Week 2, Class 4: Practice Exercises - ANSWER KEY

Summary Statistics

2024-12-31

Non-AI Exercises

1. Understanding Measures of Central Tendency

1.1 Multiple Choice: Mean vs Median

When should you use the median instead of the mean?

a) When you want the most precise calculation
b) When the data has extreme outliers or is skewed
c) When all values are the same
d) When you have categorical data

Answer: **b) When the data has extreme outliers or is skewed**

Explanation: The median is robust to outliers because it represents the middle value when data is ordered. Unlike the mean, which is pulled toward extreme values, the median remains stable. For example, if incomes are \$30k, \$35k, \$40k, \$45k, and \$1 million, the mean (\$230k) is misleading while the median (\$40k) better represents the typical value.

1.2 Fill in the Blanks: Central Tendency

Complete these statements about measures of center:

1. The **mean** is the sum of all values divided by the count
2. The **median** is the middle value when data is ordered
3. The **mode** is the most frequently occurring value
4. When data is skewed right, the mean is typically **greater** than the median

5. The **mean** is most affected by outliers

Explanation: In right-skewed data (like income), extreme high values pull the mean upward while the median stays centered, making mean > median. The mean incorporates every value in its calculation, making it sensitive to outliers.

1.3 Code Detective: Grouped Summaries

What does this code calculate?

```
approval %>%
  group_by(party_id) %>%
  summarise(
    avg_approval = mean(congress_approval, na.rm = TRUE),
    n = n()
  )
```

This code calculates: **The average congressional approval rating for each political party (Democrat, Republican, Independent) and counts how many respondents are in each party group.**

Explanation: `group_by(party_id)` splits the data by party affiliation, then `summarise()` calculates the mean approval within each group. The `n()` function counts observations per group, and `na.rm = TRUE` removes missing values before calculating the mean.

2. Measures of Spread

2.1 Match: Measures of Spread

Match each measure with its definition:

Measures:

a) Range
b) Variance
c) Standard deviation
d) Interquartile range (IQR)

Definitions:

1. Square root of the variance
2. Maximum value minus minimum value
3. Average squared deviation from the mean

4. Distance between 25th and 75th percentiles

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

2.2 Multiple Choice: Standard Deviation

If approval ratings have a mean of 45% and standard deviation of 10%, approximately what percentage of responses fall between 35% and 55%?

a) 50%
b) 68%
c) 95%
d) 99%

Answer: **b) 68%**

2.3 True or False: Spread

Mark each statement as True (T) or False (F):

T Variance is always positive or zero **T** Standard deviation has the same units as the original data **T** A larger standard deviation means data is more spread out **T** The range is affected by outliers **T** IQR is more robust to outliers than standard deviation

Explanations: - Variance is squared deviations, so always 0 - SD is the square root of variance, returning to original units - Larger SD means values are further from the mean on average - Range uses min and max, both susceptible to outliers - IQR uses middle 50% of data, ignoring extremes

3. The `summarise()` Function

3.1 Fill in the Code: summarise()

Complete this code to calculate mean, median, and standard deviation:

```
approval %>%
  summarise(
    mean_approval = mean(congress_approval, na.rm = TRUE),
    median_approval = median(congress_approval, na.rm = TRUE),
    sd_approval = sd(congress_approval, na.rm = TRUE)
  )
```

Functions needed: **mean**, **median**, **sd**

3.2 Multiple Choice: NA Values

What happens when you run `mean(c(1, 2, NA, 4))`?

a) Returns 2.33 (ignores the NA)
b) Returns NA
c) Gives an error
d) Returns 2 (treats NA as 0)

Answer: **b) Returns NA**

Explanation: R propagates NA (missing) values through calculations by default. Any operation involving NA returns NA unless you explicitly tell R to remove them using `na.rm = TRUE`. This forces you to make conscious decisions about handling missing data.

4. Real-World Applications

4.1 Match: Statistical Concepts

Match polling concepts with statistical terms:

Polling Terms:

a) Margin of error
b) Sample size
c) Confidence level
d) Poll average

Statistical Terms:

1. Mean of multiple measurements
2. Number of observations (n)
3. Related to standard error
4. Probability of capturing true value

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

Explanation: Margin of error is typically 1.96 $\times$ standard error for 95% confidence. Sample size (n) is the count of respondents. Confidence level (e.g., 95%) is the probability the interval contains the true population value. Poll averages aggregate multiple polls using the mean.

4.2 Code Detective: Real Analysis

What question does this code answer?

```
approval %>%
  group_by(party_id, region) %>%
  summarise(
    mean_approval = mean(congress_approval, na.rm = TRUE),
    sd_approval = sd(congress_approval, na.rm = TRUE),
    count = n()
  )
```

This analysis shows: **How congressional approval ratings vary by both party affiliation AND geographic region, including the average approval, spread of opinions (standard deviation), and number of respondents in each party-region combination.**

Example interpretation: We might find that Democrats in the Northeast have higher approval (mean = 55%, sd = 12%) than Republicans in the South (mean = 35%, sd = 15%), showing both partisan and regional differences.

AI Exercises

For each AI exercise:

- Work with Claude to analyze the data
- Record your prompts and key findings

5. Congressional Approval Analysis

Dataset: congressional_approval.csv

Description: Survey data on congressional approval ratings with demographics.

Variables:

- `respondent_id`: Unique identifier (int)
- `age`: Respondent's age (int)
- `education`: Education level (chr)
- `party_id`: Democrat, Republican, Independent (chr)
- `income_category`: Income bracket (chr)
- `region`: Geographic region (chr)
- `congress_approval`: Approval rating for Congress, 0-100 scale (dbl)

5.1 Initial Data Exploration

```
# 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 becor
```

```
approval <- read_csv("congressional_approval.csv")
```

```
Rows: 2000 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (4): education, party_id, income_category, region
dbl (3): respondent_id, age, congress_approval

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.
```

```
# Examine the data
glimpse(approval)
```

```
Rows: 2,000
Columns: 7
$ respondent_id     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
$ age               <dbl> 31, 68, 41, 75, 19, 55, 27, 49, 57, 56, 55, 47, 18, ~
$ education         <chr> "Some College", "Bachelor's", "Some College", "Bache~
$ party_id          <chr> "Independent", "Republican", "Republican", "Democrat~
$ income_category   <chr> "$30k-$60k", "$30k-$60k", "$30k-$60k", "$30k-$60k", ~
$ region            <chr> "Midwest", "South", "South", "South", "West", "South~
$ congress_approval <dbl> 19.132044, 19.898068, 17.585641, 52.891775, 14.57116~
```

```r
summary(approval$congress_approval)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   19.07   28.51   28.71   38.15   75.88
```

Use Claude to calculate and interpret basic summary statistics for congressional approval ratings.

5.2 Approval by Party

Work with Claude to calculate mean and median approval ratings by party affiliation. What patterns do you observe? Are there differences in how spread out opinions are within each party?

Prompt to Claude: Using the approval dataset, calculate mean, median, and standard deviation of congress_approval by party_id. Also calculate the coefficient of variation (CV = sd/mean) to compare relative variability. Which party shows most consensus? Use tidyverse. The data look like this:

Rows: 535 Columns: 10 $ legislator_id 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~ $ name “Legislator_1”, “Legislator_2”, “Legislator_3”, “Legis~ $ party ”Democrat”, “Democrat”, “Republican”, “Democrat”, “Dem~ $ state ”MN”, “OK”, “MD”, “FL”, “NE”, “VA”, “VA”, “AZ”, “NM”, ~ $ district 30, 9, 37, 17, 44, 20, 36, 38, 34, 3, 50, 17, 24, 50, ~ $ ideology_score -0.22499579, -0.07570425, 0.60032405, -0.32344155, 0.3~ $ vote_attendance 43.05888, 87.71777, 70.22108, 91.00167, 85.69489, 59.1~ $ bills_sponsored 4, 5, 7, 1, 8, 5, 5, 6, 4, 7, 6, 7, 6, 6, 8, 3, 10, 7,~ $ years_service 20, 12, 4, 30, 28, 24, 8, 5, 13, 11, 21, 28, 25, 6, 11~ $ committee_count 3, 2, 3, 3, 1, 4, 7, 5, 3, 7, 2, 4, 1, 2, 4, 2, 4, 4, ~

```r
# Summary statistics by party
party_summary <- approval %>%
    group_by(party_id) %>%
    summarise(n = n(), mean_approval = mean(congress_approval,
        na.rm = TRUE), median_approval = median(congress_approval,
        na.rm = TRUE), sd_approval = sd(congress_approval, na.rm = TRUE),
        cv = sd_approval/mean_approval, min = min(congress_approval,
            na.rm = TRUE), max = max(congress_approval, na.rm = TRUE),
        range = max - min, .groups = "drop") %>%
    arrange(desc(mean_approval))

print(party_summary)
```

```
# A tibble: 3 x 9
  party_id        n mean_approval median_approval sd_approval    cv   min   max
  <chr>       <int>         <dbl>           <dbl>       <dbl> <dbl> <dbl> <dbl>
1 Democrat      682          36.5            37.4        12.2 0.333     0  73.4
2 Republican    697          27.1            26.9        12.1 0.448     0  75.9
3 Independent   621          22.0            21.5        12.2 0.553     0  64.6
# i 1 more variable: range <dbl>
```

Interpretation: Party differences in congressional approval reflect partisan polarization. The party of the current majority often shows higher approval from their base. Standard deviations reveal whether party members are unified or divided in their views.

5.3 Regional Variations

Ask Claude to help you explore how congressional approval varies by region. Which regions show the highest and lowest approval? Which have the most consensus (lowest standard deviation)?

Prompt to Claude: Analyze congressional approval by region. Calculate summary statistics, identify which regions have highest/lowest approval, and which show most/least consensus (using standard deviation). Also explore if regional differences are statistically meaningful. Use tidyverse.

```
# Regional analysis
regional_summary <- approval %>%
    group_by(region) %>%
    summarise(n = n(), mean_approval = mean(congress_approval,
        na.rm = TRUE), median_approval = median(congress_approval,
        na.rm = TRUE), sd_approval = sd(congress_approval, na.rm = TRUE),
        se_approval = sd_approval/sqrt(n), ci_lower = mean_approval -
            1.96 * se_approval, ci_upper = mean_approval + 1.96 *
            se_approval, .groups = "drop") %>%
    arrange(desc(mean_approval))

print("Regional approval rankings:")
```

```
[1] "Regional approval rankings:"
```

```
print(regional_summary)
```

```
# A tibble: 4 x 8
  region        n mean_approval median_approval sd_approval se_approval ci_lower
  <chr>     <int>         <dbl>           <dbl>       <dbl>       <dbl>    <dbl>
1 South       782          29.2            28.9        13.6       0.488     28.2
2 Midwest     424          28.9            28.5        12.9       0.629     27.6
3 Northeast   354          28.6            27.8        14.1       0.748     27.2
4 West        440          27.8            28.1        13.4       0.638     26.6
# i 1 more variable: ci_upper <dbl>
```

Interpretation: Regional variations in congressional approval often reflect local political cultures, economic conditions, and representation. Regions with more consensus (lower SD) may have more homogeneous political views.

5.4 Creating a Summary Report

Work with Claude to create a comprehensive summary table that shows approval statistics by both party and region. What insights emerge from this analysis?

Prompt to Claude: Create a comprehensive summary table showing congressional approval by party AND region combinations. Include mean, SD, and sample size. Identify interesting patterns like which party-region combination has highest/lowest approval. Format the output clearly. Use tidyverse.

```
# Comprehensive party-region analysis
party_region_summary <- approval %>%
    group_by(party_id, region) %>%
    summarise(n = n(), mean_approval = mean(congress_approval,
        na.rm = TRUE), sd_approval = sd(congress_approval, na.rm = TRUE),
        .groups = "drop") %>%
    arrange(party_id, desc(mean_approval))

# Display as a matrix
approval_matrix <- party_region_summary %>%
    select(party_id, region, mean_approval) %>%
    pivot_wider(names_from = region, values_from = mean_approval)

print("Mean Approval by Party and Region:")
```

```
[1] "Mean Approval by Party and Region:"
```

```r
print(approval_matrix)
```

```
# A tibble: 3 x 5
  party_id    Northeast South Midwest  West
  <chr>           <dbl> <dbl>   <dbl> <dbl>
1 Democrat         38.0  36.7    36.4  35.0
2 Independent      21.3  22.4    22.7  21.2
3 Republican       27.0  27.0    27.7  26.8
```

```r
# Find extremes
extremes <- party_region_summary %>%
    mutate(combo = paste(party_id, "-", region)) %>%
    summarise(highest = combo[which.max(mean_approval)], highest_approval = max(mean_approval
        lowest = combo[which.min(mean_approval)], lowest_approval = min(mean_approval),
        most_consensus = combo[which.min(sd_approval)], consensus_sd = min(sd_approval),
        most_divided = combo[which.max(sd_approval)], divided_sd = max(sd_approval))

print("\nKey Findings:")
```

```
[1] "\nKey Findings:"
```

```r
print(paste("Highest approval:", extremes$highest, "at", round(extremes$highest_approval,
    1)))
```

```
[1] "Highest approval: Democrat - Northeast at 38"
```

```r
print(paste("Lowest approval:", extremes$lowest, "at", round(extremes$lowest_approval,
    1)))
```

```
[1] "Lowest approval: Independent - West at 21.2"
```

```r
print(paste("Most unified group:", extremes$most_consensus, "SD =",
    round(extremes$consensus_sd, 1)))
```

```
[1] "Most unified group: Democrat - Midwest SD = 11.3"
```

Interpretation: The intersection of party and region reveals nuanced patterns in political attitudes. Some combinations (like Democrats in liberal regions) show high approval and consensus, while others show internal divisions.

6. Income and Wealth Distribution

Dataset: household_panel.csv

Description: Panel data tracking household economic conditions over time.

Variables:

- `hh_id`: Unique household identifier (int)
- `year`: Year of observation (int)
- `head_age`: Age of household head (int)
- `education_head`: Education of household head (chr)
- `num_children`: Number of children (int)
- `income`: Annual household income (dbl)
- `has_employer_insurance`: Has employer insurance (lgl)
- `self_reported_health`: Health status (chr)
- `unexpected_expense`: Can handle $400 expense (lgl)

6.1 Loading and Understanding the Data

```
# Load the dataset
household <- read_csv("household_panel.csv")
```

```
Rows: 6000 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (9): hh_id, year, head_age, education_head, num_children, income, has_em...

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.
```

```
# Check the structure
glimpse(household)
```

```
Rows: 6,000
Columns: 9
$ hh_id                 <dbl> 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,~
$ year                  <dbl> 2019, 2020, 2021, 2022, 2023, 2024, 2019, 2020,~
$ head_age              <dbl> 58, 58, 58, 58, 58, 58, 26, 26, 26, 26, 26, 26,~
$ education_head        <dbl> 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4,~
$ num_children          <dbl> 1, 1, 0, 2, 0, 0, 0, 1, 0, 1, 3, 0, 2, 3, 2, 2,~
```

```
$ income                 <dbl> 47159, 114000, 178018, 60681, 42426, 40098, 931~
$ has_employer_insurance <dbl> 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,~
$ self_reported_health   <dbl> 4, 1, 4, 3, 3, 3, 3, 2, 3, 2, 3, 2, 2, 4, 4, 3,~
$ unexpected_expense     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,~
```

This is panel data (same households observed over multiple years). Ask Claude to help you understand the income and wealth distributions.

6.2 Income and Health

Work with Claude to explore the relationship between household income and self-reported health status. Do higher-income households report better health? Calculate appropriate summary statistics.

Prompt to Claude: Analyze the relationship between income and self_reported_health. Calculate mean and median income by health status, and explore what percentage of each income quartile reports good/excellent health. Use tidyverse. The data look like this:

Rows: 6,000 Columns: 9 $ hh_id 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,~ $ year 2019, 2020, 2021, 2022, 2023, 2024, 2019, 2020,~ $ head_age 58, 58, 58, 58, 58, 58, 26, 26, 26, 26, 26, 26,~ $ education_head 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4,~ $ num_children 1, 1, 0, 2, 0, 0, 0, 1, 0, 1, 3, 0, 2, 3, 2, 2,~ $ income 47159, 114000, 178018, 60681, 42426, 40098, 931~ $ has_employer_insurance 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,~ $ self_reported_health 4, 1, 4, 3, 3, 3, 3, 2, 3, 2, 3, 2, 2, 4, 4, 3,~ $ unexpected_expense 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,~

```r
household %>%
    group_by(self_reported_health) %>%
    summarise(n = n(), mean_income = mean(income, na.rm = TRUE),
        median_income = median(income, na.rm = TRUE)) %>%
    arrange(self_reported_health) %>%
    print()
```

```
# A tibble: 5 x 4
  self_reported_health     n mean_income median_income
                 <dbl> <int>       <dbl>         <dbl>
1                    1   291      70722.        62915
2                    2   943      72501.        64183
3                    3  2413      70827.        63640
4                    4  1790      71950.        64678.
5                    5   563      71696.        64635
```

```
household %>%
    mutate(income_quartile = ntile(income, 4), good_health = ifelse(self_reported_health >=
        3, 1, 0)  # Assuming 3-4 = good/excellent
) %>%
    group_by(income_quartile) %>%
    summarise(n = n(), pct_good_health = mean(good_health, na.rm = TRUE) *
        100) %>%
    print()
```

```
# A tibble: 4 x 3
  income_quartile     n pct_good_health
            <int> <int>           <dbl>
1               1  1500            79.6
2               2  1500            78.8
3               3  1500            79.9
4               4  1500            79.4
```

Interpretation: The income-health gradient is a well-documented phenomenon where higher income is associated with better health outcomes. This relationship reflects multiple factors including access to healthcare, nutrition, stress levels, and living conditions.

6.3 Financial Security

Ask Claude to help you analyze financial security using the unexpected_expense variable. What percentage of households can handle a $400 emergency expense? How does this vary by income and education?

Prompt to Claude: Analyze financial security using the unexpected_expense variable (ability to handle $400 emergency). Calculate overall percentage who can handle it, then break down by income quartiles and education levels. Identify which groups are most financially vulnerable. Use tidyverse.

```
library(tidyverse)

# Overall percentage who can handle $400 emergency
household %>%
    summarise(pct_can_handle = mean(unexpected_expense == 1,
        na.rm = TRUE) * 100)
```

```
# A tibble: 1 x 1
```

```
  pct_can_handle
           <dbl>
1           18.5
```

```
# By income quartiles
household %>%
    mutate(income_quartile = ntile(income, 4)) %>%
    group_by(income_quartile) %>%
    summarise(pct_can_handle = mean(unexpected_expense == 1,
        na.rm = TRUE) * 100)
```

```
# A tibble: 4 x 2
  income_quartile pct_can_handle
            <int>          <dbl>
1               1           18.9
2               2           19
3               3           18.8
4               4           17.3
```

```
# By education level
household %>%
    group_by(education_head) %>%
    summarise(pct_can_handle = mean(unexpected_expense == 1,
        na.rm = TRUE) * 100) %>%
    arrange(education_head)
```

```
# A tibble: 5 x 2
  education_head pct_can_handle
           <dbl>          <dbl>
1              1           17.9
2              2           18.1
3              3           17.2
4              4           21.2
5              5           18.0
```

```
# Most vulnerable groups (lowest percentages)
household %>%
    mutate(income_quartile = ntile(income, 4)) %>%
    group_by(income_quartile) %>%
    summarise(pct_can_handle = mean(unexpected_expense == 1,
        na.rm = TRUE) * 100) %>%
    filter(pct_can_handle == min(pct_can_handle))
```

```
# A tibble: 1 x 2
  income_quartile pct_can_handle
            <int>          <dbl>
1               4           17.3
```

Interpretation: The ability to handle unexpected expenses is a key indicator of financial security. Large percentages of lower-income households lacking this cushion reveals financial fragility that can spiral into debt or deprivation when emergencies occur.

7. Voter Turnout Patterns

Dataset: voter_turnout_simple.csv

Description: State-level voter turnout data from recent elections.

Variables:

- `state`: State name (chr)
- `turnout_2020`: Turnout percentage in 2020 (dbl)
- `turnout_2016`: Turnout percentage in 2016 (dbl)
- `population_millions`: State population in millions (dbl)

7.1 Turnout Overview

```
# Load the dataset
turnout <- read_csv("voter_turnout_simple.csv")
```

```
Rows: 10 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): state
dbl (3): turnout_2020, turnout_2016, population_millions

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.
```

```
# Explore the data
glimpse(turnout)
```

```
Rows: 10
Columns: 4
$ state              <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "Calif~
$ turnout_2020       <dbl> 63.1, 58.9, 60.0, 54.9, 64.5, 76.4, 65.2, 66.0, 66~
$ turnout_2016       <dbl> 59.0, 61.5, 56.0, 53.2, 58.4, 71.9, 65.7, 61.8, 65~
$ population_millions <dbl> 5.0, 0.7, 7.3, 3.0, 39.5, 5.8, 3.6, 1.0, 21.5, 10.7
```

Work with Claude to calculate summary statistics for voter turnout in both 2016 and 2020. What changed between elections?

7.2 State-Level Patterns

Ask Claude to help you identify which states had the highest and lowest turnout in both elections. How consistent are state rankings between 2016 and 2020?

Prompt to Claude: Identify the top 3 and bottom 3 states for turnout in each election. Then check how consistent state rankings are between 2016 and 2020 using correlation. Do high-turnout states stay high? Use tidyverse.

```
turnout %>%
    arrange(desc(turnout_2020)) %>%
    slice(1:3) %>%
    select(state, turnout_2020)
```

```
# A tibble: 3 x 2
  state    turnout_2020
  <chr>           <dbl>
1 Colorado         76.4
2 Florida          66.2
3 Georgia          66.2
```

```
turnout %>%
    arrange(turnout_2020) %>%
    slice(1:3) %>%
    select(state, turnout_2020)
```

```
# A tibble: 3 x 2
  state    turnout_2020
  <chr>           <dbl>
1 Arkansas         54.9
2 Alaska           58.9
3 Arizona          60
```

```r
# Top 3 and bottom 3 states for 2016
turnout %>%
    arrange(desc(turnout_2016)) %>%
    slice(1:3) %>%
    select(state, turnout_2016)
```

```
# A tibble: 3 x 2
  state       turnout_2016
  <chr>              <dbl>
1 Colorado            71.9
2 Connecticut         65.7
3 Florida             65.3
```

```r
turnout %>%
    arrange(turnout_2016) %>%
    slice(1:3) %>%
    select(state, turnout_2016)
```

```
# A tibble: 3 x 2
  state      turnout_2016
  <chr>             <dbl>
1 Arkansas           53.2
2 Arizona            56
3 California         58.4
```

Interpretation: High correlation between years suggests persistent state-level factors affecting turnout (voting laws, civic culture, demographics). States with big rank changes may have had specific mobilization efforts or law changes.

7.3 Turnout Changes

Work with Claude to calculate how turnout changed from 2016 to 2020 for each state. Which states saw the biggest increases or decreases?

Prompt to Claude: Calculate turnout changes from 2016 to 2020 for each state. Create categories for the size of change (large increase, moderate increase, etc.). Also explore if turnout changes relate to state population size. Use tidyverse.

```r
turnout_changes <- turnout %>%
    mutate(change = turnout_2020 - turnout_2016, change_category = case_when(change >=
        5 ~ "Large increase", change >= 2 ~ "Moderate increase",
        change >= -2 ~ "Stable", change >= -5 ~ "Moderate decrease",
        TRUE ~ "Large decrease")) %>%
    arrange(desc(change))

# Display changes by state
turnout_changes %>%
    select(state, turnout_2016, turnout_2020, change, change_category)
```

```
# A tibble: 10 x 5
   state       turnout_2016 turnout_2020 change change_category
   <chr>              <dbl>        <dbl>  <dbl> <chr>
 1 Georgia             59.4         66.2  6.80  Large increase
 2 California          58.4         64.5  6.1   Large increase
 3 Colorado            71.9         76.4  4.5   Moderate increase
 4 Delaware            61.8         66    4.20  Moderate increase
 5 Alabama             59           63.1  4.1   Moderate increase
 6 Arizona             56           60    4     Moderate increase
 7 Arkansas            53.2         54.9  1.70  Stable
 8 Florida             65.3         66.2  0.900 Stable
 9 Connecticut         65.7         65.2 -0.5   Stable
10 Alaska              61.5         58.9 -2.6   Moderate decrease
```

```r
# Count states in each category
turnout_changes %>%
    count(change_category)
```

```
# A tibble: 4 x 2
  change_category       n
  <chr>             <int>
1 Large increase        2
2 Moderate decrease     1
3 Moderate increase     4
4 Stable                3
```

```r
# Explore relationship with population size
correlation <- cor(turnout_changes$population_millions, turnout_changes$change)
print(paste("Correlation between population size and turnout change:",
    round(correlation, 3)))
```

```
[1] "Correlation between population size and turnout change: 0.397"
```

```r
# Show turnout changes by population size groups
turnout_changes %>%
    mutate(pop_size = case_when(population_millions < 2 ~ "Small (<2M)",
        population_millions < 10 ~ "Medium (2-10M)", TRUE ~ "Large (>10M)")) %>%
    group_by(pop_size) %>%
    summarise(n_states = n(), avg_change = mean(change), median_change = median(change))
```

```
# A tibble: 3 x 4
  pop_size       n_states avg_change median_change
  <chr>             <int>      <dbl>         <dbl>
1 Large (>10M)          3      4.60          6.1
2 Medium (2-10M)        5      2.76          4
3 Small (<2M)           2      0.800         0.800
```

Interpretation: Turnout changes reflect various factors including competitiveness, mobilization efforts, voting law changes, and demographic shifts. The relationship (or lack thereof) with population size indicates whether changes were driven by state-specific or national factors.

8. Legislative Productivity

Dataset: rollcalls.csv

Description: Individual member votes on congressional bills.

Variables:

- `congress`: Congress number (int)
- `bill_id`: Unique bill identifier (int)
- `member_id`: Member identifier (int)
- `party`: Political party (chr)
- `ideology`: Member ideology score (dbl)
- `bill_ideology`: Bill ideology score (dbl)
- `vote`: Vote choice (chr)
- `district_partisanship`: District lean (dbl)

8.1 Understanding Voting Patterns

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

```
Rows: 3000 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): party
dbl (7): congress, bill_id, member_id, ideology, bill_ideology, vote, distri...

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
# Look at the data
glimpse(rollcalls)
```

```
Rows: 3,000
Columns: 8
$ congress              <dbl> 114, 117, 116, 114, 114, 116, 116, 118, 114, 116~
$ bill_id               <dbl> 500, 238, 494, 352, 445, 177, 621, 40, 426, 165,~
$ member_id             <dbl> 66, 409, 327, 467, 532, 487, 290, 245, 464, 90, ~
$ party                 <chr> "Dem", "Rep", "Dem", "Dem", "Rep", "Rep", "Rep",~
$ ideology              <dbl> 0.20949025, 0.41277467, -0.27790813, 0.10842921,~
$ bill_ideology         <dbl> 0.43897533, -0.33229308, -0.27514849, -0.1348527~
$ vote                  <dbl> 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, ~
$ district_partisanship <dbl> 11.9, -20.8, 12.0, 15.8, -11.3, -25.6, 2.2, -13.~
```

Use Claude to calculate basic statistics about congressional voting patterns, including passage rates and participation.

8.2 Party Voting Patterns

Work with Claude to analyze how members vote based on their party affiliation. Do Democrats and Republicans vote differently on bills?

Prompt to Claude: Analyze party-line voting. For each bill, calculate what percentage of Democrats voted Yes vs Republicans. Identify bills with strongest party splits and those with bipartisan agreement. Use tidyverse.

```r
library(tidyverse)

# Calculate party voting percentages for each bill
party_voting <- rollcalls %>%
  group_by(bill_id, party) %>%
  summarise(
    total_votes = n(),
    yes_votes = sum(vote == "Yes"),
    pct_yes = mean(vote == "Yes") * 100,
    .groups = 'drop'
  ) %>%
  pivot_wider(
    names_from = party,
    values_from = c(pct_yes, total_votes),
    names_sep = "_"
  ) %>%
  mutate(
    party_split = abs(pct_yes_Dem - pct_yes_Rep)
  )

# Identify bills with strongest party splits (top 10)
strongest_splits <- party_voting %>%
  arrange(desc(party_split)) %>%
  slice(1:10) %>%
  select(bill_id, pct_yes_Dem, pct_yes_Rep, party_split)

print("Bills with Strongest Party Splits:")
```

```
[1] "Bills with Strongest Party Splits:"
```

```r
print(strongest_splits)
```

```
# A tibble: 10 x 4
   bill_id pct_yes_Dem pct_yes_Rep party_split
     <dbl>       <dbl>       <dbl>       <dbl>
 1       1           0           0           0
 2       2           0           0           0
 3       5           0           0           0
 4       6           0           0           0
 5       8           0           0           0
 6       9           0           0           0
```

```
 7      13           0           0           0
 8      14           0           0           0
 9      16           0           0           0
10      17           0           0           0
```

```
# Identify bills with bipartisan agreement (both >70% or both <30%)
bipartisan <- party_voting %>%
  filter(
    (pct_yes_Dem > 70 & pct_yes_Rep > 70) |  # Both parties support
    (pct_yes_Dem < 30 & pct_yes_Rep < 30)     # Both parties oppose
  ) %>%
  arrange(party_split) %>%
  slice(1:10) %>%
  select(bill_id, pct_yes_Dem, pct_yes_Rep, party_split)

print("Bills with Bipartisan Agreement:")
```

```
[1] "Bills with Bipartisan Agreement:"
```

```
print(bipartisan)
```

```
# A tibble: 10 x 4
   bill_id pct_yes_Dem pct_yes_Rep party_split
     <dbl>       <dbl>       <dbl>       <dbl>
 1       1           0           0           0
 2       2           0           0           0
 3       5           0           0           0
 4       6           0           0           0
 5       8           0           0           0
 6       9           0           0           0
 7      13           0           0           0
 8      14           0           0           0
 9      16           0           0           0
10      17           0           0           0
```

```
# Overall summary of party splits
party_voting %>%
  summarise(
    mean_split = mean(party_split, na.rm = TRUE),
    median_split = median(party_split, na.rm = TRUE),
    max_split = max(party_split, na.rm = TRUE)
  )
```

```
# A tibble: 1 x 3
  mean_split median_split max_split
       <dbl>        <dbl>     <dbl>
1          0            0         0
```

Interpretation: Party-line voting has increased in recent decades. Bills with high party splits often involve ideological issues, while bipartisan bills typically address non-controversial or crisis issues.

8.3 Ideology and Voting

Ask Claude to help you explore how member ideology relates to voting patterns. Do more extreme members vote differently than moderates?

Prompt to Claude: Examine how member ideology affects voting behavior. Compare voting patterns of moderates (ideology near 0) versus extremists (far from 0). Also analyze how often members vote against bills that don't match their ideology. Use tidyverse.

```
# Categorize members by ideology
votes_categorized <- rollcalls %>%
    mutate(ideology_type = case_when(abs(ideology) < 0.2 ~ "Moderate",
        abs(ideology) < 0.4 ~ "Somewhat partisan", TRUE ~ "Extremist"),
        ideology_match = ifelse((ideology > 0 & bill_ideology >
            0) | (ideology < 0 & bill_ideology < 0), "Match",
            "Mismatch"))

# Compare voting patterns by ideology type
voting_by_ideology <- votes_categorized %>%
    group_by(ideology_type) %>%
    summarise(n_votes = n(), pct_yes = mean(vote == "Yes") *
        100, avg_ideology = mean(abs(ideology))) %>%
    arrange(avg_ideology)

print("Voting Patterns by Ideology Type:")
```

```
[1] "Voting Patterns by Ideology Type:"
```

```
print(voting_by_ideology)
```

```
# A tibble: 3 x 4
  ideology_type     n_votes pct_yes avg_ideology
  <chr>               <int>   <dbl>        <dbl>
1 Moderate             1131       0       0.0974
2 Somewhat partisan     932       0       0.291
3 Extremist             937       0       0.614
```

```
# Analyze voting against ideological match
votes_categorized %>%
    group_by(ideology_match) %>%
    summarise(n_votes = n(), pct_yes = mean(vote == "Yes") *
        100)
```

```
# A tibble: 2 x 3
  ideology_match n_votes pct_yes
  <chr>            <int>   <dbl>
1 Match             1486       0
2 Mismatch          1514       0
```

```
# How often members vote against their ideology
against_ideology <- votes_categorized %>%
    filter(ideology_match == "Mismatch") %>%
    group_by(ideology_type) %>%
    summarise(pct_voted_yes_despite_mismatch = mean(vote == "Yes") *
        100)

print("Voting Yes Despite Ideological Mismatch:")
```

```
[1] "Voting Yes Despite Ideological Mismatch:"
```

```
print(against_ideology)
```

```
# A tibble: 3 x 2
  ideology_type     pct_voted_yes_despite_mismatch
  <chr>                                      <dbl>
1 Extremist                                      0
2 Moderate                                       0
3 Somewhat partisan                              0
```

```r
# Member-level summary: who crosses party lines most?
crossover_members <- votes_categorized %>%
    group_by(member_id, party, ideology) %>%
    summarise(total_votes = n(), mismatched_yes_votes = sum(ideology_match ==
        "Mismatch" & vote == "Yes"), pct_crossover = (mismatched_yes_votes/total_votes) *
        100, .groups = "drop") %>%
    arrange(desc(pct_crossover)) %>%
    slice(1:10)

print("Members Who Vote Against Ideology Most Often:")
```

```
[1] "Members Who Vote Against Ideology Most Often:"
```

```r
print(crossover_members)
```

```
# A tibble: 10 x 6
   member_id party ideology total_votes mismatched_yes_votes pct_crossover
       <dbl> <chr>    <dbl>       <int>                <int>         <dbl>
 1         1 Dem     -0.627           1                    0             0
 2         1 Dem     -0.290           1                    0             0
 3         1 Dem     -0.260           1                    0             0
 4         1 Dem      0.216           1                    0             0
 5         1 Dem      0.257           1                    0             0
 6         1 Dem      0.525           1                    0             0
 7         1 Rep     -0.326           1                    0             0
 8         2 Dem     -0.670           1                    0             0
 9         2 Dem     -0.564           1                    0             0
10         2 Dem     -0.427           1                    0             0
```

Interpretation: Ideological extremity correlates with partisan voting behavior. Moderates are more likely to cross party lines and vote based on specific bill content rather than party position. This dynamic affects legislative outcomes and coalition building.