Week 2, Class 4

Summary Statistics

Sean Westwood

In Today's Class

- Central tendency: mean, median, mode
- When to use each measure of center
- Understanding skewness and outliers
- Measures of spread: range, variance, standard deviation
- Using `group_by()` and `summarise()` for grouped analysis

Summary Statistics

Why Do We Need Summary Statistics?

The Problem: Raw data with thousands of observations is overwhelming

The Solution: Summary statistics reduce complexity while preserving key information

Goal: Describe the "typical" or "central" value in our data

Congressional Approval Example

Congressional approval helps us understand public trust in political institutions

Dataset Description:

- `congress_approval`: Approval rating for Congress (0-100 scale)
- `party_id`: Respondent's party affiliation (Democrat, Republican, Independent)
- `age`: Respondent's age
- `education`: Education level (High School, Some College, Bachelor's, etc.)
- `income_category`: Income bracket ($30k-$60k, etc.)
- `region`: Geographic region (Midwest, South, etc.)

```
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.
```

```
# A tibble: 10 x 7
   respondent_id   age education    party_id    income_category region
           <dbl> <dbl> <chr>        <chr>       <chr>           <chr>
 1             1    31 Some College Independent $30k-$60k       Midwest
 2             2    68 Bachelor's   Republican  $30k-$60k       South
 3             3    41 Some College Republican  $30k-$60k       South
 4             4    75 Bachelor's   Democrat    $30k-$60k       South
 5             5    19 Some College Democrat    $60k-$100k      West
 6             6    55 Some College Republican  $30k-$60k       South
 7             7    27 Some College Democrat    $60k-$100k      Midwest
 8             8    49 High School  Republican  Under $30k      West
 9             9    57 Bachelor's   Republican  Under $30k      Midwest
10            10    56 High School  Democrat    Over $100k      South
# i 1 more variable: congress_approval <dbl>
```

The Mean (Average)

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

What this notation means:

- $\bar{x}$ (x-bar): The sample mean
- $\sum_{i=1}^{n}$: Sum from the first observation (i=1) to the last (i=n)
- x_i: Each individual value in our dataset
- n: Total number of observations

```r
# Calculate mean approval rating
mean_approval <- mean(approval$congress_approval, na.rm = TRUE)
mean_approval
```

```
[1] 28.70509
```

Interpretation: On average, approval was 28.7%

The Median (Middle Value)

Definition: The value that splits the data in half

```
# Calculate median approval rating
median_approval <- median(approval$congress_approval, na.rm = TRUE)
median_approval
```

```
[1] 28.50776
```

The Mode (Most Common Value)

Definition: The value that appears most frequently

```
# Find mode using count
approval %>%
  count(congress_approval, sort = TRUE) %>%
  head(5)
```

```
# A tibble: 5 x 2
  congress_approval     n
              <dbl> <int>
1             0        35
2             0.238     1
3             0.526     1
4             0.808     1
5             0.831     1
```

Central Tendency

What is Central Tendency?

Definition: Central tendency describes where the "center" or "typical" value of a dataset lies.

Three main measures:

1. **Mean** (average): Mathematical center
2. **Median** (middle): Positional center

3. **Mode** (most common): Most frequent value

Key Insight: Different measures of central tendency can tell different stories about the same data, especially when distributions are skewed or have outliers

When Distributions Aren't Symmetric

Real data often has:

- **Outliers**: Extreme values that don't fit the typical pattern
- **Skewness**: Data stretched more in one direction than the other
- **Multiple modes**: More than one common value

Why this matters: Different shapes require different approaches to finding the "typical" value

Understanding Outliers

Definition: Data points that are unusually far from other observations

Examples in political data:

- A presidential approval rating of 90% during a crisis
- A candidate spending $500 million in a typical House race
- A voter turnout of 95% in a large precinct

Impact on measures:

- **Mean**: Very sensitive to outliers (gets "pulled" toward them)
- **Median**: Resistant to outliers (stays stable)
- **Mode**: Usually unaffected by outliers

Understanding Skewness

Left-skewed (negative skew):

- Long tail extends toward lower values
- Mean < Median
- Example: Test scores when most students do well

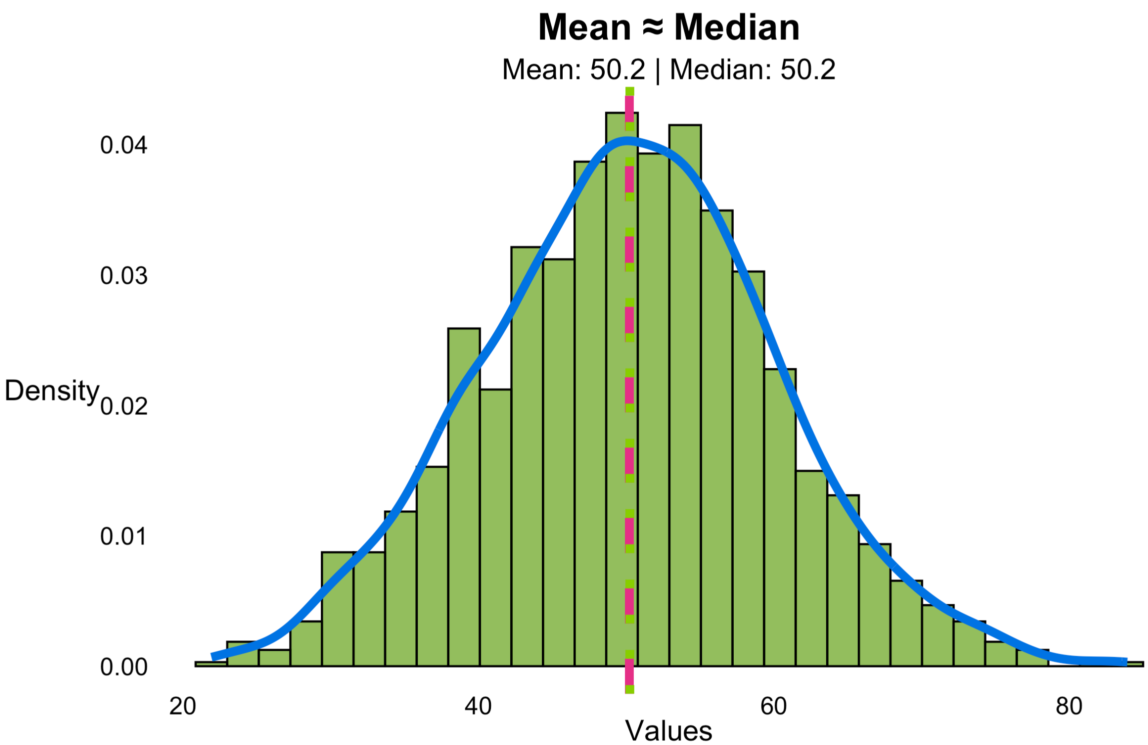
Right-skewed (positive skew):

- Long tail extends toward higher values

- Mean > Median
- Example: Income data (few very wealthy people)

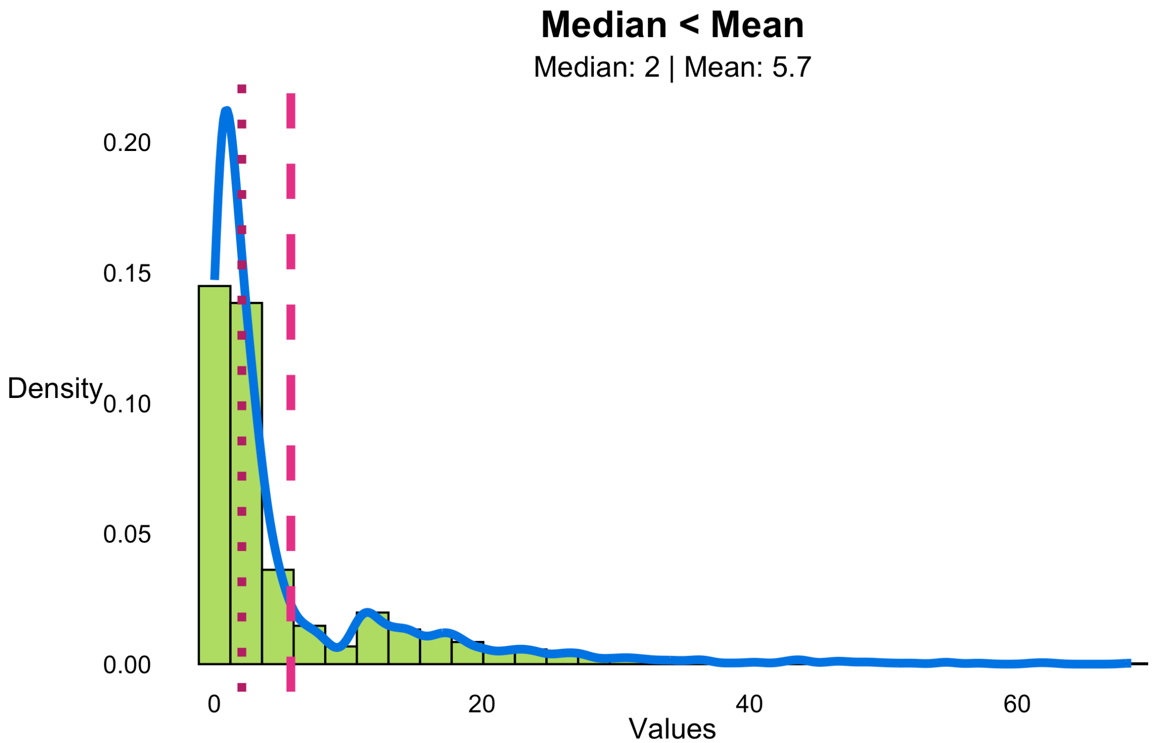
Campaign spending, wealth, and many political variables are typically right-skewed

Symmetric Distribution



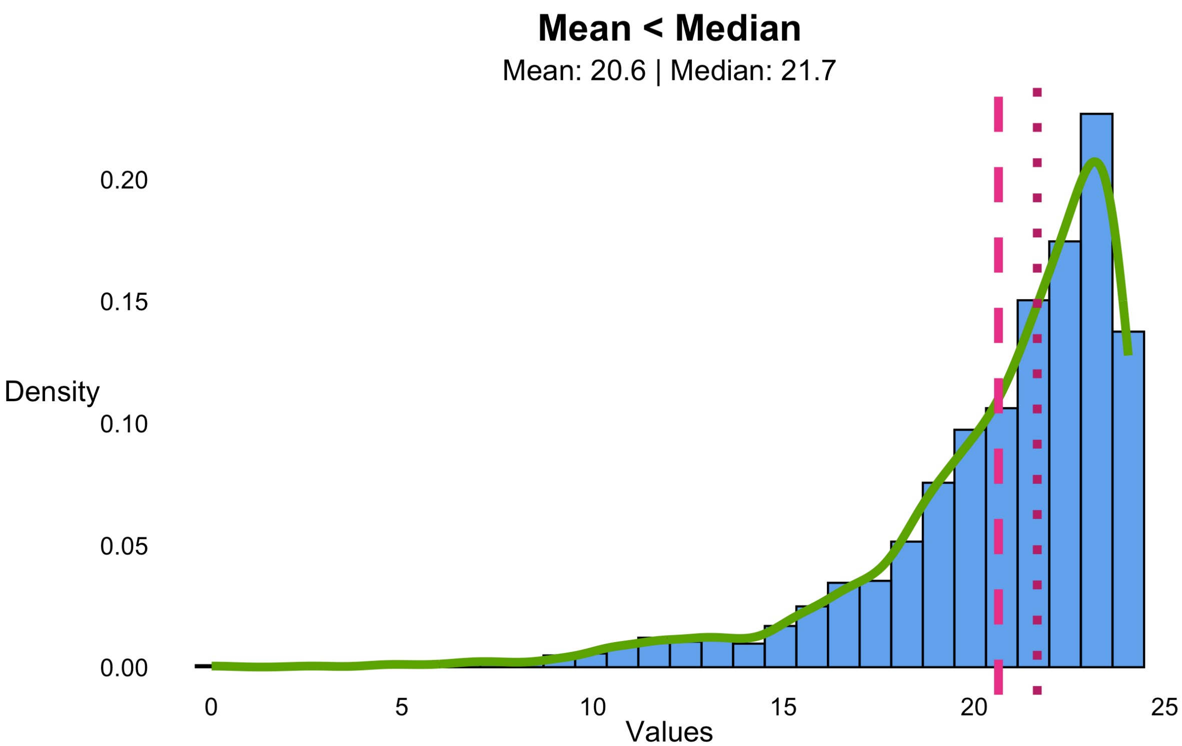
All three measures are similar

Right-Skewed Distribution



Mean pulled toward high values

Left-Skewed Distribution



Mean pulled toward low values

Example: Income Distribution

Why This Matters

```
# Calculate all three measures
mean_income <- mean(income)
median_income <- median(income)
mode_income <- income[which.max(tabulate(match(income, unique(income))))]

print(paste("Mean:", round(mean_income, 0)))
```

```
[1] "Mean: 83264"
```

```r
print(paste("Median:", round(median_income, 0)))
```

```
[1] "Median: 54176"
```

Notice: Mean is much higher than median due to wealthy outliers

Historical Context: Adolphe Quetelet

The “Average Man” (1835)

Adolphe Quetelet: Belgian statistician who pioneered the use of statistics in social science

Wanted to understand the "average man" (l'homme moyen) and developed anthropometry and BMI

- **Measured** physical characteristics of soldiers
- **Calculated** average height, weight, chest measurements, etc.
- **Identified** Human physical traits follow predictable patterns (distributions)

Unfortunately, he was a racist and used his work to justify eugenics

Measures of Spread

Why Central Tendency Isn't Enough

Consider two datasets with the same mean:

- **Midterm Dataset A**: 48, 49, 50, 51, 52 (mean = 50)
- **Midterm Dataset B**: 10, 30, 50, 70, 90 (mean = 50)

Question: Are these datasets the same?

Means are not enough! We need measures of **spread** or **variability**.

Range

Definition: Difference between maximum and minimum values

```
# Calculate range for approval ratings
approval %>%
  summarise(
    min_approval = min(congress_approval, na.rm = TRUE),
    max_approval = max(congress_approval, na.rm = TRUE),
    range = max_approval - min_approval
  )
```

```
# A tibble: 1 x 3
  min_approval max_approval range
         <dbl>        <dbl> <dbl>
1            0         75.9  75.9
```

Limitation: Sensitive to outliers, ignores distribution shape

Variance and Standard Deviation

Variance:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

Standard Deviation:

$$s = \sqrt{s^2}$$

Understanding the notation:

- s^2 = sample variance (s-squared)
- s = sample standard deviation
- x_i = each individual observation (i = 1, 2, 3, … n)
- $\bar{x}$ = sample mean (x-bar)
- n = sample size

Note: We use s and s^2 for **sample** statistics. Population parameters use Greek letters: σ (sigma) for population standard deviation and σ^2 for population variance. Since we almost always work with samples, we use s and s^2.

Understanding Variance and Standard Deviation

What do they measure?

- **Variance**: Average of squared distances from the mean
- **Standard Deviation**: Typical distance observations are from the mean
- **Both measure "spread"** - how much data points vary around the center

Step-by-Step Calculation Example

Step 1: Find the mean

Step 2: Calculate deviations from mean

Value	Mean	Deviation $(x_i - \bar{x})$	Squared Deviation $(x_i - \bar{x})^2$
45	50	-5	25
48	50	-2	4
50	50	0	0

Value	Mean	Deviation $(x_i - \bar{x})$	Squared Deviation $(x_i - \bar{x})^2$
52	50	2	4
55	50	5	25

Step 3: Sum the squared deviations

$$\sum (x_i - \bar{x})^2 = 25 + 4 + 0 + 4 + 25 = 58$$

Step 4: Calculate variance

$$s = \sqrt{\frac{58}{5-1}} = \sqrt{\frac{58}{4}} = 3.81$$

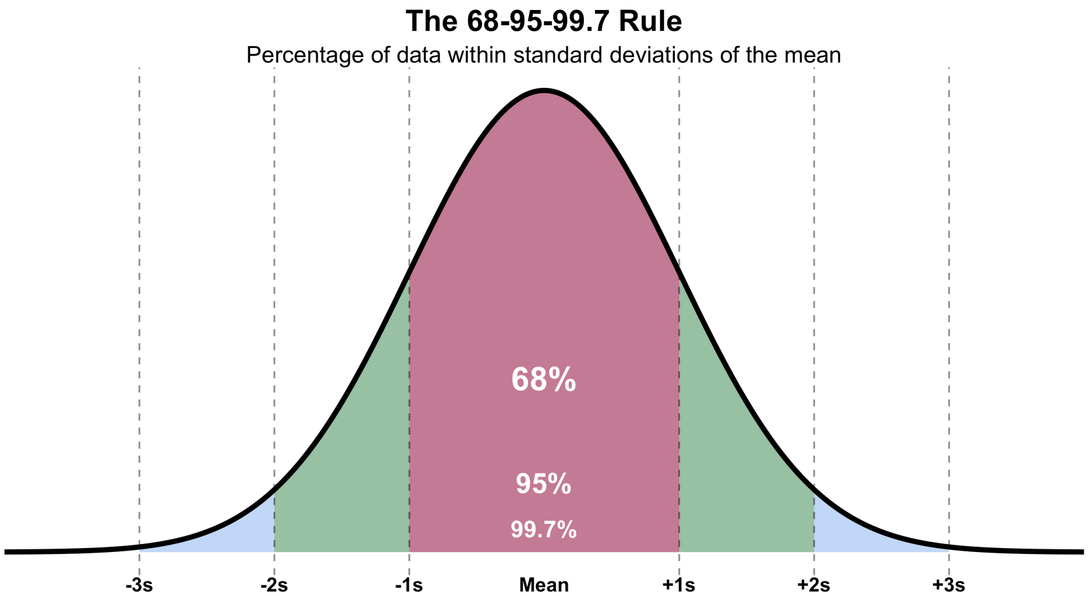
Variance and Standard Deviation: With R

```
# Calculate variance and standard deviation
approval %>%
  summarise(
    variance = var(congress_approval, na.rm = TRUE),
    std_dev = sd(congress_approval, na.rm = TRUE),
    mean = mean(congress_approval, na.rm = TRUE)
  )
```

```
# A tibble: 1 x 3
  variance std_dev  mean
     <dbl>   <dbl> <dbl>
1     183.    13.5  28.7
```

The 68-95-99.7 Rule (Empirical Rule)

For approximately normal (bell-shaped) distributions:



Interpretation: If mean approval = 40% and SD = 10%, then about 68% of observations fall between 30% and 50%, and 95% fall between 20% and 60%.

Data Analysis with summarise() and group_by()

The `summarise()` Function

Purpose: Create summary statistics from your data

Components of `summarise()`:

- **Input**: A data frame
- **Output**: A single row with your calculated statistics
- **Functions**: Any function that returns a single value (mean, median, sd, n, etc.)

Basic `summarise()` Example

```r
# Load congressional data
congress <- read_csv("../../data/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
# Single summary of the entire dataset
congress %>%
  summarise(
    total_members = n(),                                    # Count of rows
    avg_ideology = mean(nominate_dim1, na.rm = TRUE),      # Mean
    median_ideology = median(nominate_dim1, na.rm = TRUE), # Median
    spread_ideology = sd(nominate_dim1, na.rm = TRUE),     # Standard deviation
    min_ideology = min(nominate_dim1, na.rm = TRUE),       # Minimum
    max_ideology = max(nominate_dim1, na.rm = TRUE)        # Maximum
  )
```

```
# A tibble: 1 x 6
  total_members avg_ideology median_ideology spread_ideology min_ideology
          <int>        <dbl>           <dbl>           <dbl>        <dbl>
1         51044      0.00727           -0.04           0.377           -1
# i 1 more variable: max_ideology <dbl>
```

Key insight: `summarise()` reduces your entire dataset to a single row of summary statistics

Adding `group_by()` for Subgroup Analysis

But what if we want to compare the mean ideology of Republicans vs. Democrats?

`group_by()`: Apply `summarise()` to subgroups instead of the entire dataset

```r
# Same summary, but BY party
congress %>%
  group_by(party_code) %>%
  summarise(
    count = n(),
    avg_nominate_dim1 = mean(nominate_dim1, na.rm = TRUE),
    median_nominate_dim1 = median(nominate_dim1, na.rm = TRUE),
    sd_nominate_dim1 = sd(nominate_dim1, na.rm = TRUE)
  )
```

```
# A tibble: 53 x 5
   party_code count avg_nominate_dim1 median_nominate_dim1 sd_nominate_dim1
   <chr>      <int>             <dbl>                <dbl>            <dbl>
 1 1            847            0.539                0.585            0.242
 2 1060          11            0.0829               0.204            0.207
 3 108            8           -0.0103              -0.002            0.188
 4 1111           1            0.907                0.907           NA
 5 1116           1            0.068                0.068           NA
 6 112           13           -0.039                0.02             0.377
 7 114            9            0.298                0.361            0.141
 8 117            2           -0.00900             -0.00900          0.255
 9 1275         395            0.240                0.286            0.241
10 13          1976           -0.0615              -0.077            0.275
# i 43 more rows
```

Grouping Multiple Variables

You can group by multiple variables to create more detailed breakdowns:

```
# Summary by party AND chamber
congress %>%
  group_by(party_code, chamber) %>%
  summarise(
    count = n(),
    avg_nominate_dim1 = mean(nominate_dim1, na.rm = TRUE),
    .groups = "drop"  # Removes grouping after summarise
  )
```

```
# A tibble: 93 x 4
   party_code chamber   count avg_nominate_dim1
   <chr>      <chr>     <int>             <dbl>
 1 1          House       648            0.521
 2 1          President     3          NaN
 3 1          Senate      196            0.598
 4 1060       House         4           -0.177
 5 1060       Senate        7            0.231
 6 108        House         8           -0.0103
 7 1111       Senate        1            0.907
 8 1116       House         1            0.068
 9 112        House        10           -0.200
10 112        Senate        3            0.499
# i 83 more rows
```

Note: `.groups = "drop"` removes grouping after `summarise()` to avoid unexpected behavior

Comprehensive Statistical Summaries

Create complete statistical profiles for each group:

```r
# Complete statistical summary by party
congress %>%
  group_by(party_code) %>%
  summarise(
    count = n(),                                         # Sample size
    mean_ideology = mean(nominate_dim1, na.rm = TRUE), # Central tendency
    median_ideology = median(nominate_dim1, na.rm = TRUE),
    min_ideology = min(nominate_dim1, na.rm = TRUE),   # Range
    max_ideology = max(nominate_dim1, na.rm = TRUE),
    std_dev = sd(nominate_dim1, na.rm = TRUE),         # Spread
    .groups = "drop"
  ) %>%
  mutate(across(where(is.numeric), round, 3))  # Round for readability
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `across(where(is.numeric), round, 3)`.
Caused by warning:
! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
Supply arguments directly to `.fns` through an anonymous function instead.

  # Previously
  across(a:b, mean, na.rm = TRUE)

  # Now
  across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
# A tibble: 53 x 7
   party_code count mean_ideology median_ideology min_ideology max_ideology
   <chr>      <dbl>         <dbl>           <dbl>        <dbl>        <dbl>
 1 1            847         0.539           0.585       -0.921        0.998
 2 1060          11         0.083           0.204       -0.177        0.252
 3 108            8        -0.01           -0.002       -0.261        0.232
 4 1111           1         0.907           0.907        0.907        0.907
 5 1116           1         0.068           0.068        0.068        0.068
```

```
 6 112            13        -0.039          0.02         -0.559        0.499
 7 114             9         0.298          0.361         0.011        0.406
 8 117             2        -0.009         -0.009        -0.189        0.171
 9 1275          395         0.24           0.286        -0.538        0.682
10 13           1976        -0.061         -0.077        -0.996        0.919
# i 43 more rows
# i 1 more variable: std_dev <dbl>
```

The count() Function

Counting Observations

```r
# Count with conditions
congress %>%
  filter(nominate_dim1 > 0.6) %>%
  count(party_code, sort = TRUE)
```

```
# A tibble: 16 x 2
   party_code     n
   <chr>      <int>
 1 Republican  1347
 2 1            395
 3 5000          72
 4 13            29
 5 29            24
 6 1275          20
 7 22            17
 8 8888          13
 9 7777           5
10 Democrat       5
11 300            3
12 8000           2
13 1111           1
14 1346           1
15 26             1
16 331            1
```

AI Integration for Statistical Analysis

Effective Prompts for Summary Statistics

For choosing the right measure:

> "I have presidential approval rating data that might have outliers. Should I use mean or median to summarize it? Please explain the difference and provide R code for both."

For grouping analysis:

> "Help me write tidyverse code to calculate mean, median, and standard deviation of vote_share, grouped by party_code and state_abbrev. Explain what you did. My dataframe is called congress and it looks like this: <insert glimpse()>"

Interpreting Results with AI

For understanding patterns:

> "I calculated that Republican candidates have a mean vote share of 0.52 and Democrats have 0.48, with standard deviations of 0.15 and 0.18 respectively. What does this tell me about voting patterns?"

AI helps you understand:

- What the numbers mean in context
- Whether differences are meaningful
- What questions to ask next

Common Mistakes and Solutions

Forgetting to Handle Missing Values

```r
# This might give NA if there are missing values
test_data <- c(1, 2, 3, NA, 5)
mean(test_data)  # Returns NA
```

```
[1] NA
```

```r
# Solution: use na.rm = TRUE
mean(test_data, na.rm = TRUE)  # Returns 2.75
```

```
[1] 2.75
```

Choosing the Wrong Measure

Use Mean When:

- Data is roughly symmetric
- You want to include all values
- Making predictions

Use Median When:

- Data has outliers
- Data is skewed
- Describing "typical" experience

Best Practices

Report Multiple Measures

```r
# Comprehensive summary
approval %>%
  summarise(
    n = n(),
    mean = mean(congress_approval, na.rm = TRUE),
    median = median(congress_approval, na.rm = TRUE),
    sd = sd(congress_approval, na.rm = TRUE),
    min = min(congress_approval, na.rm = TRUE),
    max = max(congress_approval, na.rm = TRUE)
  ) %>%
  mutate(across(where(is.numeric), round, 2))
```

```
# A tibble: 1 x 6
      n  mean median    sd   min   max
  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
1  2000  28.7   28.5  13.5     0  75.9
```

Think About Context

Numbers without context are meaningless

- Is a 5-point difference in approval ratings large?
- What's a typical range for vote shares?
- How do current values compare to historical patterns?

Measuring Relationships: Correlation

Correlation: Do Two Variables Move Together?

Correlation coefficient (r): Measures the strength and direction of a linear relationship between two variables

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1) \cdot s_x \cdot s_y}$$

Key properties:

- Ranges from **-1 to +1**
- **r = +1**: Perfect positive relationship (as X increases, Y increases)
- **r = -1**: Perfect negative relationship (as X increases, Y decreases)
- **r = 0**: No linear relationship

Interpreting Correlation Strength

Absolute Value of r	Interpretation
0.00 - 0.19	Very weak
0.20 - 0.39	Weak
0.40 - 0.59	Moderate
0.60 - 0.79	Strong
0.80 - 1.00	Very strong

Important: Correlation measures only **linear** relationships. Two variables can be strongly related in a non-linear way but have r 0!

Correlation in R

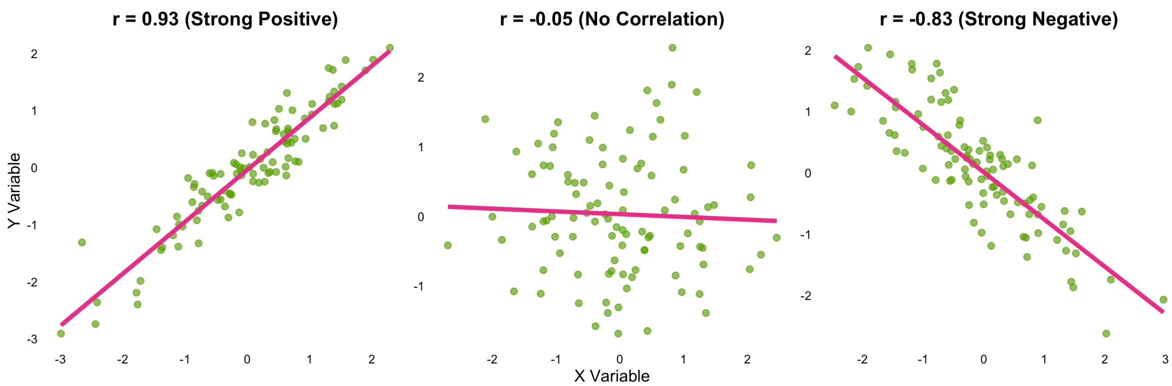
```r
# Calculate correlation between age and approval
cor(approval$age, approval$congress_approval, use = "complete.obs")
```

```
[1] 0.1305263
```

Interpretation: The correlation between age and congressional approval is weak, suggesting age is not strongly associated with approval ratings in this sample.

Visualizing Correlations

```
`geom_smooth()` using formula = 'y ~ x'
```



Correlation Causation

Critical Warning: Finding a correlation between X and Y does NOT mean X causes Y!

Possible explanations for correlation:

1. X causes Y
2. Y causes X (reverse causation)
3. Some third variable Z causes both (confounding)
4. Pure coincidence

Example: Ice cream sales and drowning deaths are positively correlated. Does ice cream cause drowning? No! Hot weather (Z) causes both.

Looking Ahead

In Our Next Class

Transforming and Creating Variables

- Creating new variables with `mutate()`
- Conditional logic: comparison operators
- Using `if_else()` for binary outcomes
- Using `case_when()` for multiple categories

Key Concepts to Remember

- **Mean** includes all values but sensitive to outliers
- **Median** resistant to outliers, good for skewed data

- **Standard deviation** measures spread around the mean
- **group_by() + summarise()** powerful for comparing groups
- **Context matters** - interpret statistics in real-world terms

Questions?

Key takeaway: Summary statistics are tools for understanding data patterns. Choose the right tool for your data and always interpret results in context.

Next class: We'll learn about different research designs and when we can make causal claims from data.