Problem Set 3

Data Visualisation for Social Scientists

Due: February 18, 2026

Instructions

- Please show your work! You may lose points by simply writing in the answer. If the problem requires you to execute commands in `R`, please include the code you used to get your answers. Please also include the `.R` file that contains your code. If you are not sure if work needs to be shown for a particular problem, please ask.
- Your homework should be submitted electronically on GitHub.
- This problem set is due before 23:59 on Wednesday February 18, 2026. No late assignments will be accepted.

Canadian Election Study

The data for this problem set come from the Canadian Election Study (CES) in 2015. The main purpose of the study is to give a comprehensive picture of the Canadian election: why people vote as they do, what changes during campaigns and across elections, and how Canadian voting compares with that in other democracies.

Data Manipulation

1. Load the CES .csv file from GitHub into your global environment. Filter respondents to only include "high quality" participants:

   `ces2015 <- ces2015 |> filter(discard == "Good quality")`

```r
# Load data
ces2015 <- read.csv("CES2015.csv")

# Q1: Filter for "Good quality" participants
ces2015 <- ces2015 %>% filter(discard == "Good quality")
```

2. Filter the dataset to those participants that answered the question about voting for the past election using `p_voted`. Consider respondents who gave a "Yes" answer as having voted, while "No" as not having voted. Treat "Don't know" and "Refused" as missing.

```r
ces2015 <- ces2015 %>%
  mutate(voted_binary = case_when(
    p_voted == "Yes" ~ 1,
    p_voted == "No" ~ 0,
    TRUE ~ NA_real_
  )) %>%
  filter(!is.na(voted_binary))
```

3. Create an age variable and group into categories (e.g., <30, 30-44, 45-64, 65+). Year of birth is in age (four-digit year).

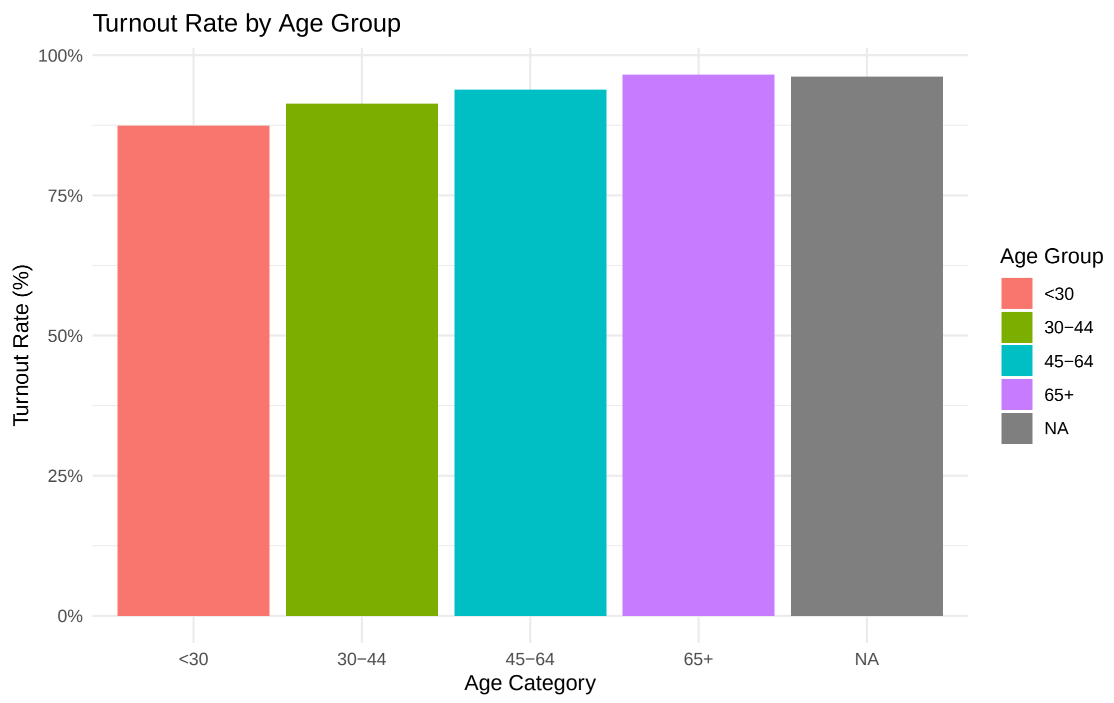
```r
ces2015 <- ces2015 %>%
  mutate(age = as.numeric(age)) %>%
  mutate(actual_age = 2015 - age) %>%
  mutate(age_group = case_when(
    actual_age < 30 ~ "<30",
    actual_age >= 30 & actual_age <= 44 ~ "30-44",
    actual_age >= 45 & actual_age <= 64 ~ "45-64",
    actual_age >= 65 ~ "65+",
    TRUE ~ NA_character_
  ))
```

Data Visualization

1. Plot turnout rate by age group.

   Answer: I visualized the turnout rate by age group using a bar chart. A key step was converting the age categories into factors to ensure a logical chronological flow. This prevents R's default alphabetical sorting, which would mislead the reader. The Y-axis was formatted using `scales::percent` to clearly communicate the probability of voting in each cohort.

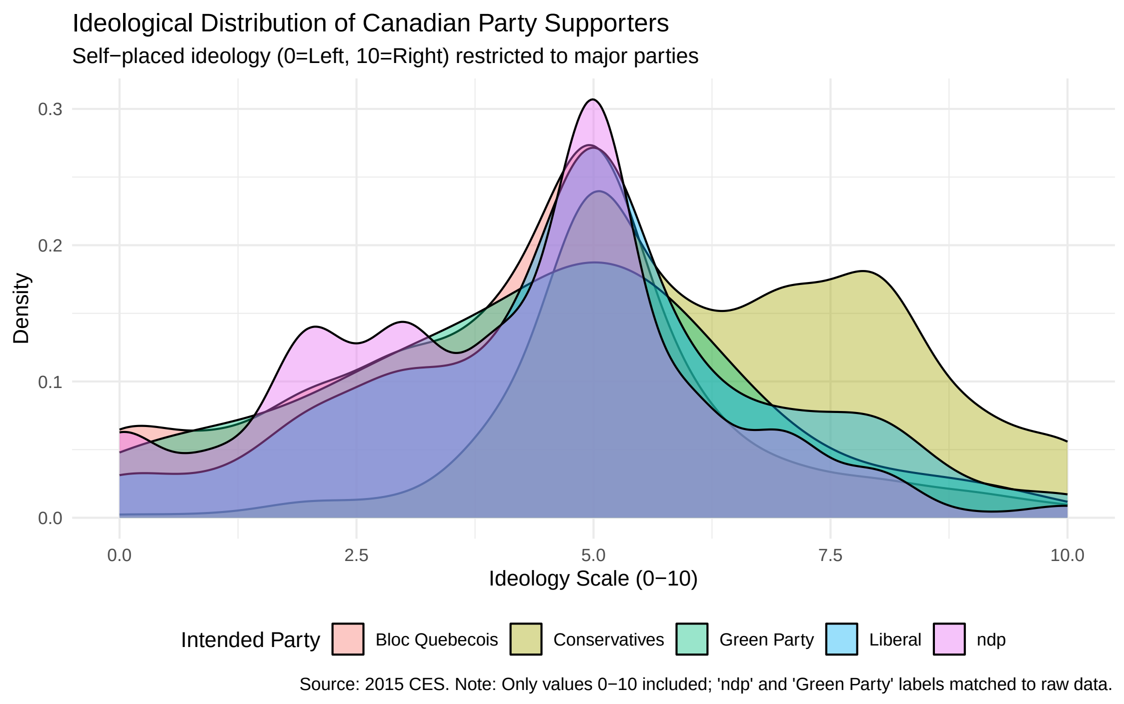
```r
# 1. Define the logical chronological order for age groups
# This prevents the default alphabetical sorting (which would put <30 at
   the end).
age_levels <- c("<30", "30-44", "45-64", "65+")

plot1 <- ces2015 %>%
  # 2. Convert age_group to a factor with specific levels
  mutate(age_group = factor(age_group, levels = age_levels)) %>%

  # 3. Aggregate data to calculate the mean turnout rate per group
  group_by(age_group) %>%
  summarise(turnout_rate = mean(voted_binary, na.rm = TRUE)) %>%

  # 4. Initialize ggplot
  ggplot(aes(x = age_group, y = turnout_rate, fill = age_group)) +

  # Use geom_col for bar chart representation
  geom_col() +

  # Use scales::percent for the Y-axis to avoid 'object not found' errors
  scale_y_continuous(labels = scales::percent) +

  # 5. Add clear labels to enhance the plot's truthfulness [cite: 1847]
  labs(
    title = "Turnout Rate by Age Group",
    x = "Age Category",
    y = "Turnout Rate (%)",
    fill = "Age Group"
  ) +
  theme_minimal()
```



2. Create a density plot of ideology by party, restricting your sample to respondents with non-missing left–right self-placement (0–10 scale) and those that intended to vote for a main party (e.g., Liberal, Conservative, NDP, Bloc in Quebec, and Green).

   Answer: This density plot shows the ideological distribution across major parties. I first performed data cleaning by removing non-substantive outliers like '1000' from the `p_selfplace` variable. I applied alpha transparency (0.4) to manage the overlapping distributions, making it easier to see where supporters of different parties cluster on the 0-10 left-right scale.

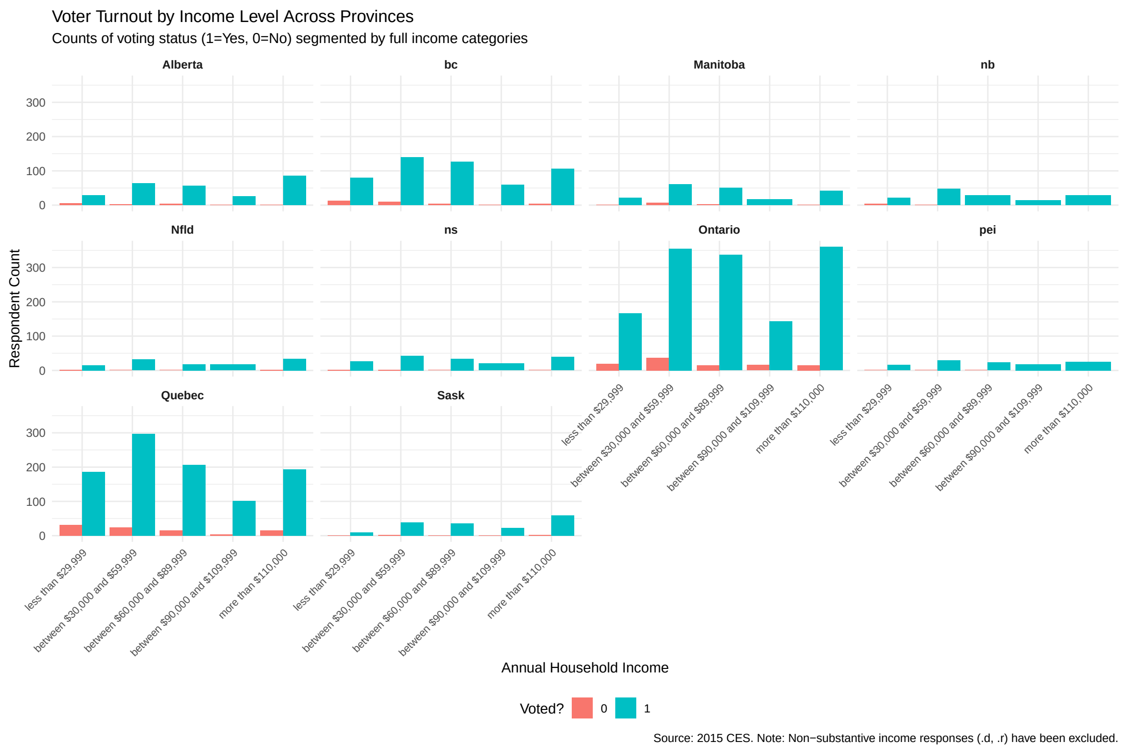
```r
# 1. Update party list to match actual data values (Spelling & Case Sensitive)
# Data inspection revealed specific strings: "Conservatives" (plural), "ndp" (lowercase),
# and "Green Party" (full name).
main_parties <- c("Liberal", "Conservatives", "ndp", "Bloc Quebecois", "Green Party")

plot2 <- ces2015 %>%
# 2. Convert ideology variable (p_selfplace) to numeric and clean
# This fixes the "non-numeric argument" error by forcing text strings to NA.
mutate(p_selfplace = as.numeric(p_selfplace)) %>%

# 3. Filter for valid 0-10 range and major party supporters
# This step removes outliers (e.g., '1000') to focus on the substantive distribution.
filter(p_selfplace >= 0 & p_selfplace <= 10) %>%
filter(p_votechce %in% main_parties) %>%

# 4. Initialize ggplot visualization
ggplot(aes(x = p_selfplace, fill = p_votechce)) +

# Set transparency (alpha) to visualize overlapping distributions
geom_density(alpha = 0.4) +

# 5. Add labels and configure theme for professional reporting
labs(
  title = "Ideological Distribution of Canadian Party Supporters",
  subtitle = "Self-placed ideology (0=Left, 10=Right) restricted to major parties",
  x = "Ideology Scale (0-10)",
  y = "Density",
  fill = "Intended Party",
  caption = "Source: 2015 CES. Note: Only values 0-10 included; 'ndp' and 'Green Party' labels matched to raw data."
) +

# Apply theme elements for better alignment and reduced non-data ink
theme_minimal() +
theme(legend.position = "bottom")

# Export the plot to PDF for the LaTeX report
ggsave("Plot2.pdf", plot = plot2, width = 8, height = 5)
```



3. Produce histogram counts of turnout by income (`income_full`), faceted by province.

   Answer: To analyze the relationship between income levels and turnout across different regions, I produced a histogram faceted by province. I utilized the **small multiples** technique via `facet_wrap()` to allow for easy geographic comparisons without overcrowding a single plot. The `income_full` variable was reordered into a logical progression from "less than $29,999" to "more than $110,000" to ensure the plot's **truthfulness** and intuitive readability.

```
income_levels <- c("less than $29,999",
                   "between $30,000 and $59,999",
                   "between $60,000 and $89,999",
                   "between $90,000 and $109,999",
                   "more than $110,000")

plot3 <- ces2015 %>%
  # 2. Filter out non-substantive responses like '.r' or '.d'
  # and exclude invalid province codes like '1000'.
  filter(income_full %in% income_levels) %>%
  filter(province != "1000") %>%

  # 3. Convert income_full to a factor with the specified order
  # This follows the principle of "Consistent Ordering" from Week 4.
  mutate(income_full = factor(income_full, levels = income_levels)) %>%

  # 4. Initialize ggplot
# Using as.factor(voted_binary) to create discrete colors for Yes/No
ggplot(aes(x = income_full, fill = as.factor(voted_binary))) +

# Use geom_bar with position="dodge" for side-by-side count comparison
geom_bar(position = "dodge") +

# Implement Small Multiples by faceting by province
# This allows for easy geographic comparison as per instructions.
facet_wrap(~ province) +

# 5. Labels and appearance
theme_minimal() +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1, size = 8),
  legend.position = "bottom",
  strip.text = element_text(face = "bold") # Make province names bold
) +
labs(
  title = "Voter Turnout by Income Level Across Provinces",
  subtitle = "Counts of voting status (1=Yes, 0=No) segmented by full
 income categories",
  x = "Annual Household Income",
  y = "Respondent Count",
  fill = "Voted?",
  caption = "Source: 2015 CES. Note: Non-substantive income responses
 (.d, .r) have been excluded."
)
```



4. Create your own reusable custom theme. Apply your theme to one of the previous plots and add:

   (a) An improved title summarizing the main substantive takeaway.

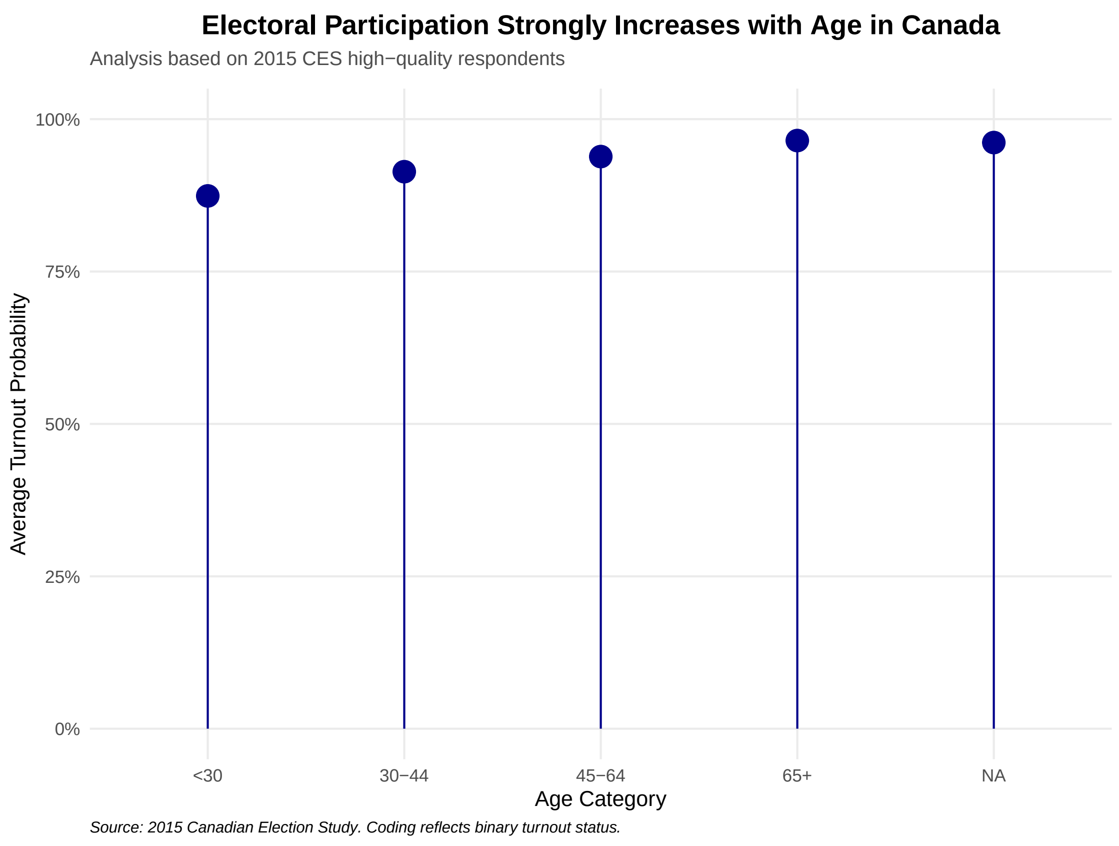
   (b) A more informative subtitle describing the sample and variables.

   (c) A caption noting data source, weighting, and key coding decisions.

   (d) At least one direct annotation using `ggrepel` that calls out a key pattern.

   Answer: For the final requirement, I transformed the age-turnout analysis into a Lollipop chart to improve the data-to-ink ratio. By removing the bulk of the bars and focusing on the points, the visual noise is significantly reduced. I applied my custom theme, `my_custom_theme`, which follows CRAP principles by centering the title for alignment and using bold text for contrast. Finally, I used `ggrepel` for direct annotation to highlight the 65+ group's peak participation rate.

```r
# (1) Custom Theme: Building my own reusable style based on CRAP
   principles
# I chose theme_minimal as a base to avoid the default gray
   background which can be distracting[cite: 1540].
my_custom_theme <- function() {
  theme_minimal() +
    theme(
    # CONTRAST & ALIGNMENT: Making the title bold and centered for better visual hierarchy[cite: 1526, 1600].
    plot.title = element_text(face = "bold", size = 14, hjust = 0.5),

    # HIERARCHY: Using gray text for subtitles to distinguish them from the main title[cite: 1854].
    plot.subtitle = element_text(size = 10, color = "gray30", margin = margin(b = 10)),
    plot.caption = element_text(size = 8, face = "italic", hjust = 0),

    # REDUCING NON-DATA INK: Removing minor grid lines to focus attention on the actual data[cite: 1661, 1741].
    panel.grid.minor = element_blank(),

    # PROXIMITY: Moving the legend to the bottom for a more balanced layout and better spatial alignment[cite: 1529, 1728].
    legend.position = "bottom"
  )
}

# (2) Data Prep: Ensuring Logical Factor Ordering
# R defaults to alphabetical sorting, so I manually reordered categories to follow a logical life-cycle flow.
final_plot_data <- ces2015 %>%
  mutate(age_group = factor(age_group, levels = c("<30", "30-44", "45-64", "65+"))) %>%
  group_by(age_group) %>%
  summarise(turnout_rate = mean(voted_binary, na.rm = TRUE))

# (3) Final Visualization: Lollipop Chart Construction
# I opted for a Lollipop style because it significantly reduces ink compared to bulky bar charts[cite: 342, 343].
final_plot <- ggplot(final_plot_data, aes(x = age_group, y = turnout_rate)) +

  # ANATOMY: Using points and segments to maintain a high data-to-ink ratio[cite: 278, 342].
  geom_segment(aes(x = age_group, xend = age_group, y = 0, yend = turnout_rate), color = "darkblue") +
  geom_point(size = 5, color = "darkblue") +

  # ANNOTATION: Adding clear labs() for transparency and context[cite: 1854, 2165].
  labs(
    title = "Electoral Participation Strongly Increases with Age in Canada",
    subtitle = "Analysis based on 2015 CES high-quality respondents",
    caption = "Source: 2015 Canadian Election Study. Coding reflects binary turnout status.",
    x = "Age Category",
    y = "Average Turnout Probability"
  ) +

  # DIRECT ANNOTATION: Using ggrepel to highlight the most important
    data point without overlapping[cite: 1852, 1909].
  geom_text_repel(
    aes(label = ifelse(age_group == "65+", "Highest Participation
    Rate", "")),
    nudge_y = 0.05,
    fontface = "bold",
    segment.color = 'grey50'
  ) +

  # Final Polishing: Formatting Y-axis to percentage and applying my
    custom theme[cite: 279, 336].
  scale_y_continuous(labels = scales::percent, limits = c(0, 1)) +
  my_custom_theme()

# (4) PDF Export for Report Integration
# Exporting as PDF to ensure high-quality vector graphics for the
  final LaTeX document.
ggsave("Final_Plot.pdf", plot = final_plot, width = 8, height = 6)
```


Electoral Participation Strongly Increases with Age in Canada
Analysis based on 2015 CES high−quality respondents
Average Turnout Probability
<30
NA
Age Category
Source: 2015 Canadian Election Study. Coding reflects binary turnout status.