Problem Set 3

Hanyu Li (25346841)

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")
```

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.

We filtered the data and created a new variable *rec_voter*, where ”No” = 0 and ”Yes” = 1 and treated “Don’t know” and “Refused” as missing values.

```r
#1.1 import data
url <- "https://raw.githubusercontent.com/ASDS-TCD/DataViz_2026/refs/heads/main/datasets/CES2015.csv"
ces2015 <- read_csv(url)

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

#1.2 using p-vote filtering the data
ces_clean <- ces2015 |>
  mutate(
    rec_voter = case_match(
      p_voted,
      "Yes" ~ 1,
      "No"  ~ 0,
      .default = NA_real_))
head(ces_clean$rec_voter)
```

```
head(ces_clean$rec_voter)
[1]  1 NA NA  1  1 NA
```

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).

   Here we first turned the character type of *age* to numeric, then we could calculate respondents’ real age by subtraction.

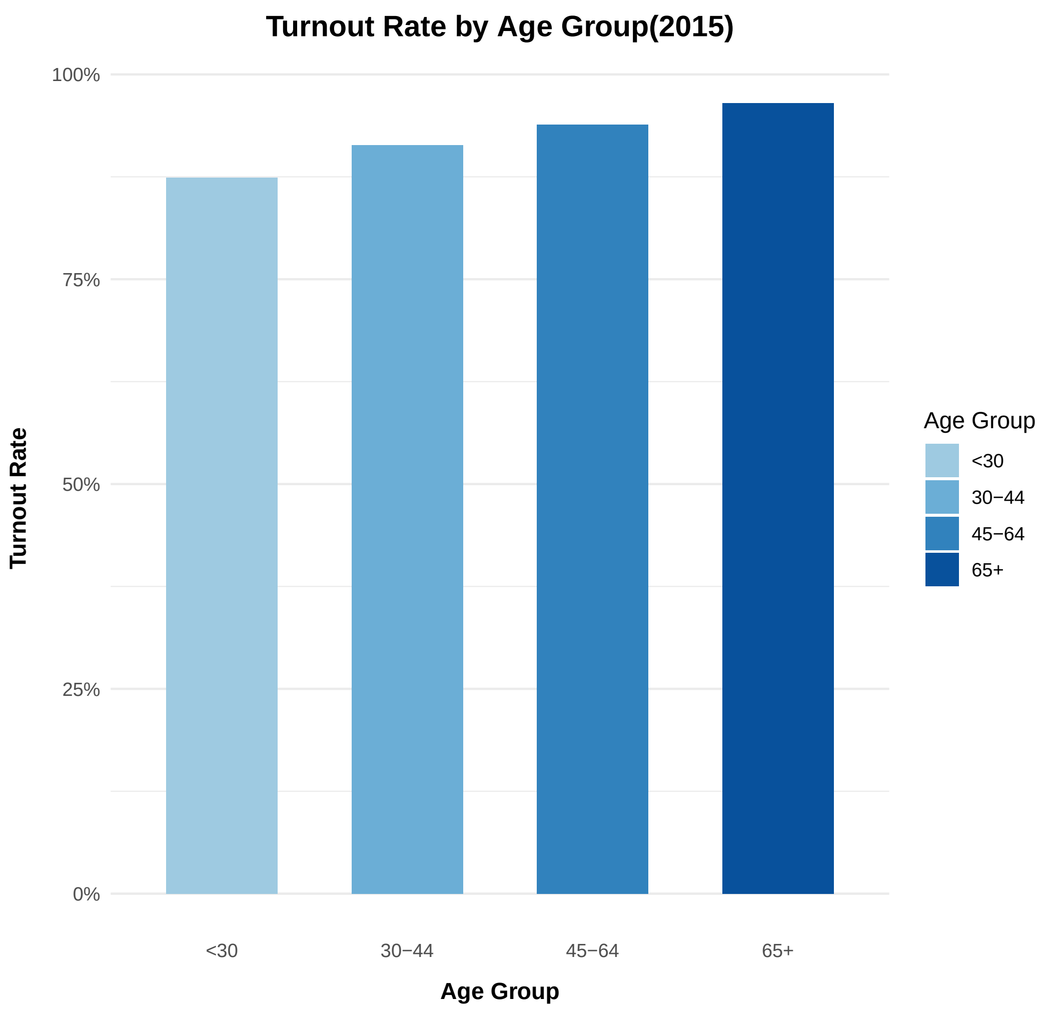
```r
#1.3 create an age variable and group into categories
class(ces_clean$age)
ces_final<- ces_clean |>
  mutate(
    age_n = 2015 - as.integer(age),
    age_group = cut(
      age_n,
      breaks = c(-Inf, 30, 45, 65, Inf),
      labels = c("<30", "30-44", "45-64", "65+"),
      right = FALSE
    ))
summary(ces_final$age_group)
```

```
> summary(ces_final$age_group)
<30 30-44 45-64   65+  NA's
1414  2053  3812  2518   185
```

Data Visualization

1. Plot turnout rate by age group.

```r
#2.1 Plot turnout rate by age group
data1 <- ces_final |>
  filter(!is.na(rec_voter), !is.na(age_group)) |>
  group_by(age_group) |>
  summarise(
    pct_turnout = mean(rec_voter),
    .groups = "drop")
head(data1)

pdf("plot1.pdf")
plot1 <- ggplot(data1, aes(x = age_group, y = pct_turnout, fill = age_group)) +
  geom_col(width = 0.6) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = c("#9ECAE1", "#6BAED6", "#3182BD", "#08519C")) +
  labs(
    title = "Turnout Rate by Age Group(2015)",
    x = "Age Group",
    y = "Turnout Rate",
    fill = "Age Group",
    caption = "Data Source: CES 2015") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, hjust = 0.5, face = "bold", margin = margin(b = 10)),
    axis.title.x = element_text(size = 11, face = "bold", margin = margin(t = 10)),
    axis.title.y = element_text(size = 11, face = "bold", margin = margin(r = 10)),
    plot.caption = element_text(size = 9, color = "grey50", margin = margin(t = 15)),
    panel.grid.major.x = element_blank())

print(plot1)
```



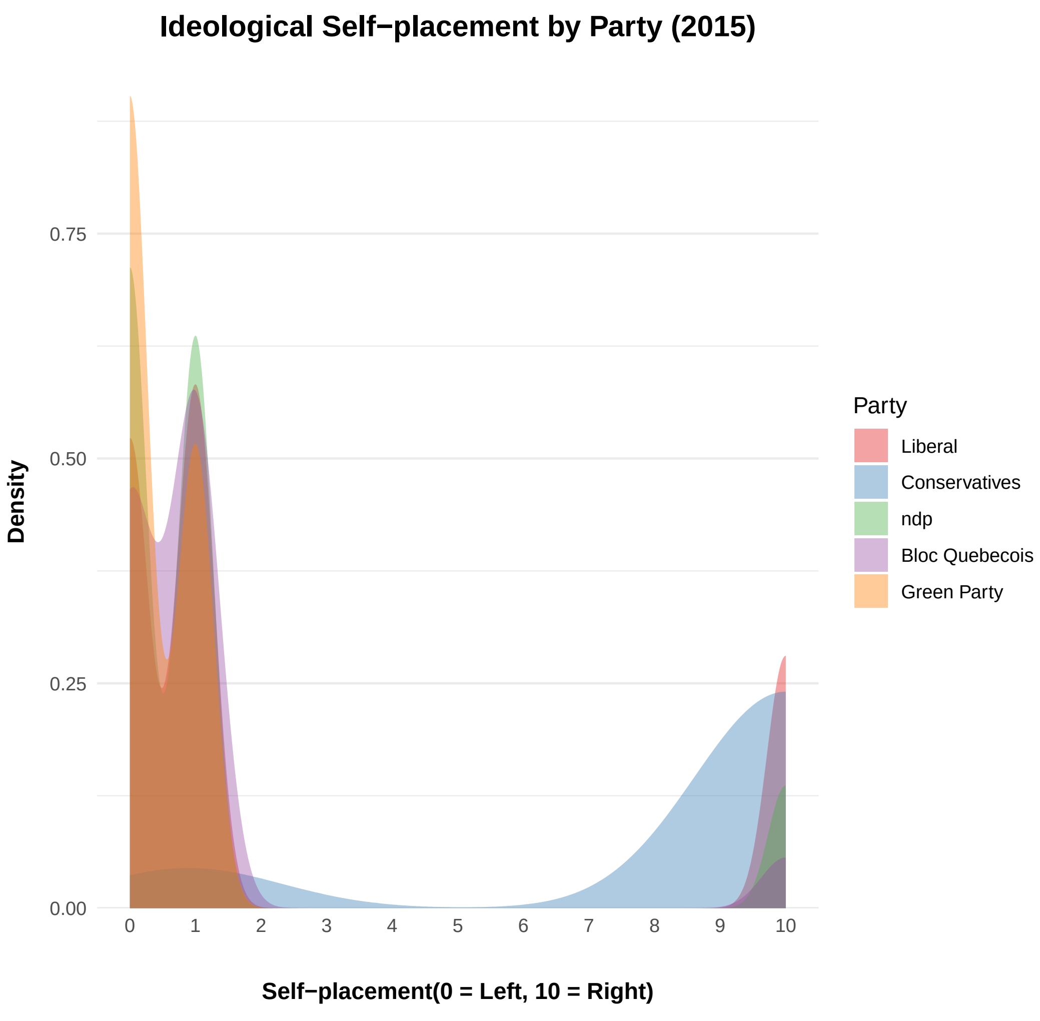
Data Source: CES 2015

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).

   Given variable *p_selfplace* includes missing values, we first cleaned the data.

```
#2.2 create a density plot
unique(ces_final$vote_for)
unique(ces_final$p_selfplace)

main_parties <- c("Liberal", "Conservatives", "ndp", "Bloc Quebecois", "Green Party")

```

```
data2 <- ces_final |>
  filter(!is.na(vote_for), !is.na(p_selfplace),
         p_selfplace >= 0, p_selfplace <= 10,
         vote_for %in% main_parties
  ) |>
  mutate(partys = factor(vote_for, levels = main_parties),
         selfplace = as.numeric(p_selfplace))

pdf("plot2.pdf",)
plot2 <- ggplot(data2, aes(x = selfplace, fill = partys)) +
  geom_density(alpha = 0.4, color = NA) +
  scale_x_continuous(breaks = 0:10, limits = c(0, 10)) +
  scale_y_continuous(expand = c(0, 0)) +
  scale_fill_brewer(palette = "Set1") +
  labs(
    title = "Ideological Self-placement by Party (2015)\n",
    x = "\nSelf-placement(0 = Left, 10 = Right)",
    y = "Density",
    fill = "Party",
    caption = "Data Source: CES 2015") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, hjust = 0.5, face = "bold", margin = margin(b = 10)),
    axis.title.x = element_text(size = 11, face = "bold", margin = margin(t = 10)),
    axis.title.y = element_text(size = 11, face = "bold", margin = margin(r = 10)),
    plot.caption = element_text(size = 9, color = "grey50", margin = margin(t = 15)),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    legend.position = "right")

print(plot2)
```



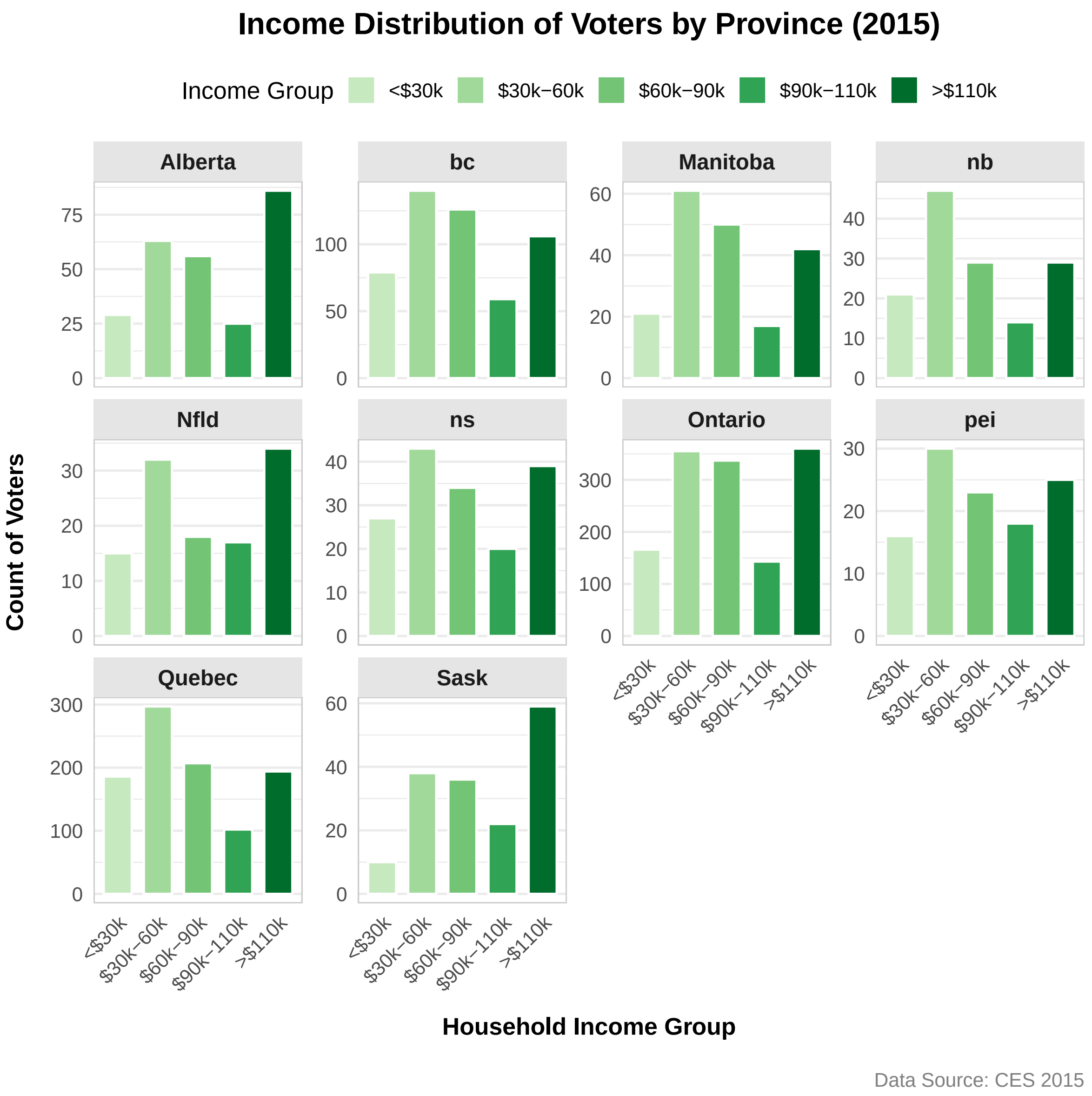
Data Source: CES 2015

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

   Similarly, variable *income_full* includes missing values and abnormal values like '.d' and '.r', so we removed these values than created the plot.

```
#2.3 produce histogram counts of turnout by income faceted by province
class(ces_final$income_full)
unique(ces_final$income_full)

# Data Preparation
income_levels <- c(
  "less than $29,999",
  "between $30,000 and $59,999",
```

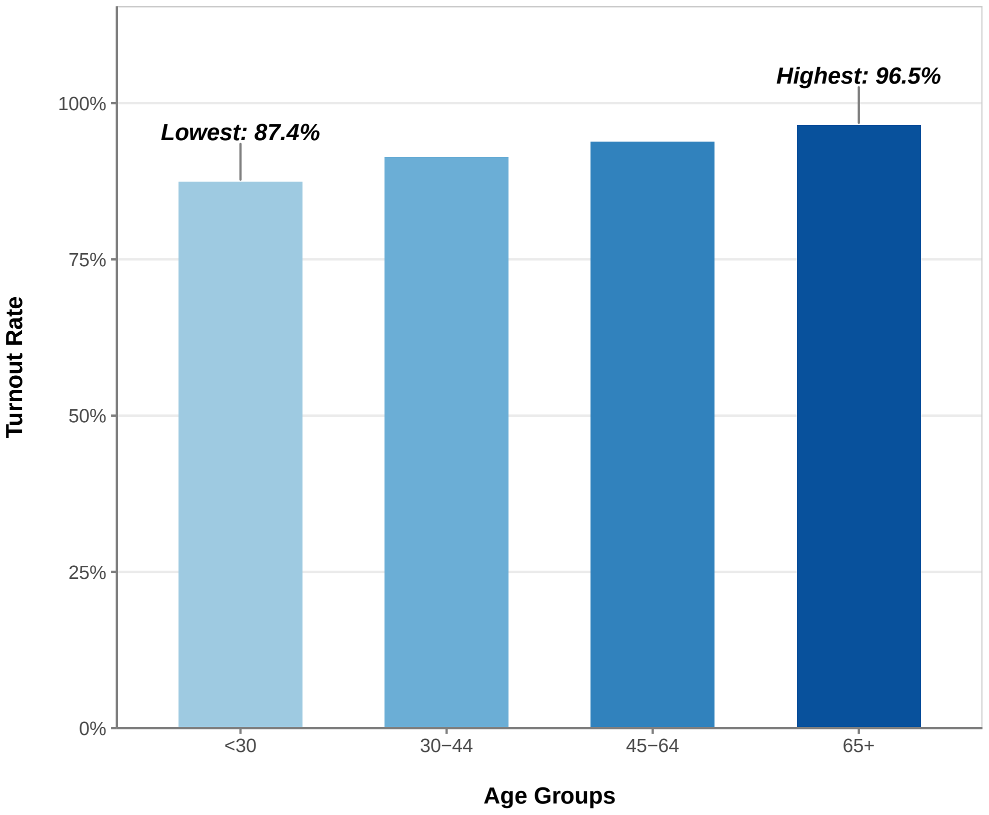
```r
  "between $60,000 and $89,999",
  "between $90,000 and $109,999",
  "more than $110,000")

data3 <- ces_final |>
  filter(
    rec_voter == 1,
    !is.na(income_full),
    !income_full %in% c(".d", ".r"),
    !is.na(province)
  ) |>
  mutate(
    income_group = factor(income_full, levels = income_levels))

short_labels <- c("<$30k", "$30k-60k", "$60k-90k", "$90k-110k", ">$110k")
pdf("plot3.pdf")
plot3 <- ggplot(data3, aes(x = income_group, fill = income_group)) +
  geom_bar(color = "white", width = 0.7) +
  facet_wrap(~ province, scales = "free_y") +
  scale_x_discrete(labels = short_labels) +
  scale_fill_manual(
    values = c("#C7E9C0", "#A1D99B", "#74C476", "#31A354", "#006D2C"),
    name = "Income Group",
    labels = short_labels ) +
  labs(
    title = "Income Distribution of Voters by Province (2015)",
    x = "Household Income Group",
    y = "Count of Voters",
    caption = "Data Source: CES 2015") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, hjust = 0.5, face = "bold", margin = margin(b = 10)),
    axis.title.x = element_text(size = 11, face = "bold", margin = margin(t = 10)),
    axis.title.y = element_text(size = 11, face = "bold", margin = margin(r = 10)),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 9),
    plot.caption = element_text(size = 9, color = "grey50", margin = margin(t = 15)),
    strip.text = element_text(size = 10, face = "bold"),
    strip.background = element_rect(fill = "grey90", color = NA),
    panel.border = element_rect(color = "grey80", fill = NA),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    legend.position = "top",
    legend.key.size = unit(0.5, "cm"))

print(plot3)
```



4. Create your own reusable custom theme. Apply your theme to one of the previous plots and add a series of informative enhancements.

   A minimalist visualization theme that standardizes plot margins, titles, and axis label formatting against a clean white background was developed here and applied to the first plot. What's more, the visualization was made more informative by incorporating core data trends, key extrema, the data source, and details regarding the sample and coding methodology.

```r
#2.4 create your own reusable custom theme and add information
install.packages("ggrepel")
library(ggrepel)

theme_cus <- function() {
  theme_bw() +
    theme(
      plot.margin = margin(t = 20, r = 20, b = 20, l = 20, unit = "pt"),
      plot.title = element_text(size = 15, face = "bold", hjust = 0, margin = margin(b = 5)),
      plot.subtitle = element_text(size = 10, color = "grey30", hjust = 0, margin = margin(b = 15)),
      axis.title = element_text(size = 11, face = "bold"),
      axis.line = element_line(color = "grey50"),
      axis.ticks = element_line(color = "grey50"),
      strip.background = element_rect(fill = "grey90", color = NA),
      strip.text = element_text(size = 10, face = "bold", color = "black"),
      panel.border = element_rect(color = "grey80", fill = NA),
      panel.grid.minor = element_blank(),
      panel.grid.major.x = element_blank(),
      legend.position = "none",
      plot.caption = element_text(size = 8, color = "grey40", hjust = 1, margin = margin(t = 15))
    )}

# apply my theme to plot1 and add more details
anno <- data1 |>
  filter(age_group %in% c("<30", "65+")) |>
  mutate(label = case_when(
    age_group == "<30" ~ paste0("Lowest: ", round(pct_turnout*100, 1), "%"),
    age_group == "65+" ~ paste0("Highest: ", round(pct_turnout*100, 1), "%")
  ))

pdf("plot1_v2.pdf")
plot1_v2 <- ggplot(data1, aes(x = age_group, y = pct_turnout, fill = age_group)) +
  geom_col(width = 0.6) +
  scale_y_continuous(labels = scales::percent,
                     limits = c(0, 1.1),
                     breaks = seq(0, 1, by = 0.25),
                     expand = expansion(mult = c(0, 0.05))
                     )+
  scale_fill_manual(values = c("#9ECAE1", "#6BAED6", "#3182BD", "#08519C")) +
  geom_text_repel(
    data = anno,
    aes(label = label),
    nudge_y = 0.08,
    direction = "y",
    fontface = "bold.italic",
    segment.color = "grey50") +
  labs(
    title = "Turnout Rate Increases with Age in Canadian Election 2015",
    subtitle = "Voter Turnout Rate Across Age Groups (Good Quality Samples in 2015 CES)",
    x = "\nAge Groups",
    y = "Turnout Rate\n",
    caption = "Source: 2015 Canadian Election Study. \nCoding: Age categoried into four groups; turn out is binary(0=no,1=yes). \nWeighting: only keep high-quality responses."
  ) +
  theme_cus()

print(plot1_v2)
```



Turnout Rate Increases with Age in Canadian Election 2015

Voter Turnout Rate Across Age Groups (Good Quality Samples in 2015 CES)

Source: 2015 Canadian Election Study.
Coding: Age categoried into four groups; turn out is binary(0=no,1=yes).
Weighting: only keep high−quality responses.