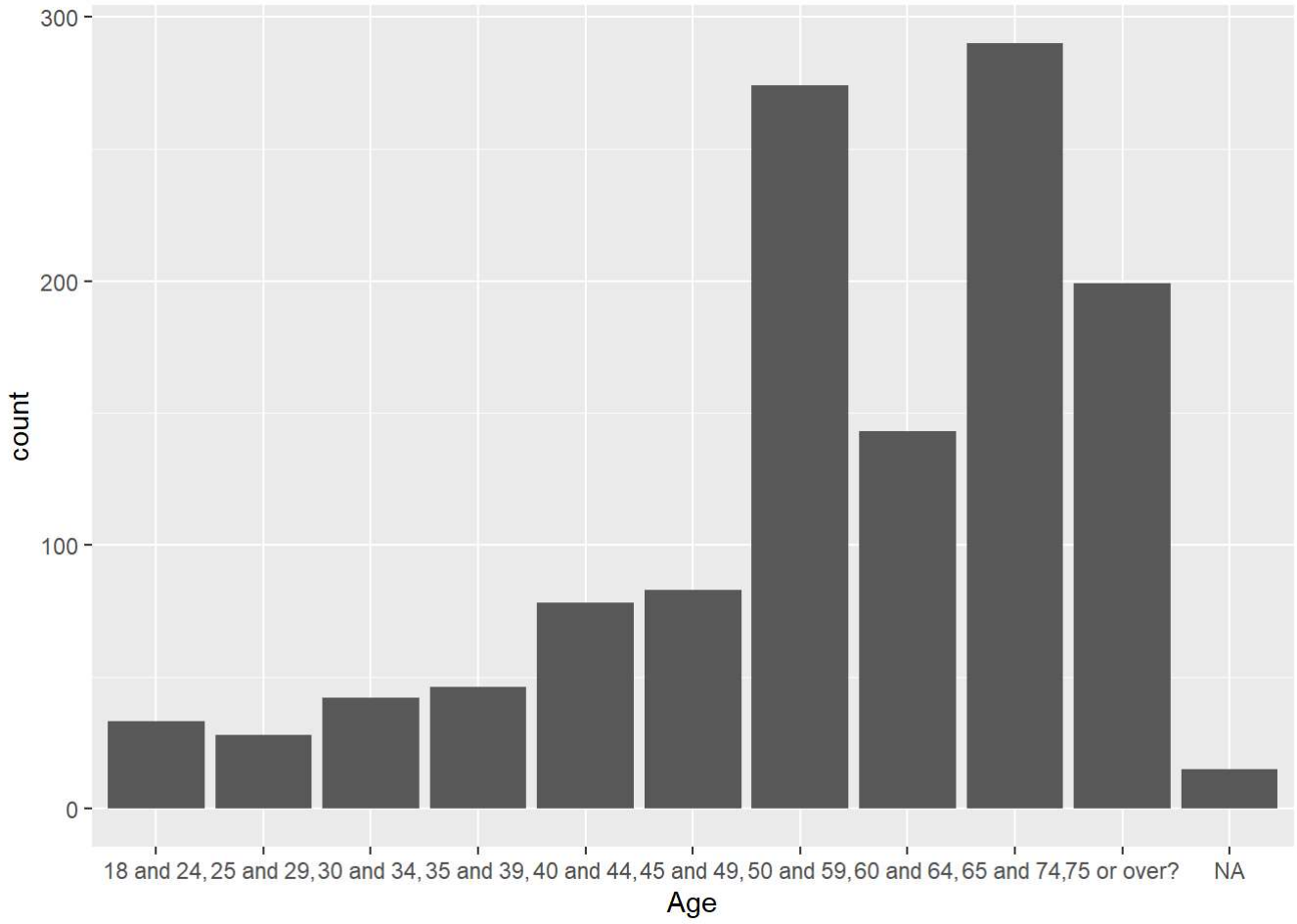


30.4% of LGBT-identifying voters supported Trump.

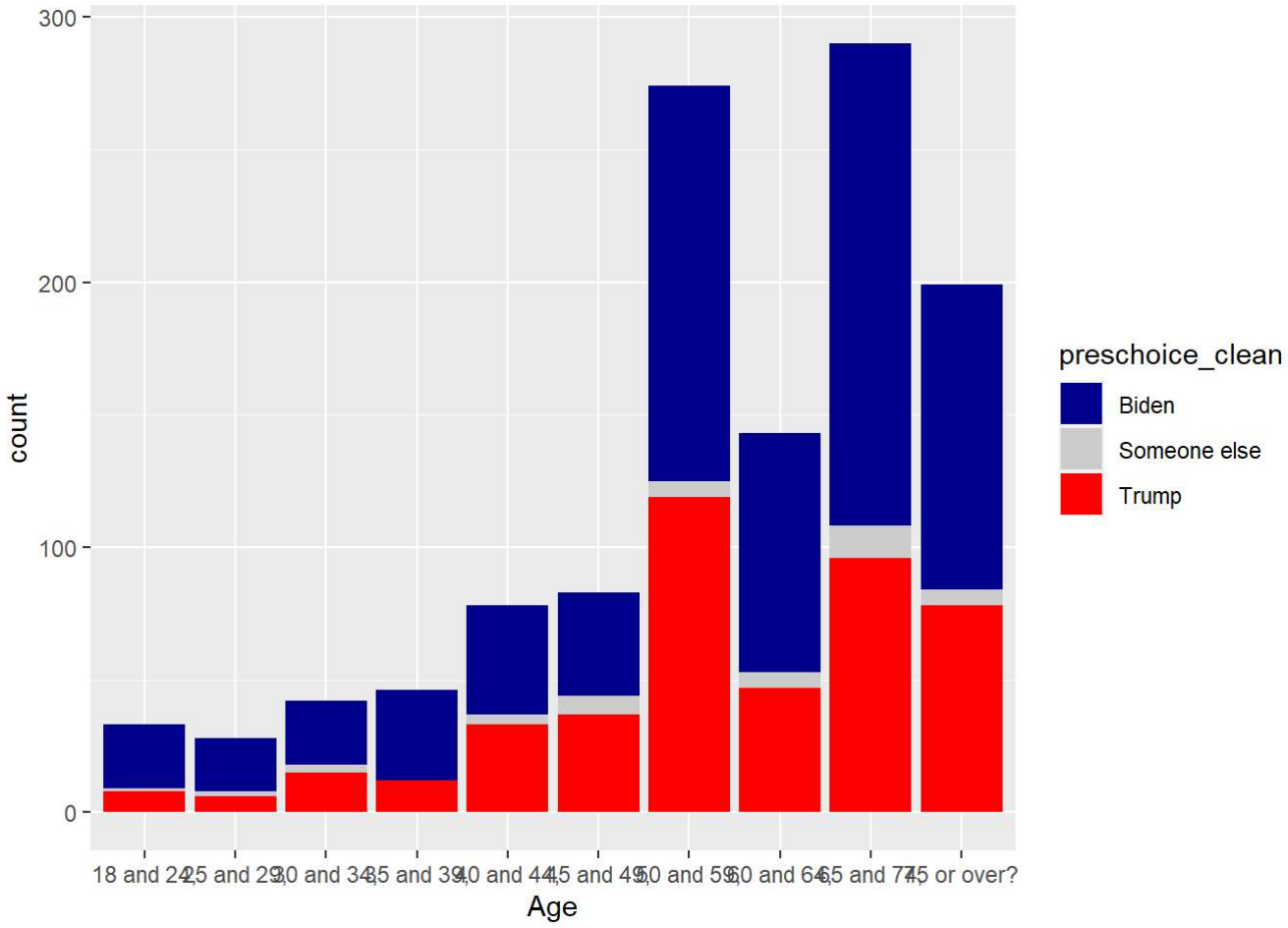
Question 10 [1 point + 1 EC]

Plot the distribution of ages in the data. EXTRA CREDIT: color by the number of voters in each bracket that supported Trump, Biden, or someone else. Make sure to drop voters who didn't indicate who they voted for **AND** those who didn't indicate their age.

```
# Simple
MI_clean %>%
  ggplot(aes(x = Age)) +
  geom_bar(stat = 'count')
```




```
# Extra Credit
MI_clean %>%
  mutate(preschoice_clean = ifelse(grepl('Biden',preschoice),'Biden',
                                   ifelse(grepl('Trump',preschoice),'Trump','Someone else'))) %
>%
  filter(!is.na(preschoice),
         !is.na(Age)) %>%
  ggplot(aes(x = Age,fill = preschoice_clean)) +
  geom_bar(stat = 'count') +
  scale_fill_manual(values = c('Biden' = 'darkblue','Someone else' = 'grey80','Trump' = 'red'))
```


300
200
100
0
count
18 and 24
25 and 29
30 and 34
35 and 39
40 and 44
45 and 49
50 and 59
60 and 64
65 and 74
75 or over?
Age
preschoice_clean
Biden
Someone else
Trump