Week 2, Class 3: Practice Exercises - ANSWER KEY

Working with Real Data

2024-12-31

1 Non-AI Exercises

1.1 1. Working with Missing Data

1.1.1 1.1 The Challenge of Missing Data

Why can't we use `== NA` to check for missing values in R? What does this tell us about how R thinks about missing data?

Answer: **We cannot use `== NA` because NA represents an unknown value, and comparing anything to an unknown value returns NA, not TRUE or FALSE.** This tells us that R treats missing data as truly "unknown" - it's not zero, it's not empty, it's simply unknown. When you ask "is this value equal to NA?", R responds with "I don't know" (which is NA). This is why we need the special function `is.na()` which is specifically designed to detect missing values.

1.1.2 1.2 Code Detective: Missing Data

Explain what's wrong with this code and how to fix it:

```
data %>%
  filter(age == NA)
```

Problem:The expression `age == NA` will always return NA, not TRUE or FALSE, so the filter won't work properly. It will return zero rows regardless of how many missing values exist. Use `is.na(age)` instead: `data %>% filter(is.na(age))` to keep rows with missing age values, or `data %>% filter(!is.na(age))` to remove them.

1.2 2. The Pipe Operator

1.2.1 2.1 Pipe vs No Pipe

These two pieces of code do the same thing. Which is easier to read and why?

```
# Version A
arrange(select(filter(congress, party_code == "Republican"),
bioname, nominate_dim1), nominate_dim1)

# Version B
congress %>%
  filter(party_code == "Republican") %>%
  select(bioname, nominate_dim1) %>%
  arrange(nominate_dim1)
```

Answer: **Version B is much easier to read because it follows a logical, left-to-right flow that matches how we think about data manipulation.** Version A requires reading from the inside out, making it hard to follow the sequence of operations. Version B reads like a recipe: "Take congress data, then filter for Republicans, then select these columns, then arrange by ideology." The pipe operator (`%>%`) transforms nested function calls into a clear, step-by-step process that's easier to write, read, and debug.

1.3 3. Understanding DW-NOMINATE Scores

1.3.1 3.1 The Scale

DW-NOMINATE scores range from -1 to +1. What do these values represent in terms of political ideology? If a member of Congress has a score of -0.8, what does this tell you about their political positions?

Answer: **DW-NOMINATE scores measure political ideology on a liberal-conservative spectrum, where -1 represents the most liberal position and +1 represents the most conservative position.** A score of -0.8 indicates a very liberal member of Congress - they are in the left wing of the political spectrum, likely voting consistently with progressive/liberal positions on most issues. This person would be among the most liberal members of Congress, comparable to representatives from very progressive districts.

1.3.2 3.2 Interpreting Changes

If the average DW-NOMINATE score for Democrats has moved from -0.3 in 1980 to -0.4 in 2020, what does this suggest about the party? What about if Republicans moved from +0.3 to +0.5?

Answer: **Democrats moving from -0.3 to -0.4 suggests the party has become more liberal over this 40-year period.** Republicans moving from +0.3 to +0.5 indicates they've become more conservative. **This pattern reveals increasing partisan polarization** - both parties have moved away from the center toward their respective ideological poles. This matches the common observation that American politics has become more polarized, with fewer moderate members in both parties and a widening ideological gap between them.

1.4 4. Advanced Tidyverse Operations

1.4.1 4.1 The %in% Operator

Explain what this code does and when you would use the `%in%` operator instead of `==`:

```
congress %>%
  filter(state_abbrev %in% c("CA", "TX", "NY", "FL"))
```

Answer: **This code filters the congress dataset to keep only rows where the state abbreviation is California, Texas, New York, or Florida.** The `%in%` operator checks if each value appears anywhere in the provided vector. You use `%in%` instead of `==` when you want to match against multiple values. With `==` you can only check one value at a time (e.g., `state_abbrev == "CA"`). To check multiple states with `==`, you'd need: `filter(state_abbrev == "CA" | state_abbrev == "TX" | state_abbrev == "NY" | state_abbrev == "FL")`, which is much more verbose and error-prone.

1.4.2 4.2 Select Helpers

Match each `select()` helper function with what it does:

a) `starts_with("vote")` 1. Selects columns containing "district" anywhere
b) `ends_with("_id")` 2. Selects all remaining columns
c) `contains("district")` 3. Selects columns starting with "vote"
d) `everything()` 4. Selects columns ending with "_id"

Matches: a = **3**, b = **4**, c = **1**, d = **2**

1.5 5. Building Complex Analyses

1.5.1 5.1 Step-by-Step Approach

Why is it important to build your analysis step by step rather than writing complex chains all at once? Give two specific benefits.

Answer:

1. **Easier debugging**: When you build step-by-step, you can check the output at each stage to ensure it's working correctly. If something breaks in a complex chain, it's hard to identify which step caused the problem.
2. **Better understanding**: Building incrementally helps you understand what each transformation does to your data. You can examine row counts, check for unexpected NAs, and verify that filters are working as intended before adding more complexity.

1.6 6. Data Exploration Best Practices

1.6.1 6.1 Understanding Your Data

Why is it important to use functions like `count()`, `summary()`, and `glimpse()` before doing complex analysis? What might happen if you skip this step?

Answer: **These exploration functions help you understand your data's structure, quality, and characteristics before analysis, preventing serious errors.**

If you skip exploration, you might:

- **Analyze the wrong data**: Not realize you have the wrong dataset or time period
- **Mishandle data types**: Treat numeric codes as actual numbers, or text numbers as characters
- **Miss data quality issues**: Not notice impossible values, outliers, or extensive missing data
- **Make incorrect assumptions**: Assume balanced groups when they're highly imbalanced
- **Choose wrong methods**: Use techniques inappropriate for your data's distribution

Example: Without checking, you might calculate average income not realizing that -999 is used to code missing values, completely distorting your results.

1.7 7. The NOT Operator

1.7.1 7.1 Using the NOT Operator

What will each of these expressions return if `x = 5` and `y = NA`:

a) `!TRUE` = **FALSE**
b) `!(x > 3)` = **FALSE** (since x > 3 is TRUE, and !TRUE is FALSE)
c) `!is.na(x)` = **TRUE** (since is.na(5) is FALSE, and !FALSE is TRUE)
d) `!is.na(y)` = **FALSE** (since is.na(NA) is TRUE, and !TRUE is FALSE)

1.7.2 7.2 Practical Application

Write the filter condition to keep only rows where the `vote_share` column is NOT missing:

Answer: `filter(!is.na(vote_share))`

1.8 8. Multiple Sorting

1.8.1 8.1 Understanding arrange() with Multiple Variables

What does this code do, and in what order will the results appear?

```
congress %>%
  arrange(state_abbrev, desc(nominate_dim1))
```

Answer: **This code sorts the congress data first alphabetically by state abbreviation (A to Z), then within each state, it sorts by DW-NOMINATE score from highest to lowest (most conservative to most liberal).**

2 AI Exercises

For each AI exercise:

- Write your prompt in the designated space
- Record Claude's response
- Run the code and document the results

2.1 9. Congressional Ideology Analysis

Dataset: HSall_members.csv

Description: DW-NOMINATE scores for all members of Congress throughout history.

Variables:

- `congress`: Congress number (int)
- `chamber`: House or Senate (chr)
- `state_abbrev`: State abbreviation (chr)
- `party_code`: Political party (chr)
- `bioname`: Member name (chr)
- `nominate_dim1`: Liberal-conservative score, -1 to 1 (dbl)
- `nominate_dim2`: Second dimension score (dbl)
- `nominate_geo_mean_probability`: Prediction accuracy (dbl)

2.1.1 9.1 Loading and Initial Exploration

```r
# Load the dataset
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
congress <- read_csv("HSall_members.csv")
```

```
Rows: 51044 Columns: 22
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (5): chamber, state_abbrev, party_code, bioname, bioguide_id
dbl (16): congress, icpsr, state_icpsr, district_code, occupancy, last_means...
lgl  (1): conditional
```

```
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Explore the DW-NOMINATE data
glimpse(congress)
```

```
Rows: 51,044
Columns: 22
$ congress                     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ chamber                      <chr> "President", "House", "House", "House", ~
$ icpsr                        <dbl> 99869, 379, 4854, 6071, 1538, 2010, 3430~
$ state_icpsr                  <dbl> 99, 44, 44, 44, 52, 52, 52, 52, 52, 52, ~
$ district_code                <dbl> 0, 2, 1, 3, 6, 3, 5, 2, 4, 1, 1, 3, 2, 8~
$ state_abbrev                 <chr> "USA", "GA", "GA", "GA", "MD", "MD", "MD~
$ party_code                   <chr> "5000", "4000", "4000", "4000", "5000", ~
$ occupancy                    <dbl> NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ last_means                   <dbl> NA, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ bioname                      <chr> "WASHINGTON, George", "BALDWIN, Abraham"~
$ bioguide_id                  <chr> NA, "B000084", "J000017", "M000234", "CO~
$ born                         <dbl> NA, 1754, 1757, 1739, 1730, 1755, 1756, ~
$ died                         <dbl> NA, 1807, 1806, 1812, 1796, 1815, 1815, ~
$ nominate_dim1                <dbl> NA, -0.165, -0.320, -0.428, 0.116, -0.08~
$ nominate_dim2                <dbl> NA, -0.373, -0.181, -0.317, -0.740, -0.3~
$ nominate_log_likelihood      <dbl> NA, -28.55029, -24.89986, -12.62728, -23~
$ nominate_geo_mean_probability <dbl> NA, 0.758, 0.776, 0.880, 0.783, 0.788, 0~
$ nominate_number_of_votes     <dbl> NA, 103, 98, 99, 96, 92, 94, 106, 103, 9~
$ nominate_number_of_errors    <dbl> NA, 12, 9, 2, 11, 13, 10, 34, 30, 20, 10~
$ conditional                  <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ nokken_poole_dim1            <dbl> NA, -0.429, -0.559, -0.413, 0.114, -0.09~
$ nokken_poole_dim2            <dbl> NA, -0.817, -0.052, -0.232, -0.779, -0.4~
```

This dataset contains ideological scores for all members of Congress. Work with Claude to understand what the nominate_dim1 variable represents.

2.1.2 9.2 Party Polarization Over Time

Ask Claude to help you analyze how the ideological distance between Democrats and Republicans in the most recent Congress.

Your Prompt to Claude: I have a dataset called congress with DW-NOMINATE scores. The nominate_dim1 variable ranges from -1 (liberal) to +1 (conservative). I want to analyze the ideological distance between Democrats and Republicans in Congress 118 (the most recent Congress). Calculate the average ideology score for each party and the gap between them. Use tidyverse. Here is the what the data look like.

Rows: 51,044 Columns: 22 $ congress 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~ $ chamber "President", "House", "House", "House", ~ $ icpsr 99869, 379, 4854, 6071, 1538, 2010, 3430~ $ state_icpsr 99, 44, 44, 44, 52, 52, 52, 52, 52, 52, ~ $ district_code 0, 2, 1, 3, 6, 3, 5, 2, 4, 1, 1, 3, 2, 8~ $ state_abbrev "USA", "GA", "GA", "GA", "MD", "MD", "MD~ $ party_code "5000", "4000", "4000", "4000", "5000", ~ $ occupancy NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~ $ last_means NA, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~ $ bioname "WASHINGTON, George", "BALDWIN, Abraham"~ $ bioguide_id NA, "B000084", "J000017", "M000234", "C0~ $ born NA, 1754, 1757, 1739, 1730, 1755, 1756, ~ $ died NA, 1807, 1806, 1812, 1796, 1815, 1815, ~ $ nominate_dim1 NA, -0.165, -0.320, -0.428, 0.116, -0.08~ $ nominate_dim2 NA, -0.373, -0.181, -0.317, -0.740, -0.3~ $ nominate_log_likelihood NA, -28.55029, -24.89986, -12.62728, -23~ $ nominate_geo_mean_probability NA, 0.758, 0.776, 0.880, 0.783, 0.788, 0~ $ nominate_number_of_votes NA, 103, 98, 99, 96, 92, 94, 106, 103, 9~ $ nominate_number_of_errors NA, 12, 9, 2, 11, 13, 10, 34, 30, 20, 10~ $ conditional NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~ $ nokken_poole_dim1 NA, -0.429, -0.559, -0.413, 0.114, -0.09~ $ nokken_poole_dim2 NA, -0.817, -0.052, -0.232, -0.779, -0.4~

Keep this simple and explain what you are doing

```
congress %>%
  filter(congress == 118) %>%
  filter(party_code %in% c("Democrat", "Republican")) %>%  # Using actual party names
  group_by(party_code) %>%
  summarise(
    mean_ideology = mean(nominate_dim1, na.rm = TRUE),
    n_members = n()
  ) %>%
  summarise(
    gap = max(mean_ideology) - min(mean_ideology)
  )
```

```
# A tibble: 1 x 1
    gap
  <dbl>
1 0.892
```

Interpretation: The analysis shows the current ideological distance between Democrats and Republicans in Congress 118. The gap represents how far apart the two parties are on the liberal-conservative spectrum, with larger values indicating greater polarization.

2.1.3 9.3 State-Level Analysis

Work with Claude to find which state has the most ideologically diverse congressional delegation in the current Congress.

Your Prompt to Claude: Using the congress dataset, help me identify which state has the most ideologically diverse delegation in Congress 118.

```r
congress %>%
  filter(congress == 118) %>%
  filter(!is.na(nominate_dim1)) %>%  # Remove missing ideology scores
  filter(chamber %in% c("House", "Senate")) %>%  # Only legislators, not President
  group_by(state_abbrev) %>%
  summarise(
    n_members = n(),
    mean_ideology = mean(nominate_dim1, na.rm = TRUE),
    sd_ideology = sd(nominate_dim1, na.rm = TRUE),
    min_ideology = min(nominate_dim1, na.rm = TRUE),
    max_ideology = max(nominate_dim1, na.rm = TRUE),
    range_ideology = max_ideology - min_ideology
  ) %>%
  filter(n_members > 1) %>%  # Need at least 2 members to calculate diversity
  arrange(desc(sd_ideology))
```

```
# A tibble: 50 x 7
   state_abbrev n_members mean_ideology sd_ideology min_ideology max_ideology
   <chr>            <int>         <dbl>       <dbl>        <dbl>        <dbl>
 1 GA                  16      0.218          0.564       -0.477        0.891
 2 CO                  11      0.0564         0.547       -0.431        0.894
 3 AZ                  11      0.187          0.527       -0.598        0.818
 4 TX                  41      0.247          0.517       -0.783        0.826
 5 WI                  11      0.240          0.491       -0.524        0.643
 6 VA                  13      0.000385       0.477       -0.549        0.8
 7 NC                  16      0.153          0.461       -0.498        0.703
 8 MO                  10      0.424          0.445       -0.453        0.927
 9 OH                  18      0.205          0.441       -0.453        0.85
10 MS                   6      0.339          0.430       -0.519        0.637
# i 40 more rows
# i 1 more variable: range_ideology <dbl>
```

2.2 10. School Climate and Outcomes

Dataset: school_climate.csv

Description: Student-level data nested in 500 schools measuring climate, discipline, academics.

Variables:

- `student_id`: Unique student identifier (int)
- `school_id`: School identifier (int)
- `grade_level`: 9-12 (int)
- `gender`: male, female, nonbinary (factor)
- `race_ethnicity`: Standard categories (factor)
- `suspensions`: Annual suspension count (int)
- `gpa`: 0-4 scale (num)
- `school_safety`: 0-10 student perception (int)
- `teacher_support`: 0-10 (int)
- `parent_involvement`: 0-10 (int)
- `school_poverty_rate`: % eligible for free lunch (num)

2.2.1 10.1 Data Loading and Quality Check

```r
# Load the dataset
school <- read_csv("school_climate.csv")
```

```
Rows: 5000 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): school_id, gender, race_ethnicity
dbl (8): student_id, grade_level, school_poverty_rate, gpa, suspensions, sch...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Check the structure
glimpse(school)
```

```
Rows: 5,000
Columns: 11
$ student_id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
```

```
$ school_id           <chr> "S371", "S313", "S053", "S083", "S092", "S082", "S~
$ grade_level         <dbl> 9, 12, 10, 9, 10, 12, 11, 10, 10, 11, 12, 11, 9, 1~
$ gender              <chr> "male", "female", "male", "female", "female", "fem~
$ race_ethnicity      <chr> "Other", "Black", "Other", "Black", "Latino", "Bla~
$ school_poverty_rate <dbl> 75.5, 88.0, 15.0, 89.3, 79.0, 49.2, 20.3, 80.3, 84~
$ gpa                 <dbl> 2.87, 3.33, 2.72, 4.00, 1.77, 3.58, 1.84, 2.31, 2.~
$ suspensions         <dbl> 1, 0, 2, 0, 0, 0, 0, 0, 1, 0, 2, 0, 1, 0, 0, 1, 2,~
$ school_safety       <dbl> 5.6, 3.4, 5.4, 5.5, 7.3, 7.0, 9.4, 4.9, 6.1, 5.4, ~
$ teacher_support     <dbl> 5.4, 6.1, 8.7, 8.2, 4.8, 5.1, 7.6, 6.5, 7.2, 6.8, ~
$ parent_involvement  <dbl> 9.1, 6.2, 6.9, 3.8, 3.0, 6.7, 5.1, 6.4, 6.5, 5.1, ~
```

```
# Data quality check
school %>%
    summarise(total_students = n(), n_schools = n_distinct(school_id),
        avg_students_per_school = n()/n_distinct(school_id),
        missing_gpa = sum(is.na(gpa)), missing_suspensions = sum(is.na(suspensions)),
        gpa_range = paste(min(gpa, na.rm = TRUE), "-", max(gpa,
            na.rm = TRUE)), suspension_range = paste(min(suspensions,
            na.rm = TRUE), "-", max(suspensions, na.rm = TRUE)))
```

```
# A tibble: 1 x 7
  total_students n_schools avg_students_per_school missing_gpa
           <int>     <int>                   <dbl>       <int>
1           5000       500                      10           0
# i 3 more variables: missing_suspensions <int>, gpa_range <chr>,
#   suspension_range <chr>
```

This dataset contains student-level data nested within schools.

2.2.2 10.2 Analyzing Discipline Disparities

Ask Claude to help you look for missing data.

Your Prompt to Claude: I have a school climate dataset called school. Help me check for missing data in all columns. Show me which columns have missing values and how many. Use tidyverse. It looks like this: Rows: 5,000 Columns: 11 $ student_id 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, … $ school_id “S371”, “S313”, “S053”, “S083”, “S092”, “S082”, “S3… $ grade_level 9, 12, 10, 9, 10, 12, 11, 10, 10, 11, 12, 11, 9, 12… $ gender ”male”, “female”, “male”, “female”, “female”, “fema… $ race_ethnicity ”Other”, “Black”, “Other”, “Black”, “Latino”, “Blac… $ school_poverty_rate 75.5, 88.0, 15.0, 89.3, 79.0, 49.2, 20.3, 80.3, 84.… $ gpa 2.87, 3.33, 2.72, 4.00, 1.77, 3.58, 1.84, 2.31, 2.9… $ suspensions 1, 0, 2, 0, 0, 0, 0, 0, 1, 0, 2, 0, 1, 0, 0, 1, 2, … $ school_safety 5.6, 3.4, 5.4, 5.5, 7.3, 7.0, 9.4, 4.9, 6.1, 5.4, 4… $ teacher_support 5.4, 6.1, 8.7, 8.2, 4.8, 5.1, 7.6, 6.5, 7.2, 6.8, 6… $ parent_involvement 9.1, 6.2, 6.9, 3.8, 3.0, 6.7, 5.1, 6.4, 6.5, 5.1, 5…

Give me the simplest approach.

```
school %>%
    summarise(across(everything(), ~sum(is.na(.x)))) %>%
    pivot_longer(everything(), names_to = "column", values_to = "missing_count")
```

```
# A tibble: 11 x 2
   column             missing_count
   <chr>                      <int>
 1 student_id                     0
 2 school_id                      0
 3 grade_level                    0
 4 gender                         0
 5 race_ethnicity                 0
 6 school_poverty_rate            0
 7 gpa                            0
 8 suspensions                    0
 9 school_safety                  0
10 teacher_support                0
11 parent_involvement             0
```

Interpretation: The missing data check shows which variables have incomplete information. Understanding patterns of missingness is crucial before conducting analysis, as missing data can bias results if not handled properly.

2.2.3 10.3 School Climate and Academic Performance

Work with Claude to find mean safety perceptions and school poverty rate.

Your Prompt to Claude: Using the school dataset, calculate the mean school_safety perception and mean school_poverty_rate. Use tidyverse.

```
school %>%
    summarise(mean_school_safety = mean(school_safety, na.rm = TRUE),
        mean_poverty_rate = mean(school_poverty_rate, na.rm = TRUE))
```

```
# A tibble: 1 x 2
  mean_school_safety mean_poverty_rate
               <dbl>             <dbl>
1               5.99              52.4
```

Interpretation: The mean values provide a baseline understanding of typical school safety perceptions and poverty levels across the schools in the dataset. These summary statistics help contextualize individual school conditions.

2.3 11. Protest Participation Patterns

Dataset: protest_panel.csv

Description: Three-wave panel survey on protest activity & engagement.

Variables:

- `respondent_id`: Panel respondent ID (int)
- `wave`: 1, 2, 3 (int)
- `age`: Baseline age (int)
- `education`: Highest degree (factor)
- `political_interest`: 0-10, time-varying (int)
- `ideology`: 1-7 scale (int)
- `participated_protest`: Protested in past 12 months (binary)
- `protest_issue`: Racial Justice, Climate, Abortion, Economy, NA (factor)
- `social_media_use`: 0-5 daily usage index (int)
- `voted_last_election`: Turnout indicator (binary)

2.3.1 11.1 Understanding Panel Data

```r
# Load the dataset
protest <- read_csv("protest_panel.csv")
```

```
Rows: 3000 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): protest_issue
dbl (9): respondent_id, wave, age, education, ideology, political_interest, ...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Examine panel structure
glimpse(protest)
```

```
Rows: 3,000
Columns: 10
$ respondent_id         <dbl> 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6~
$ wave                  <dbl> 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2~
$ age                   <dbl> 78, 78, 78, 74, 74, 74, 61, 61, 61, 33, 33, 33, 7~
$ education             <dbl> 4, 4, 4, 4, 4, 4, 5, 5, 5, 2, 2, 2, 5, 5, 5, 3, 3~
$ ideology              <dbl> 2, 2, 2, 2, 2, 2, 3, 3, 3, 7, 7, 7, 4, 4, 4, 7, 7~
$ political_interest    <dbl> 2, 7, 5, 3, 6, 5, 6, 5, 3, 3, 5, 4, 3, 6, 5, 6, 4~
$ social_media_use      <dbl> 5, 1, 1, 3, 3, 0, 3, 3, 5, 4, 2, 5, 4, 4, 2, 2, 3~
$ participated_protest  <dbl> 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0~
$ protest_issue         <chr> "Climate", NA, NA, "Racial Justice", "Economy", "~
$ voted_last_election   <dbl> 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1~
```

Panel data follows the same individuals over time.

2.3.2 11.2 Who Protests?

Ask Claude to help you create a profile of protesters vs non-protesters. Which group has more males? Which group is older on average?

Your Prompt to Claude: Using the protest panel data, compare protesters vs non-protesters. For each respondent, determine if they EVER protested across all three waves. Then compare age between ever-protesters and never-protesters. Note: this dataset doesn't have a gender variable, so we cannot compare by gender. Use tidyverse.

```r
ever_protested <- protest %>%
  group_by(respondent_id) %>%
  summarise(
    ever_protested = max(participated_protest),  # 1 if ever protested, 0 if never
    age = first(age)  # Age is constant across waves, so take first value
  ) %>%
  mutate(protester_type = ifelse(ever_protested == 1, "Ever-Protester", "Never-Protester"))

# Compare age between the two groups
age_comparison <- ever_protested %>%
  group_by(protester_type) %>%
  summarise(
    n = n(),
```

```
    mean_age = mean(age, na.rm = TRUE),
    sd_age = sd(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    min_age = min(age, na.rm = TRUE),
    max_age = max(age, na.rm = TRUE)
  )

# Display results
cat("Age Comparison: Ever-Protesters vs Never-Protesters\n")
```

```
Age Comparison: Ever-Protesters vs Never-Protesters
```

```
cat("====================================================\n")
```

```
====================================================
```

```
print(age_comparison)
```

```
# A tibble: 2 x 7
  protester_type      n mean_age sd_age median_age min_age max_age
  <chr>           <int>    <dbl>  <dbl>      <dbl>   <dbl>   <dbl>
1 Ever-Protester    554     49.9   18.0       51        18      80
2 Never-Protester   446     48.9   18.4       48.5      18      80
```

```
# Calculate the difference
age_diff <- age_comparison$mean_age[age_comparison$protester_type == "Ever-Protester"] -
            age_comparison$mean_age[age_comparison$protester_type == "Never-Protester"]
```

Interpretation: The age comparison shows differences between those who have ever participated in protests versus those who haven't. Age can be an important factor in political participation, with different age groups having varying levels of resources, time, and motivation for protest activities.