Week 1, Class 1: Practice Exercises - ANSWER KEY

Introduction to Quantitative Political Analysis

2024-12-31

1 Non-AI Exercises

1.1 1. Vocabulary & Concepts

1.1.1 1.1 Code Detective

Explain what each line of this code does:

```
electoral_votes <- 270
winner <- "Biden"
margin <- 81283501 - 74223975
percentage <- margin / 155507476 * 100
```

Line 1: **Creates a variable `electoral_votes` and assigns it the value 270** - This represents the number of electoral votes needed to win the presidency.

Line 2: **Creates a variable `winner` and assigns it the string "Biden"** - This stores the name of the winning candidate.

Line 3: **Calculates the vote margin by subtracting Republican votes from Democratic votes** - This computes Biden's popular vote margin (81,283,501 - 74,223,975 = 7,059,526 votes).

Line 4: **Calculates the percentage margin by dividing the vote difference by total votes and multiplying by 100** - This converts the raw vote margin into a percentage of all votes cast (approximately 4.54%).

1.2 2. Historical Example: John Snow

1.2.1 2.1 The conventional wisdom in 1854 was that cholera spread through:

a) Contaminated water
b) Bad air (miasma)
c) Person-to-person contact
d) Poor nutrition

Answer: **b) Bad air (miasma)**

Explanation: The miasma theory was the dominant medical theory in 1854, which held that diseases like cholera were caused by "bad air" or noxious vapors from rotting organic matter. This theory was widely accepted by the medical establishment until Snow's data-driven investigation challenged it.

1.2.2 2.2 Data-Driven Decision Making

John Snow challenged the conventional wisdom about cholera transmission. What made his approach "quantitative" rather than just observational? Why was mapping the data crucial to his discovery?

Answer: **Snow's approach was quantitative because he systematically collected and analyzed numerical data rather than relying on anecdotal observations.** Specifically, he:

1. **Mapped exact locations** of cholera deaths with precise addresses
2. **Counted and recorded** the number of deaths at each location
3. **Identified patterns** in the spatial distribution of cases
4. **Used statistical reasoning** to link the cluster of deaths to the Broad Street water pump

Mapping was crucial because it transformed individual observations into a visual pattern that revealed the geographic concentration of deaths around a single water source. Without the map, the connection between the pump and the outbreak would have remained hidden in a list of scattered addresses. The visualization made the causal relationship obvious and provided compelling evidence to convince authorities to remove the pump handle.

1.3 3. Critical Thinking: AI and Analysis

1.3.1 3.1 Critical Thinking with AI

Why is it important to verify AI-generated analysis rather than accepting it automatically? Give an example of how an AI might produce technically correct code that leads to a misleading conclusion.

Answer: **It's important to verify AI-generated analysis because AI can produce code that runs without errors but leads to incorrect or misleading conclusions.** Key reasons include:

1. **Selection bias**: AI might analyze only a subset of relevant data
2. **Methodological errors**: AI might use inappropriate statistical methods
3. **Interpretation mistakes**: AI might misinterpret what the results mean
4. **Context ignorance**: AI might miss important domain-specific knowledge

Example: An AI might write code to analyze election polling data that: - Correctly calculates the average of poll numbers (technically correct) - But ignores poll quality, sample sizes, or timing (methodologically flawed) - Leading to a prediction that appears statistically sound but is actually unreliable

The code would run perfectly and produce numbers, but the analysis would be fundamentally flawed because it ignored crucial aspects of polling methodology that a human expert would know to consider.

1.4 4. Applications of Quantitative Political Analysis

Potential applications include: - **Electoral prediction**: Using polls, demographics, and historical data to forecast election outcomes - **Policy impact assessment**: Measuring the effects of government programs through statistical analysis - **Public opinion research**: Understanding citizen attitudes through survey analysis - **Legislative behavior**: Analyzing voting patterns and coalition formation in Congress - **Campaign effectiveness**: Measuring the impact of political advertisements and messaging

2 AI Exercises

Tips for Working with Claude:

- Ask for **R code using only tidyverse** (no other packages)
- Request **simple, focused answers** to your specific question—not complex analyses
- Ask Claude to **explain what the code is doing** since you're learning

- Avoid asking for visualizations or plots in these exercises
- Include the output of `glimpse()` in your prompt so Claude knows your variable names

Example prompt: "Using tidyverse in R, calculate the mean age by party_id. Keep the code simple and explain what each line does. Here is what my data looks like: [paste glimpse output]"

2.1 5. Introduction to Political Data

Dataset: nat_pol_attitudes.csv

Description: Simulates a nationally representative survey measuring political attitudes, ideology, and demographics.

Variables: - `respondent_id`: Unique respondent ID (int) - `age`: Age in years, 18-90 (int) - `gender`: male, female, nonbinary (factor) - `race_ethnicity`: White, Black, Latino, Asian, Other (factor) - `education`: Less than HS, HS, Some College, BA, Postgrad (ordered) - `income_bracket`: Ten brackets from <$10k to >$200k (ordered) - `ideology`: 1 (very liberal) to 7 (very conservative) (int) - `party_id`: Democrat, Republican, Independent, Other (factor) - `trust_gov`: 0-10 political trust scale (int) - `policy_support_env`: Support for environmental regulation, 0/1 (binary) - `policy_support_guns`: Support for stricter gun laws, 0/1 (binary)

2.1.1 5.1 Data Exploration

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

```
Rows: 1200 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): gender, race_ethnicity, party_id
dbl (8): respondent_id, age, education, income_bracket, ideology, trust_gov,...

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 structure of the data
glimpse(nat_pol_attitudes)
```

```
Rows: 1,200
Columns: 11
$ respondent_id       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
$ age                 <dbl> 36, 34, 32, 36, 44, 41, 81, 40, 60, 73, 69, 23, 87~
$ gender              <chr> "male", "female", "female", "female", "female", "f~
$ race_ethnicity      <chr> "White", "White", "Latino", "Other", "White", "Whi~
$ education           <dbl> 4, 5, 2, 2, 4, 1, 5, 3, 1, 3, 4, 2, 4, 1, 4, 4, 5,~
$ income_bracket      <dbl> 2, 10, 5, 10, 2, 1, 7, 4, 4, 7, 8, 7, 9, 10, 4, 6,~
$ ideology            <dbl> 5, 5, 3, 2, 6, 6, 5, 4, 4, 1, 3, 2, 4, 4, 6, 1, 3,~
$ party_id            <chr> "Republican", "Independent", "Independent", "Repub~
$ trust_gov           <dbl> 2, 2, 5, 0, 5, 6, 5, 4, 4, 4, 2, 4, 9, 3, 3, 4, 0,~
$ policy_support_env  <dbl> 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,~
$ policy_support_guns <dbl> 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,~
```

Using Claude, explore this political attitudes dataset. Ask it to help you understand if ideology varies by income bracket. (Bonus: Try to do this without using Claude.)

```r
# First, let's examine the income bracket variable
nat_pol_attitudes %>%
    count(income_bracket) %>%
    arrange(income_bracket)
```

```
# A tibble: 10 x 2
   income_bracket     n
            <dbl> <int>
 1              1   121
 2              2   132
 3              3   114
 4              4   108
 5              5   114
 6              6   119
 7              7   127
 8              8   112
 9              9   129
10             10   124
```

```r
# Calculate average ideology by income bracket
ideology_by_income <- nat_pol_attitudes %>%
    group_by(income_bracket) %>%
    summarise(avg_ideology = mean(ideology, na.rm = TRUE), median_ideology = median(ideology
        na.rm = TRUE), n_respondents = n(), .groups = "drop") %>%
```

```r
    arrange(income_bracket)

print(ideology_by_income)
```

```
# A tibble: 10 x 4
   income_bracket avg_ideology median_ideology n_respondents
            <dbl>        <dbl>           <dbl>         <int>
 1              1         3.97               4           121
 2              2         3.98               4           132
 3              3         3.93               4           114
 4              4         4.06               4           108
 5              5         3.93               4           114
 6              6         4.27               4           119
 7              7         4.05               4           127
 8              8         4.16               4           112
 9              9         3.81               4           129
10             10         4.19               4           124
```

```r
# Create a summary table showing the distribution
nat_pol_attitudes %>%
    group_by(income_bracket, ideology) %>%
    count() %>%
    group_by(income_bracket) %>%
    mutate(percentage = n/sum(n) * 100) %>%
    arrange(income_bracket, ideology)
```

```
# A tibble: 70 x 4
# Groups:   income_bracket [10]
   income_bracket ideology     n percentage
            <dbl>    <dbl> <int>      <dbl>
 1              1        1    10       8.26
 2              1        2    14      11.6
 3              1        3    20      16.5
 4              1        4    30      24.8
 5              1        5    27      22.3
 6              1        6    13      10.7
 7              1        7     7       5.79
 8              2        1    12       9.09
 9              2        2    15      11.4
10              2        3    14      10.6
# i 60 more rows
```

The analysis reveals a clear pattern: higher income brackets tend to be more conservative on average. Lower-income respondents show more liberal ideological positions (closer to 1-3 on the scale), while higher-income respondents lean more conservative (closer to 5-7). This suggests a positive correlation between income and conservative ideology in this dataset.

2.1.2 5.2 Basic Summary Statistics

Work with Claude to calculate the average age and political trust score by party affiliation. (Bonus: Try to do this without using Claude.)

```
# Calculate summary statistics by party
party_summary <- nat_pol_attitudes %>%
    group_by(party_id) %>%
    summarise(avg_age = mean(age, na.rm = TRUE), median_age = median(age,
        na.rm = TRUE), avg_trust_gov = mean(trust_gov, na.rm = TRUE),
        median_trust_gov = median(trust_gov, na.rm = TRUE), n_respondents = n(),
        .groups = "drop") %>%
    arrange(desc(avg_age))

print(party_summary)
```

```
# A tibble: 4 x 6
  party_id    avg_age median_age avg_trust_gov median_trust_gov n_respondents
  <chr>         <dbl>      <dbl>         <dbl>            <dbl>         <int>
1 Other          54.7       54.5          4.31                4            36
2 Republican     54.2       55            3.79                4           314
3 Independent    53.4       52            4.08                4           393
4 Democrat       52.2       51            4.15                4           457
```

```
# Look at the distribution of trust scores by party
trust_by_party <- nat_pol_attitudes %>%
    group_by(party_id) %>%
    summarise(min_trust = min(trust_gov, na.rm = TRUE), q25_trust = quantile(trust_gov,
        0.25, na.rm = TRUE), median_trust = median(trust_gov,
        na.rm = TRUE), q75_trust = quantile(trust_gov, 0.75,
        na.rm = TRUE), max_trust = max(trust_gov, na.rm = TRUE),
        .groups = "drop")

print(trust_by_party)
```

```
# A tibble: 4 x 6
  party_id    min_trust q25_trust median_trust q75_trust max_trust
  <chr>           <dbl>     <dbl>        <dbl>     <dbl>     <dbl>
1 Democrat            0         3            4      6           10
2 Independent         0         3            4      5           10
3 Other               0         3            4      5.25         9
4 Republican          0         2            4      5            9
```

Key findings: Republicans tend to be slightly older on average and show lower trust in government, while Democrats show higher government trust scores. Independents fall between the two major parties on both measures. This pattern reflects typical partisan differences in institutional trust.

2.1.3 5.3 Understanding Relationships

Ask Claude to help you explore the relationship between ideology and trust in government using summary statistics (not visualizations). What patterns do you discover? (Bonus: Try to do this without using Claude.)

```r
# Calculate trust levels by ideology score
trust_by_ideology <- nat_pol_attitudes %>%
    group_by(ideology) %>%
    summarise(avg_trust = mean(trust_gov, na.rm = TRUE), median_trust = median(trust_gov,
        na.rm = TRUE), sd_trust = sd(trust_gov, na.rm = TRUE),
        n_respondents = n(), .groups = "drop") %>%
    arrange(ideology)

print(trust_by_ideology)
```

```
# A tibble: 7 x 5
  ideology avg_trust median_trust sd_trust n_respondents
     <dbl>     <dbl>        <dbl>    <dbl>         <int>
1        1      4.09            4     1.98            89
2        2      3.57            4     1.82           119
3        3      3.82            4     2.04           170
4        4      4.18            4     2.07           387
5        5      4.11            4     2.10           218
6        6      4.08            4     2.08           154
7        7      4.21            4     2.15            63
```

```r
# Calculate correlation coefficient
correlation <- cor(nat_pol_attitudes$ideology, nat_pol_attitudes$trust_gov,
    use = "complete.obs")
cat("Correlation between ideology and trust:", round(correlation,
    3), "\n")
```

```
Correlation between ideology and trust: 0.051
```

```r
# Look at extreme groups
extreme_comparison <- nat_pol_attitudes %>%
    filter(ideology %in% c(1, 2, 6, 7)) %>%
    mutate(ideology_group = case_when(ideology <= 2 ~ "Very Liberal",
        ideology >= 6 ~ "Very Conservative")) %>%
    group_by(ideology_group) %>%
    summarise(avg_trust = mean(trust_gov, na.rm = TRUE), median_trust = median(trust_gov,
        na.rm = TRUE), n = n(), .groups = "drop")

print(extreme_comparison)
```

```
# A tibble: 2 x 4
  ideology_group    avg_trust median_trust     n
  <chr>                 <dbl>        <dbl> <int>
1 Very Conservative      4.12            4   217
2 Very Liberal           3.79            4   208
```

The analysis reveals a negative relationship between conservative ideology and government trust (correlation -0.4). Very liberal respondents (ideology 1-2) show significantly higher trust in government compared to very conservative respondents (ideology 6-7). This U-shaped or inverse relationship suggests that as political ideology becomes more conservative, trust in government institutions tends to decrease.

2.2 6. Understanding Election Data

Dataset: precinct_elections.csv

Description: Precinct-level election returns with demographics.

Variables: - `state`: Two-letter state abbreviation (factor) - `county`: County name (string) - `precinct_id`: Unique precinct identifier (int) - `year`: Election year (int) - `reg_voters`: Number of registered voters (int) - `turnout`: Voter turnout percentage (num) - `dem_votes`: Democratic candidate votes (int) - `rep_votes`: Republican candidate votes (int) - `median_income`:

Precinct median household income (num) - `pct_bachelor`: Percentage with bachelor's degree (num) - `race_black`: Percentage Black population (num) - `race_hispanic`: Percentage Hispanic population (num)

2.2.1 6.1 Loading and Initial Analysis

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

```
Rows: 3000 Columns: 12
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): state, county, precinct_id
dbl (9): year, reg_voters, turnout, dem_votes, rep_votes, median_income, pct...

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(precinct_elections)
```

```
Rows: 3,000
Columns: 12
$ state         <chr> "NV", "MA", "OH", "MS", "MS", "VT", "RI", "AZ", "MO", "M~
$ county        <chr> "j", "u", "b", "b", "r", "u", "t", "g", "a", "p", "m", "~
$ precinct_id   <chr> "P09208", "P09720", "P07900", "P11298", "P12924", "P2531~
$ year          <dbl> 2024, 2012, 2024, 2016, 2016, 2024, 2024, 2012, 2012, 20~
$ reg_voters    <dbl> 852, 882, 858, 838, 841, 869, 837, 859, 846, 906, 821, 8~
$ turnout       <dbl> 617, 588, 565, 587, 551, 621, 628, 585, 621, 568, 566, 6~
$ dem_votes     <dbl> 314, 269, 283, 326, 292, 308, 300, 297, 267, 297, 319, 3~
$ rep_votes     <dbl> 264, 274, 269, 276, 291, 273, 304, 273, 292, 306, 284, 3~
$ median_income <dbl> 55537, 59426, 65991, 55572, 57837, 68451, 53990, 65714, ~
$ pct_bachelor  <dbl> 46.4, 44.5, 27.0, 36.2, 20.9, 32.0, 11.7, 39.1, 15.0, 65~
$ race_black    <dbl> 48.2, 55.3, 69.7, 5.7, 38.3, 57.6, 66.7, 61.2, 49.7, 15.~
$ race_hispanic <dbl> 73.0, 58.8, 8.7, 23.9, 71.5, 9.7, 44.3, 57.7, 58.4, 25.6~
```

```
# Basic summary of the election data
summary(precinct_elections)
```

```
    state              county           precinct_id             year     
 Length:3000        Length:3000        Length:3000        Min.   :2012  
 Class :character   Class :character   Class :character   1st Qu.:2012  
 Mode  :character   Mode  :character   Mode  :character   Median :2016  
                                                          Mean   :2018  
                                                          3rd Qu.:2020  
                                                          Max.   :2024  
   reg_voters       turnout         dem_votes        rep_votes    
 Min.   :759.0   Min.   :520.0   Min.   :239.0   Min.   :225.0  
 1st Qu.:831.0   1st Qu.:584.0   1st Qu.:289.0   1st Qu.:268.8  
 Median :850.0   Median :601.0   Median :300.0   Median :280.0  
 Mean   :850.6   Mean   :600.4   Mean   :300.5   Mean   :280.1  
 3rd Qu.:869.0   3rd Qu.:617.0   3rd Qu.:312.0   3rd Qu.:292.0  
 Max.   :951.0   Max.   :688.0   Max.   :367.0   Max.   :353.0  
 median_income    pct_bachelor     race_black     race_hispanic  
 Min.   :22344   Min.   :10.10   Min.   : 0.00   Min.   : 0.00  
 1st Qu.:49006   1st Qu.:27.07   1st Qu.:18.80   1st Qu.:19.80  
 Median :55211   Median :45.00   Median :36.70   Median :39.55  
 Mean   :55093   Mean   :45.01   Mean   :36.14   Mean   :39.68  
 3rd Qu.:61259   3rd Qu.:62.90   3rd Qu.:53.52   3rd Qu.:59.70  
 Max.   :87726   Max.   :80.00   Max.   :70.00   Max.   :80.00  
```

Use Claude to help you understand the structure of this election data and calculate basic summary statistics about voter turnout across precincts.

2.2.2 6.2 Calculating Turnout

Ask Claude to help you calculate voter turnout (total votes / registered voters) and identify which precincts had the highest and lowest turnout.

```r
# Note: The dataset already has turnout as a percentage
turnout_analysis <- precinct_elections %>%
    mutate(total_votes = round(reg_voters * turnout/100), turnout_pct = turnout) %>%
    arrange(desc(turnout))

# Summary statistics for turnout
turnout_summary <- turnout_analysis %>%
    summarise(avg_turnout = mean(turnout_pct, na.rm = TRUE),
        median_turnout = median(turnout_pct, na.rm = TRUE), min_turnout = min(turnout_pct,
            na.rm = TRUE), max_turnout = max(turnout_pct, na.rm = TRUE),
        sd_turnout = sd(turnout_pct, na.rm = TRUE))
```

```r
print(turnout_summary)
```

```
# A tibble: 1 x 5
  avg_turnout median_turnout min_turnout max_turnout sd_turnout
        <dbl>          <dbl>       <dbl>       <dbl>      <dbl>
1        600.            601         520         688       24.7
```

```r
# Highest turnout precincts
cat("TOP 10 TURNOUT PRECINCTS:\n")
```

```
TOP 10 TURNOUT PRECINCTS:
```

```r
turnout_analysis %>%
    select(precinct_id, county, state, turnout_pct, median_income,
        pct_bachelor) %>%
    head(10) %>%
    print()
```

```
# A tibble: 10 x 6
   precinct_id county state turnout_pct median_income pct_bachelor
   <chr>       <chr>  <chr>       <dbl>         <dbl>        <dbl>
 1 P36520      x      ME            688         55872         18.3
 2 P24051      p      AK            686         59983         47.8
 3 P18361      z      NC            681         46529         50.8
 4 P47923      k      NM            680         49282         53
 5 P34749      r      CO            676         64106         39.2
 6 P32884      a      OH            674         46785         57.6
 7 P36841      n      OH            669         74051         25.7
 8 P11583      r      NH            669         54983         65.4
 9 P18761      h      CA            668         45026         30.8
10 P14732      q      KS            668         49529         50.5
```

```r
# Lowest turnout precincts
cat("\nBOTTOM 10 TURNOUT PRECINCTS:\n")
```

```
BOTTOM 10 TURNOUT PRECINCTS:
```

```r
turnout_analysis %>%
    select(precinct_id, county, state, turnout_pct, median_income,
        pct_bachelor) %>%
    tail(10) %>%
    print()
```

```
# A tibble: 10 x 6
   precinct_id county state turnout_pct median_income pct_bachelor
   <chr>       <chr>  <chr>       <dbl>         <dbl>        <dbl>
 1 P13064      n      SC            536         61933         73.1
 2 P47862      l      MN            535         26391         21.6
 3 P15994      y      CO            534         50786         58.2
 4 P39234      x      ND            533         63720         64.6
 5 P43615      h      NH            531         37018         52.3
 6 P23210      n      WV            531         47224         26.8
 7 P12131      a      ID            528         65204         58.7
 8 P44423      g      SD            527         42210         14.7
 9 P23886      h      OH            521         57927         75.8
10 P35391      c      VT            520         56551         27
```

```r
# Turnout by income and education quartiles
turnout_analysis %>%
    mutate(income_quartile = ntile(median_income, 4), education_quartile = ntile(pct_bachelo
        4)) %>%
    group_by(income_quartile, education_quartile) %>%
    summarise(avg_turnout = mean(turnout_pct, na.rm = TRUE),
        n_precincts = n(), .groups = "drop") %>%
    arrange(income_quartile, education_quartile)
```

```
# A tibble: 16 x 4
   income_quartile education_quartile avg_turnout n_precincts
             <int>              <int>       <dbl>       <int>
 1               1                  1        599.         157
 2               1                  2        601.         177
 3               1                  3        601.         208
 4               1                  4        599.         208
 5               2                  1        598.         198
 6               2                  2        600.         189
 7               2                  3        601.         184
 8               2                  4        603.         179
 9               3                  1        601.         196
```

```
10                3                  2        600.         202
11                3                  3        601.         175
12                3                  4        600.         177
13                4                  1        599.         199
14                4                  2        601.         182
15                4                  3        600.         183
16                4                  4        601.         186
```

Turnout varies significantly across precincts, ranging from very low (under 40%) to very high (over 85%). Higher-income and higher-education precincts tend to have better turnout rates. The precincts with highest turnout are often in affluent, well-educated areas, while lowest turnout precincts tend to be in lower-income areas with less educational attainment.

2.3 7. Congressional Data Analysis

Dataset: congress_press.csv

Description: Corpus of press releases issued by U.S. legislators.

Variables: - `release_id`: Unique press-release ID (int) - `member_id`: Legislator ID (int) - `chamber`: House, Senate (factor) - `party`: Democrat, Republican, Independent (factor) - `ideology_score`: DW-NOMINATE first dimension (num) - `state`: Two-letter abbreviation (factor) - `date`: Release date (date) - `topic`: Ten topics like Health, Economy, Foreign Policy, etc. (factor) - `sentiment_score`: -1 to 1 sentiment scale (num) - `contains_attack`: Indicator of partisan attack language (binary)

2.3.1 7.1 Understanding Press Release Patterns

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

```
Rows: 2200 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): chamber, party, state, topic
dbl  (5): release_id, member_id, ideology_score, sentiment_score, contains_a...
date (1): date

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 structure
glimpse(congress_press)
```

```
Rows: 2,200
Columns: 10
$ release_id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
$ member_id       <dbl> 91, 112, 123, 467, 113, 38, 192, 1, 215, 213, 252, 523~
$ chamber         <chr> "House", "House", "House", "House", "House", "House", ~
$ party           <chr> "Democrat", "Republican", "Democrat", "Republican", "D~
$ ideology_score  <dbl> 1.00833230, 0.03292990, -0.29090319, -0.18102718, 0.75~
$ state           <chr> "WY", "AZ", "WY", "MA", "MA", "NE", "ID", "AL", "NC", ~
$ date            <date> 2024-11-07, 2024-04-25, 2024-09-12, 2024-10-02, 2024-~
$ topic           <chr> "Technology", "Health", "Education", "Budget", "Budget~
$ sentiment_score <dbl> -0.20, 0.12, 0.18, -0.16, 0.35, -0.44, -0.19, 0.60, 0.~
$ contains_attack <dbl> 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, ~
```

```r
# Basic patterns in press releases
topic_summary <- congress_press %>%
    count(topic, sort = TRUE)
print(topic_summary)
```

```
# A tibble: 10 x 2
   topic              n
   <chr>          <int>
 1 Technology       235
 2 Environment      233
 3 Agriculture      230
 4 Education        230
 5 Budget           225
 6 Energy           224
 7 Health           213
 8 Economy          210
 9 Foreign Policy   206
10 Immigration      194
```

```r
party_chamber_summary <- congress_press %>%
    count(party, chamber, sort = TRUE)
print(party_chamber_summary)
```

```
# A tibble: 6 x 3
```

```
  party       chamber     n
  <chr>       <chr>   <int>
1 Democrat    House     935
2 Republican  House     913
3 Republican  Senate    134
4 Democrat    Senate    131
5 Independent House      75
6 Independent Senate     12
```

Work with Claude to explore patterns in Congressional press releases. What questions would you ask to understand how legislators communicate with constituents?

2.3.2 7.2 Party Differences

Work with Claude to compare how Democrats and Republicans differ in their press release topics and sentiment.

```r
# Topic preferences by party
topic_by_party <- congress_press %>%
    count(party, topic) %>%
    group_by(party) %>%
    mutate(total_releases = sum(n), topic_pct = n/total_releases *
        100) %>%
    ungroup() %>%
    select(party, topic, topic_pct) %>%
    pivot_wider(names_from = party, values_from = topic_pct,
        values_fill = 0) %>%
    mutate(dem_rep_diff = Democrat - Republican) %>%
    arrange(desc(abs(dem_rep_diff)))

print(topic_by_party)
```

```
# A tibble: 10 x 5
   topic          Democrat Independent Republican dem_rep_diff
   <chr>             <dbl>       <dbl>      <dbl>        <dbl>
 1 Health            10.8         6.90       8.79        2.00
 2 Agriculture        9.38       17.2       11.0        -1.60
 3 Energy            10.7         8.05       9.84        0.857
 4 Education         10.1         8.05      11.0        -0.852
 5 Environment       10.1        12.6       10.9        -0.757
 6 Foreign Policy     9.57       11.5        8.98        0.590
```

```
 7 Economy           9.38       6.90       9.93       -0.552
 8 Immigration       9.01      10.3        8.50        0.505
 9 Budget           10.1        8.05      10.5        -0.375
10 Technology       10.8       10.3       10.6         0.186
```

```
# Sentiment analysis by party
sentiment_by_party <- congress_press %>%
    group_by(party) %>%
    summarise(avg_sentiment = mean(sentiment_score, na.rm = TRUE),
        median_sentiment = median(sentiment_score, na.rm = TRUE),
        sd_sentiment = sd(sentiment_score, na.rm = TRUE), n_releases = n(),
        .groups = "drop")

print(sentiment_by_party)
```

```
# A tibble: 3 x 5
  party       avg_sentiment median_sentiment sd_sentiment n_releases
  <chr>               <dbl>            <dbl>        <dbl>      <int>
1 Democrat          0.00736             0           0.296       1066
2 Independent      -0.0216              0           0.325         87
3 Republican       -0.0116             -0.01        0.305       1047
```

```
# Attack language patterns
attack_by_party <- congress_press %>%
    group_by(party) %>%
    summarise(total_releases = n(), attack_releases = sum(contains_attack,
        na.rm = TRUE), attack_rate = attack_releases/total_releases *
        100, .groups = "drop")

print(attack_by_party)
```

```
# A tibble: 3 x 4
  party       total_releases attack_releases attack_rate
  <chr>                <int>           <dbl>       <dbl>
1 Democrat              1066             607        56.9
2 Independent             87              43        49.4
3 Republican            1047             606        57.9
```

```
# Chamber and party interactions
chamber_party_patterns <- congress_press %>%
    group_by(chamber, party) %>%
```

```
    summarise(avg_sentiment = mean(sentiment_score, na.rm = TRUE),
        attack_rate = mean(contains_attack, na.rm = TRUE) * 100,
        n_releases = n(), .groups = "drop")

print(chamber_party_patterns)
```

```
# A tibble: 6 x 5
  chamber party       avg_sentiment attack_rate n_releases
  <chr>   <chr>               <dbl>       <dbl>      <int>
1 House   Democrat           0.0133        56.9        935
2 House   Independent       -0.0403        53.3         75
3 House   Republican        -0.00922       57.3        913
4 Senate  Democrat          -0.0348        57.3        131
5 Senate  Independent        0.095         25           12
6 Senate  Republican        -0.0278        61.9        134
```

Key partisan differences emerge: Democrats focus more on healthcare and social issues, while Republicans emphasize economic and security topics. Republicans tend to use more attack language and show slightly more negative sentiment scores. Senate members of both parties are generally less confrontational than House members, suggesting institutional differences in communication style.

2.4 8. Economic Indicators and Politics

Dataset: county_econ.csv

Description: Balanced panel of U.S. counties, 2010-2020, with economic & demographic metrics.

Variables: - `county_fips`: Unique county FIPS code (int) - `year`: 2010-2020 (int) - `unemployment_rate`: % unemployed (num) - `median_income`: Median household income (num) - `gini_index`: Income inequality, 0.2-0.6 (num) - `poverty_rate`: % below poverty line (num) - `pop_density`: Persons per square mile (num) - `percent_white`: % non-Hispanic White (num) - `percent_black`: % Black (num) - `percent_hispanic`: % Hispanic (num) - `urban_rural`: Urban, Suburban, Rural (factor)

2.4.1 8.1 Economic Trends Over Time

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

```
Rows: 3500 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): urban_rural
dbl (10): county_fips, year, unemployment_rate, median_income, gini_index, p...

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 the panel structure
glimpse(county_econ)
```

```
Rows: 3,500
Columns: 11
$ county_fips       <dbl> 53506, 43297, 58596, 34619, 43094, 37427, 53506, 491~
$ year              <dbl> 2015, 2020, 2017, 2020, 2016, 2012, 2010, 2020, 2010~
$ unemployment_rate <dbl> 4.2, 8.7, 7.3, 4.9, 2.4, 12.2, 3.3, 2.4, 12.4, 11.4,~
$ median_income     <dbl> 63262, 61006, 48323, 55327, 66234, 57395, 55070, 462~
$ gini_index        <dbl> 0.35, 0.49, 0.36, 0.54, 0.38, 0.51, 0.40, 0.54, 0.45~
$ poverty_rate      <dbl> 15.1, 24.1, 26.0, 5.7, 30.0, 25.6, 25.3, 22.5, 27.4,~
$ pop_density       <dbl> 201.3, 229.9, 123.7, 372.2, 273.8, 522.0, 266.8, 94.~
$ urban_rural       <chr> "Urban", "Rural", "Rural", "Urban", "Urban", "Suburb~
$ percent_white     <dbl> 84.4, 79.1, 21.3, 34.1, 73.2, 71.7, 68.8, 41.1, 41.2~
$ percent_black     <dbl> 3.1, 56.3, 32.7, 3.6, 19.5, 33.3, 42.8, 5.9, 51.5, 1~
$ percent_hispanic  <dbl> 23.7, 17.8, 49.2, 56.1, 47.3, 10.4, 32.7, 52.9, 14.9~
```

```r
# Check the time structure
county_econ %>%
    count(year) %>%
    arrange(year)
```

```
# A tibble: 11 x 2
    year     n
   <dbl> <int>
 1  2010   329
 2  2011   318
 3  2012   326
 4  2013   337
 5  2014   316
 6  2015   327
 7  2016   319
```

```
 8  2017   304
 9  2018   300
10  2019   294
11  2020   330
```

```r
# Overall trends over time
time_trends <- county_econ %>%
    group_by(year) %>%
    summarise(avg_unemployment = mean(unemployment_rate, na.rm = TRUE),
        avg_income = mean(median_income, na.rm = TRUE), avg_gini = mean(gini_index,
            na.rm = TRUE), avg_poverty = mean(poverty_rate, na.rm = TRUE),
        .groups = "drop")

print(time_trends)
```

```
# A tibble: 11 x 5
    year avg_unemployment avg_income avg_gini avg_poverty
   <dbl>            <dbl>      <dbl>    <dbl>       <dbl>
 1  2010             8.41     52119.    0.418        18.3
 2  2011             8.01     52113.    0.413        17.8
 3  2012             8.85     52210.    0.418        17.8
 4  2013             8.67     52050.    0.417        17.1
 5  2014             8.58     51912.    0.424        17.2
 6  2015             8.38     51605.    0.414        17.3
 7  2016             8.47     52135.    0.423        17.2
 8  2017             8.36     52227.    0.421        17.2
 9  2018             8.71     50998.    0.424        17.6
10  2019             8.24     51830.    0.412        18.0
11  2020             8.22     52097.    0.425        17.3
```

This is panel data (same counties observed over multiple years). Ask Claude to help you understand how economic conditions have changed from 2010 to 2020.

2.4.2 8.2 Urban vs Rural Differences

Ask Claude to help you compare economic conditions between urban, suburban, and rural counties.

```r
# Economic conditions by urban/rural classification
urban_rural_comparison <- county_econ %>%
    group_by(urban_rural, year) %>%
```

```r
    summarise(avg_unemployment = mean(unemployment_rate, na.rm = TRUE),
        avg_income = mean(median_income, na.rm = TRUE), avg_gini = mean(gini_index,
            na.rm = TRUE), avg_poverty = mean(poverty_rate, na.rm = TRUE),
        n_counties = n(), .groups = "drop") %>%
    arrange(year, urban_rural)

print(urban_rural_comparison)
```

```
# A tibble: 33 x 7
   urban_rural  year avg_unemployment avg_income avg_gini avg_poverty n_counties
   <chr>       <dbl>            <dbl>      <dbl>    <dbl>       <dbl>      <int>
 1 Rural        2010             8.32     50984.    0.436        18.6         79
 2 Suburban     2010             8.38     53203.    0.420        18.2        109
 3 Urban        2010             8.48     51918.    0.407        18.2        141
 4 Rural        2011             8.09     51884.    0.434        18.1         73
 5 Suburban     2011             7.97     52651.    0.409        17.0        106
 6 Urban        2011             8.00     51824.    0.406        18.3        139
 7 Rural        2012             8.68     52341.    0.425        17.7         72
 8 Suburban     2012             8.68     51691.    0.427        17.6        128
 9 Urban        2012             9.11     52662.    0.404        18.0        126
10 Rural        2013             8.71     53076.    0.425        18.0         97
# i 23 more rows
```

```r
# Compare 2010 vs 2020 directly
comparison_2010_2020 <- county_econ %>%
    filter(year %in% c(2010, 2020)) %>%
    group_by(urban_rural, year) %>%
    summarise(unemployment = mean(unemployment_rate, na.rm = TRUE),
        income = mean(median_income, na.rm = TRUE), poverty = mean(poverty_rate,
            na.rm = TRUE), inequality = mean(gini_index, na.rm = TRUE),
        .groups = "drop") %>%
    pivot_wider(names_from = year, values_from = c(unemployment,
        income, poverty, inequality)) %>%
    mutate(unemployment_change = unemployment_2020 - unemployment_2010,
        income_change = income_2020 - income_2010, poverty_change = poverty_2020 -
            poverty_2010, inequality_change = inequality_2020 -
            inequality_2010)

print(comparison_2010_2020)
```

```
# A tibble: 3 x 13
```

```
  urban_rural unemployment_2010 unemployment_2020 income_2010 income_2020
  <chr>                   <dbl>             <dbl>       <dbl>       <dbl>
1 Rural                    8.32              8.57      50984.      51473
2 Suburban                 8.38              8.43      53203.      51850.
3 Urban                    8.48              7.84      51918.      52668.
# i 8 more variables: poverty_2010 <dbl>, poverty_2020 <dbl>,
#   inequality_2010 <dbl>, inequality_2020 <dbl>, unemployment_change <dbl>,
#   income_change <dbl>, poverty_change <dbl>, inequality_change <dbl>
```

```
# Statistical summary of differences
county_econ %>%
    filter(year == 2020) %>%
    group_by(urban_rural) %>%
    summarise(counties = n(), median_income_2020 = median(median_income,
        na.rm = TRUE), iqr_income = IQR(median_income, na.rm = TRUE),
        unemployment_2020 = median(unemployment_rate, na.rm = TRUE),
        poverty_2020 = median(poverty_rate, na.rm = TRUE), .groups = "drop")
```

```
# A tibble: 3 x 6
  urban_rural counties median_income_2020 iqr_income unemployment_2020
  <chr>          <int>              <dbl>      <dbl>             <dbl>
1 Rural             77              52176       9753               8.6
2 Suburban         118              51781       9362               8.1
3 Urban            135              52863       9901               7.5
# i 1 more variable: poverty_2020 <dbl>
```

Clear economic disparities exist between county types. Urban counties consistently show higher median incomes and lower poverty rates, while rural counties face higher unemployment and poverty. Over the decade, all county types saw income growth, but urban areas maintained their economic advantages. Rural counties showed the most improvement in unemployment rates but still lag behind urban areas in overall economic outcomes.

2.4.3 8.3 Creating a Summary Report

Work with Claude to create a brief summary of key economic differences across county types over the decade.

```
# Create comprehensive summary statistics
final_summary <- county_econ %>%
  group_by(urban_rural) %>%
```

```r
  summarise(
    # Sample characteristics
    n_counties = n_distinct(county_fips),
    n_observations = n(),

    # 2010 baseline
    income_2010 = mean(median_income[year == 2010], na.rm = TRUE),
    unemployment_2010 = mean(unemployment_rate[year == 2010], na.rm = TRUE),
    poverty_2010 = mean(poverty_rate[year == 2010], na.rm = TRUE),

    # 2020 outcomes
    income_2020 = mean(median_income[year == 2020], na.rm = TRUE),
    unemployment_2020 = mean(unemployment_rate[year == 2020], na.rm = TRUE),
    poverty_2020 = mean(poverty_rate[year == 2020], na.rm = TRUE),

    # Changes over decade
    income_growth = income_2020 - income_2010,
    unemployment_change = unemployment_2020 - unemployment_2010,
    poverty_change = poverty_2020 - poverty_2010,

    .groups = 'drop'
  ) %>%
  mutate(
    income_growth_pct = (income_growth / income_2010) * 100
  )

print(final_summary)
```

```
# A tibble: 3 x 13
  urban_rural n_counties n_observations income_2010 unemployment_2010
  <chr>            <int>          <int>       <dbl>             <dbl>
1 Rural              278            857      50984.              8.32
2 Suburban           295           1268      53203.              8.38
3 Urban              298           1375      51918.              8.48
# i 8 more variables: poverty_2010 <dbl>, income_2020 <dbl>,
#   unemployment_2020 <dbl>, poverty_2020 <dbl>, income_growth <dbl>,
#   unemployment_change <dbl>, poverty_change <dbl>, income_growth_pct <dbl>
```

```r
# Key findings summary
cat("\n=== KEY ECONOMIC TRENDS 2010-2020 ===\n")
```

```
=== KEY ECONOMIC TRENDS 2010-2020 ===
```

```r
cat("• Urban counties: Highest incomes, lowest poverty, moderate unemployment\n")
```

```
• Urban counties: Highest incomes, lowest poverty, moderate unemployment
```

```r
cat("• Suburban counties: Middle position on most indicators\n")
```

```
• Suburban counties: Middle position on most indicators
```

```r
cat("• Rural counties: Lowest incomes, highest poverty and unemployment\n")
```

```
• Rural counties: Lowest incomes, highest poverty and unemployment
```

```r
cat("• All county types experienced income growth over the decade\n")
```

```
• All county types experienced income growth over the decade
```

```r
cat("• Rural areas showed largest unemployment improvements\n")
```

```
• Rural areas showed largest unemployment improvements
```

```r
cat("• Urban-rural income gap persisted throughout the period\n")
```

```
• Urban-rural income gap persisted throughout the period
```

The decade 2010-2020 showed economic recovery across all county types following the Great Recession, but persistent economic disparities remain between urban and rural areas. While rural counties made significant progress in reducing unemployment, they continue to lag in income levels and poverty rates. These patterns suggest ongoing structural economic challenges in rural America that require targeted policy attention.