Week 3, Class 5: Practice Exercises

Data Transformation and Variable Creation

2024-12-31

1 Non-AI Exercises

1.1 1. Understanding mutate()

1.1.1 1.1 Multiple Choice: mutate() Function

What does the `mutate()` function do?

a) Removes columns from a data frame
b) Creates new columns or modifies existing ones
c) Filters rows based on conditions
d) Sorts data by a variable

Answer: _____

1.1.2 1.2 Code Detective: Basic mutate()

What does this code create?

```
data %>%
  mutate(
    vote_margin = dem_votes - rep_votes,
    winner = if_else(vote_margin > 0, "Democrat", "Republican")
  )
```

Line 3 creates: _____ Line 4 creates: _____

1.1.3 1.3 Fill in the Blanks: Variable Types

When creating new variables, we often need to:

1. Calculate _____ between existing variables
2. _____ text variables into categories
3. Create _____ (TRUE/FALSE) indicators
4. Convert between _____ types
5. Handle _____ values appropriately

Word bank: differences, recode, logical, data, missing

1.2 2. Conditional Logic

1.2.1 2.1 Match: if_else() vs case_when()

Match each function with when to use it:

Functions: a) if_else() b) case_when()

Use cases: 1. Creating a variable with only two possible outcomes 2. Creating a variable with multiple categories 3. Simple yes/no binary coding 4. Complex multi-condition logic

Matches: a = _____ and _____, b = _____ and _____

1.2.2 2.2 Code Detective: case_when()

What categories does this code create?

```
data %>%
  mutate(
    age_group = case_when(
      age < 30 ~ "Young",
      age < 50 ~ "Middle",
      age < 65 ~ "Older",
      TRUE ~ "Senior"
    )
  )
```

For someone age 25: _____ For someone age 45: _____ For someone age 70: _____

1.2.3 2.3 Spot the Error

What's wrong with this case_when() statement?

```
mutate(
  income_cat = case_when(
    income < 30000 ~ "Low",
    income < 50000 ~ "Medium",
    income < 30000 ~ "Low",
    TRUE ~ "High"
  )
)
```

Problem: _____

1.3 3. Working with Proportions

1.3.1 3.1 Multiple Choice: Calculating Proportions

To calculate the proportion of Democrats in a dataset, you would:

a) Count Democrats and divide by Republicans
b) Count Democrats and divide by total observations
c) Count total and divide by Democrats
d) Use mean() on a logical variable

Answer: _____

1.3.2 3.2 True or False: summarise()

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

_____ summarise() reduces multiple rows to a single summary row _____ You can calculate multiple statistics in one summarise() call _____ summarise() automatically groups by all variables _____ n() counts the number of rows in each group _____ summarise() can only calculate numeric summaries

1.3.3 3.3 Fill in the Code

Complete this code to calculate turnout rate:

```
data %>%
  summarise(
    total_voters = _____,
    total_voted = sum(_____ == "Yes"),
    turnout_rate = _____ / _____
  )
```

1.4 4. Advanced Transformations

1.4.1 4.1 Match: Functions and Purposes

Match each function with its purpose:

Functions: a) round() b) log() c) sqrt() d) abs() e) lag()

Purposes: 1. Remove decimal places 2. Transform skewed data 3. Calculate square root 4. Make negative values positive 5. Get previous row's value

Matches: a = _____, b = _____, c = _____, d = _____, e = _____

1.4.2 4.2 Code Detective: Complex Transformation

What does this code calculate?

```
data %>%
  group_by(state) %>%
  mutate(
    state_avg = mean(income, na.rm = TRUE),
    income_diff = income - state_avg,
    above_avg = income_diff > 0
  )
```

This code calculates: _____

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, create a new variable called ideology_category using case_when() that categorizes ideology_score into 'left', 'center', and 'right' based on terciles. Keep the code simple and explain what each line does. Here is what my data looks like: [paste glimpse output]"

For each AI exercise: - Work with Claude to analyze the data - Record your prompts and key findings

2.1 5. Creating Political Variables

Dataset: legislative_votes.csv

Description: Roll call voting data with legislator information.

Variables: - `legislator_id`: Unique identifier (int) - `name`: Legislator name (chr) - `party`: Political party (chr) - `state`: State abbreviation (chr) - `district`: District number (int) - `ideology_score`: -1 (liberal) to 1 (conservative) (dbl) - `vote_attendance`: Percentage of votes attended (dbl) - `bills_sponsored`: Number of bills sponsored (int) - `years_service`: Years in office (int) - `committee_count`: Number of committees served on (int)

2.1.1 5.1 Initial Data Exploration

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

```
Warning: package 'ggplot2' was built under R version 4.5.2
```

```
-- 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   4.0.1     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
```

```
legislators <- read_csv("legislative_votes.csv")
```

```
Rows: 500 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): name, party, state
dbl (7): legislator_id, district, ideology_score, vote_attendance, bills_spo...

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(legislators)
```

```
Rows: 500
Columns: 10
$ legislator_id   <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
$ name            <chr> "Legislator 1", "Legislator 2", "Legislator 3", "Legis~
$ party           <chr> "Democrat", "Democrat", "Republican", "Democrat", "Dem~
$ state           <chr> "MO", "AK", "PA", "FL", "HI", "CO", "NM", "CA", "IN", ~
$ district        <dbl> 17, 7, 10, 12, 4, 20, 3, 1, 16, 5, 19, 17, 11, 1, 1, 1~
$ ideology_score  <dbl> -0.765710472, 0.399555999, 0.117957905, 0.576843757, 0~
$ vote_attendance <dbl> 91.73352, 86.48068, 77.40871, 76.54805, 89.24765, 97.6~
$ bills_sponsored <dbl> 4, 8, 10, 5, 4, 2, 6, 8, 7, 5, 3, 7, 8, 10, 2, 7, 3, 4~
$ years_service   <dbl> 2, 36, 19, 36, 36, 22, 14, 11, 5, 9, 1, 8, 4, 29, 37, ~
$ committee_count <dbl> 1, 4, 4, 4, 6, 4, 8, 8, 5, 5, 3, 5, 4, 1, 6, 3, 8, 8, ~
```

2.1.2 5.2 Creating Categorical Variables

Ask Claude to help you create: - An “ideology_category” variable (left, center, right) based on terciles of `ideology_score` - An “effectiveness” score based on bills sponsored and committee work (assume more than 4 is effective) - A “seniority” category based on years of service (>20 is senior)

2.1.3 5.3 Calculating Party Metrics

Work with Claude to calculate party-level statistics for the variables you created in the last step. For each party, what percent of legislators are in each category you created.

2.2 6. Education and Political Participation

Dataset: civic_engagement.csv

Description: Survey data on education and political participation.

Variables: - `respondent_id`: Unique identifier (int) - `education`: Highest degree earned (chr) - `age`: Age in years (int) - `income`: Annual income (dbl) - `voted_2020`: Whether voted in 2020 (chr: Yes/No) - `political_interest`: 1-10 scale (int) - `volunteer_hours`: Political volunteer hours per month (int) - `donations`: Political donations in dollars (dbl) - `social_media_political`: Hours per week on political social media (dbl)

2.2.1 6.1 Loading and Initial Transformation

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

```
Rows: 500 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): education, voted_2020
dbl (7): respondent_id, age, income, political_interest, volunteer_hours, do...

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(civic)
```

```
Rows: 500
Columns: 9
$ respondent_id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
$ education              <chr> "High School", "Graduate", "Less than HS", "BA"~
$ age                    <dbl> 54, 77, 70, 19, 27, 57, 73, 27, 19, 29, 52, 25,~
$ income                 <dbl> 84998.32, 84840.91, 92201.61, 46382.02, 22652.7~
```

```
$ voted_2020             <chr> "Yes", "Yes", "Yes", "No", "Yes", "No", "No", "~
$ political_interest     <dbl> 9, 10, 9, 7, 10, 5, 1, 7, 1, 7, 4, 6, 10, 9, 10~
$ volunteer_hours        <dbl> 4, 1, 2, 3, 0, 1, 3, 4, 3, 3, 4, 5, 4, 3, 3, 5,~
$ donations              <dbl> 260.36498, 0.00000, 149.78584, 23.66047, 377.70~
$ social_media_political <dbl> 6.96331577, 0.08703583, 7.53645203, 1.83398582,~
```

2.2.2 6.2 Education and Engagement Index

Work with Claude to: - Recode education into "High School +" and "Less than High School" - Create a variable called `activist` that is TRUE if the person donated more than the median, volunteered more than the median and voted in 2020.

2.3 7. Campaign Finance Transformations

Dataset: campaign_finance_2024.csv

Description: Campaign contribution data for 2024 elections.

Variables: - `contribution_id`: Unique identifier (int) - `candidate`: Candidate name (chr) - `party`: Political party (chr) - `office`: Office sought (chr) - `amount`: Contribution amount (dbl) - `date`: Date of contribution (date) - `contributor_type`: Individual, PAC, etc. (chr) - `state`: Contributor state (chr) - `employer`: Contributor employer (chr) - `occupation`: Contributor occupation (chr)

2.3.1 7.1 Data Preparation

```r
# Load the dataset
finance <- read_csv("campaign_finance_2024.csv")
```

```
Rows: 500 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (7): candidate, party, office, contributor_type, state, employer, occup...
dbl  (2): contribution_id, amount
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
# Explore the data
glimpse(finance)
```

```
Rows: 500
Columns: 10
$ contribution_id  <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16~
$ candidate        <chr> "Candidate I", "Candidate F", "Candidate E", "Candida~
$ party            <chr> "Democrat", "Republican", "Republican", "Democrat", "~
$ office           <chr> "Senate", "Governor", "Senate", "Governor", "Senate",~
$ amount           <dbl> 579.59202, 146.27368, 642.60921, 459.48787, 567.46356~
$ date             <date> 2024-04-18, 2024-01-12, 2024-03-06, 2024-08-30, 2024~
$ contributor_type <chr> "Individual", "Individual", "Individual", "Individual~
$ state            <chr> "WV", "VT", "HI", "AR", "OK", "GA", "KY", "IL", "RI",~
$ employer         <chr> "Tech Company", "University", "Tech Company", "Univer~
$ occupation       <chr> "Executive", "Physician", "Executive", "Retired", "Re~
```

2.3.2 7.2 Creating Contribution Categories

Ask Claude to help you: - Identify small donors vs major donors (< mean or > mean) - Create variables for in-state vs out-of-state contributions (assume that your state is New York)

2.3.3 7.3 Time-Based Transformations

Work with Claude to summarize the data. - When was the most money given? Contribution timing (early, middle, late in campaign) - How much money was given each month? - How much was contributed over time

2.4 8. Demographic Transformations

Dataset: census_political.csv

Description: Census data merged with political outcomes.

Variables: - `county_id`: County FIPS code (int) - `county_name`: County name (chr) - `state`: State abbreviation (chr) - `population`: Total population (int) - `median_age`: Median age (dbl) - `pct_college`: Percent with college degree (dbl) - `median_income`: Median household income (dbl) - `unemployment_rate`: Unemployment percentage (dbl) - `dem_vote_share_2020`: Democratic vote share 2020 (dbl) - `turnout_2020`: Voter turnout 2020 (dbl)

2.4.1 8.1 Initial Exploration

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

```
Rows: 500 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): county_name, state
dbl (8): county_id, population, median_age, pct_college, median_income, unem...

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.
```

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

```
Rows: 500
Columns: 10
$ county_id           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
$ county_name         <chr> "County 1", "County 2", "County 3", "County 4", "C~
$ state               <chr> "NE", "KY", "KY", "CT", "RI", "KY", "WV", "IA", "A~
$ population          <dbl> 283363, 365881, 612761, 700633, 228297, 982323, 14~
$ median_age          <dbl> 42.17757, 39.16953, 47.61192, 32.49899, 33.38091, ~
$ pct_college         <dbl> 43.18251, 42.55179, 16.03140, 35.61717, 39.63398, ~
$ median_income       <dbl> 80961.13, 54125.69, 56570.75, 68557.12, 38715.02, ~
$ unemployment_rate   <dbl> 7.517104, 4.788997, 8.653567, 4.613327, 2.577154, ~
$ dem_vote_share_2020 <dbl> 46.79610, 42.95137, 48.17567, 25.27829, 46.37056, ~
$ turnout_2020        <dbl> 54.75303, 75.73557, 58.97276, 75.85745, 77.93041, ~
```

2.4.2 8.2 Creating Composite Indicators

Work with Claude to: - Create urban/rural classifications based on population density (above or below the median) - Create a variable that captures the quartiles of unemployment