Milestone 02

FirstName LastName

Environment

Update the YAML with your first and last name.

Load the packages.

```
library(here)
library(tidyverse)
library(haven)
library(gssr)
library(gssrdoc)
library(summarytools)
```

Load the 2024 GSS 2024 data and the GSS panel data

```
# load the gss 2024 data (add your code below)
gss24 <- gss_get_yr(2024)

# Use here() to construct the file path of the panel data
gss_panel.dta <- here("data", "GSS_2020_panel_stata_1a/gss2020panel_r1a.dta")

#load the panel data using `haven::read_dta()`
gss_panel <- read_dta(gss_panel.dta)
```

You'll be working with the following GSS variables:

- `hrs1`: Hours worked per week
- `tvhours`: Hours spent watching TV per day
- `sex`: Respondent's gender
- `polviews`: Political views (from extremely liberal to extremely conservative)

Code, Output, Meaning

Use R to complete the checkpoints below. Show your work (e.g., R code chunks) where appropriate. Add narrative (text outside code chunks) or comments (text inside the code chunks) throughout.

Reference specific statistics (where appropriate) from your output to justify your answers. Explain what the values tell you about the data; interpret their meaning in relation to the question.

Use the 2024 GSS data for checkpoints 01-05.

Use the panel data for checkpoints 06-10.

Checkpoint 01: Select relevant variables

Use `select()` to create a new data frame with only the four variables listed above.

```
my_data <- gss24 |>
  select(hrs1, tvhours, sex, polviews)
```

Checkpoint 02: Clean and create variables

A. Use `zap_missing`, `as_factor`, and `droplevels` to make `sex` into a factor variable.

```
my_data$sex <- zap_missing(my_data$sex)
my_data$sex <- as_factor(my_data$sex)
my_data$sex <- droplevels(my_data$sex)
```

B. Use `mutate()` and `case_when()` to create a new variable called `work_category` that groups respondents as follows:

- "Not working" if hrs1 == 0
- "Part-time" if hrs1 is between 1 and 34
- "Full-time" if hrs1 is 35 or more

```
my_data <- my_data %>%
  mutate(
    work_category = case_when(
      hrs1 == 0 ~ "Not working",
      hrs1 >= 1 & hrs1 <= 34 ~ "Part-time",
      hrs1 >= 35 ~ "Full-time"
  ))
```

C. Use `mutate()` and `case_when()` to create a new variable called `pol3cat` that groups respondents as follows:

- "Liberal" if polviews equals "extremely liberal", "liberal", or "slightly liberal"
- "Moderate" if polviews equals "moderate, middle of the road"
- "Conservative" if polviews equals "extremely conservative", "conservative", "slightly conservative"

```
my_data <- my_data %>%
  mutate(
    pol3cat = case_when(
      polviews >= 1 & polviews <= 3 ~ "Liberal",
      polviews == 4 ~ "Moderate",
      polviews >= 5 & polviews <= 7 ~ "Conservative"
  ))
```

Checkpoint 03: Remove rows with missing data

Use `drop_na()` to keep only respondents who have non-missing values for all variables.

```
my_data <- my_data %>%
  drop_na()
```

Checkpoint 04: Summarize work and tv hours by political identity

A. Use `group_by()` and `sumamrize()` to create a table showing the frequency, mean, median, and sd of **`hrs1`** for each of the three political identity groups.

```
my_data |>
  group_by(pol3cat) |>
  summarise(
    count = n(),
    mean_hrs1 = round(mean(hrs1), digits = 2),
    median_hrs1 = round(median(hrs1)),
    sd_hrs1 = round(sd(hrs1), digits = 2)
    )
```

```
# A tibble: 3 x 5
  pol3cat      count mean_hrs1 median_hrs1 sd_hrs1
  <chr>        <int>     <dbl>       <dbl>   <dbl>
```

```
1 Conservative   368      39.9          40    14.7
2 Liberal        350      38.4          40    13.4
3 Moderate       414      39.4          40    13.8
```

Which political group reported the most work hours in 2024? Was there a lot or a little variability in work hours?

Is the median value for `tvhours` consistent with the mean? Speculate what might explain any differences.

B. Use `group_by()` and `sumamrize()` to create a table showing the frequency, mean, median, and sd of **`tvhours`** for each of the three political identity groups.

```
my_data |>
  group_by(pol3cat) |>
  summarise(
    count = n(),
    mean_tv = round(mean(tvhours), digits = 2),
    median_tv = round(median(tvhours)),
    sd_tv = round(sd(tvhours), digits = 2)
    )
```

```
# A tibble: 3 x 5
  pol3cat      count mean_tv median_tv sd_tv
  <chr>        <int>   <dbl>     <dbl> <dbl>
1 Conservative   368    2.56         2  2.71
2 Liberal        350    2.47         2  2.3
3 Moderate       414    2.75         2  2.49
```

Which political group watched the most television in 2024? Was there a lot of a little variability in television hours?

Checkpoint 05: Create a summary dataframe

Use `group_by()` and `summarise()` to create and save a dataframe showing the average work and tv hours for men and women in 2024.

```r
gss24_summary <- my_data |>
  group_by(sex) |>
  summarise(
    avg_work_24 = round(mean(hrs1), digits = 2),
    avg_tv_24 = round(mean(tvhours), digits = 2)
    )

head(gss24_summary)
```

```
# A tibble: 2 x 3
  sex    avg_work_24 avg_tv_24
  <fct>        <dbl>     <dbl>
1 male          41.7      2.53
2 female        37.0      2.67
```

Did men or women report more average work hours in 2024? What about TV hours?

What percentage of men worked 40 hours or less than per week in 2024? (You can either add `R` code chunk to this document or calculate it with the formula. Show your work either way.)

Checkpoint 06: Code for within-person change

Run the following code to produce the average within-person change between 2018 and 2020.

It shows the average change from 2018 to 2020 in work and TV hours for each gender.

Positive values mean an increase from 2018 to 2020; negative values mean a decrease.

```r
gss_panel %>%
  select(sex_2, hrs1_2, hrs1_1b, tvhours_2, tvhours_1b) |>
  drop_na(sex_2) |>
  mutate(
    work_change = hrs1_2 - hrs1_1b,
    tv_change = tvhours_2 - tvhours_1b,
  ) %>%
  group_by(as_factor(sex_2)) %>%
  summarise(
    avg_work_change = round(mean(work_change, na.rm = TRUE), digits = 2),
    sd_work_change = round(sd(work_change, na.rm = TRUE), digits = 2),
    avg_tv_change = round(mean(tv_change, na.rm = TRUE), digits = 2),
```

```
    sd_tv_change = round(sd(tv_change, na.rm = TRUE), digits = 2)
  )
```

```
# A tibble: 2 x 5
  `as_factor(sex_2)` avg_work_change sd_work_change avg_tv_change sd_tv_change
  <fct>                        <dbl>          <dbl>         <dbl>        <dbl>
1 male                         -2.11           15.0          0.67         2.69
2 female                       -0.31           14.0          0.55         2.74
```

Did the number of work hours increase or decrease from 2018 to 2020 for men? What about for women?

Did the number of hours spent watching TV increase or decrease from 2018 to 2020 for men? What about for women?

Was there a gender difference in the average within-person change between 2018 and 2020 for work hours? What about for tv hours?

Checkpoint 07: Reshape panel data so it is tidy

Use `select()` and `pivot_longer()` to create tidy data.

Include `relocate()` to put the variables in a logical order and `drop_na()` to remove rows with any missing data.

```
my_gss_panel <- gss_panel |>
  select(yearid, sex_2,
         starts_with("hrs1_"),
         starts_with("tvhours_")) |>
    pivot_longer(
        cols = c( -yearid, -sex_2),
        names_to = c(".value", "panel"),
        names_sep = "_") |>
  relocate(panel, .after = yearid) |>
  drop_na()

head(my_gss_panel)
```

```
# A tibble: 6 x 5
    yearid panel sex_2      hrs1      tvhours
     <dbl> <chr> <dbl+lbl>  <dbl+lbl> <dbl+lbl>
```

```
1 20160001 1a    1 [male]   50        1
2 20160001 2     1 [male]   45        2
3 20160002 1a    1 [male]   42        1
4 20160002 2     1 [male]   20        4
5 20160004 1a    2 [female] 30        1
6 20160009 1a    1 [male]   80        2
```

Checkpoint 08: Recode panel variables

A. Use `mutate()` and `case_when()` to re-code the `sex_2` variable so that "1" = man, "2" = woman.

```
my_gss_panel <- my_gss_panel |>
  mutate(sex = case_when(
         sex_2 == 1 ~ "male",
         sex_2 == 2 ~ "female",
         TRUE ~ NA_character_))
```

B. Use `mutate()` and `case_when()` to re-code the `panel` variable so that "1a" = 2016, "1b" = 2018, and "2" = 2020.

```
my_gss_panel <- my_gss_panel |>
  mutate(panel = case_when(
         panel == "1a" ~ 2016,
         panel == "1b" ~ 2018,
         panel == "2" ~ 2020,
         TRUE ~ NA_integer_))

head(my_gss_panel)
```

```
# A tibble: 6 x 6
    yearid panel sex_2      hrs1      tvhours   sex
     <dbl> <dbl> <dbl+lbl>  <dbl+lbl> <dbl+lbl> <chr>
1 20160001  2016 1 [male]   50        1         male
2 20160001  2020 1 [male]   45        2         male
3 20160002  2016 1 [male]   42        1         male
4 20160002  2020 1 [male]   20        4         male
5 20160004  2016 2 [female] 30        1         female
6 20160009  2016 1 [male]   80        2         male
```

Checkpoint 09: Create a summary panel table

Use `group_by()` and `summarise()` to look at a table showing the average work hours for men and women for each panel year.

```
panel_summary <- my_gss_panel |>
  group_by(sex, panel) |>
    summarise(
    count = n(),
    avg_work = round(mean(hrs1), digits = 2),
    avg_tv = round(mean(tvhours), digits = 2)
    )
```

```
`summarise()` has grouped output by 'sex'. You can override using the `.groups`
argument.
```

```
panel_summary
```

```
# A tibble: 6 x 5
# Groups:   sex [2]
  sex    panel count avg_work avg_tv
  <chr>  <dbl> <int>    <dbl>  <dbl>
1 female  2016   159     37.6   2.03
2 female  2018   217     37.8   2.19
3 female  2020   325     38.6   3.02
4 male    2016   143     41.6   2.11
5 male    2018   166     44.7   2.3
6 male    2020   279     42.2   2.96
```

Do women differ in their average work and television hours from 2016, 2018, and 2020? What about men?

Checkpoint 10: Join and calculate the average panel differences by sex

Calculate the difference in average work and tv hours between 2016, 2018, 2020 and the 2024 average.

```r
## join the data
my_data_all <- full_join(gss24_summary, panel_summary, by = "sex")

## Create change variables
my_data_all <- my_data_all |>
  mutate(
    work_change = avg_work_24 - avg_work,
    tv_change = avg_tv_24 - avg_tv
    ) |>
  select(sex, panel, count,
         avg_work_24, avg_work, work_change,
         avg_tv_24, avg_tv, tv_change
         )

my_data_all
```

```
# A tibble: 6 x 9
  sex    panel count avg_work_24 avg_work work_change avg_tv_24 avg_tv tv_change
  <chr>  <dbl> <int>       <dbl>    <dbl>       <dbl>     <dbl>  <dbl>     <dbl>
1 male    2016   143        41.7     41.6       0.170      2.53   2.11      0.42
2 male    2018   166        41.7     44.7      -2.93       2.53   2.3       0.23
3 male    2020   279        41.7     42.2      -0.490      2.53   2.96     -0.43
4 female  2016   159        37.0     37.6      -0.690      2.67   2.03      0.64
5 female  2018   217        37.0     37.8      -0.840      2.67   2.19      0.48
6 female  2020   325        37.0     38.6      -1.70       2.67   3.02     -0.35
```

Speculate about whether there is evidence of a post-pandemic (2024) shift in work or TV time. Are the patterns the same or different for men and women?

IPUMS Data

To keep your Research Brief progress on track, you'll complete short exercises that correspond with the new course material using your own dataset as part of your milestones.

- Present a (pretty) relative frequency table of a nominal or ordinal variable in your dataset.
- Create a summary table for a key interval-ratio (e.g., continuous) variable you are interested in exploring more in your dataset. Include the frequency, mean, median, and mode.

Some potentially helpful resources:

- package `ipumsr()`
- Webinar: Using IPUMS data in R with ipumsr
- Just the slides: Using IPUMS data in R with ipumsr