

# 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")
```

```
1 # DM 1: Load Data #
2 ces2015 <- read_csv("~/GitHub/DataViz_2026/datasets/CES2015.csv")
3 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.

```

1 # DM 2: Filter by answered question
2 unique(ces2015$p_voted) #unique answers incl NA/ "1000"
3 table(ces2015$p_voted) #see rough split to make sure its right
4 ces2015 <- ces2015 %>%
5   filter(p_voted %in% c("Yes", "No")) #filter only to those who responded
    yes or no

```

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).

```

1 # DM 3: Create Age Variable
2 str(ces2015$age) #character need to convert integer
3 table(ces2015$age) #check for missing values / potentially miscoded
  entries (i.e 1000)
4
5 ces2015 <- ces2015 %>%
6   filter(!age %in% c("refused", "don't know", "1000")) %>% #remove
    nonvalid yobs
7   mutate(
8     age = as.integer(age), #convert to numeric
9     age = 2015 - age, #years old as of 2015
10    cat_age = cut( #create categorical variable
11      age,
12      breaks = c(0, 25, 35, 45, 55, 65, 75, 150), #range breaks
13      labels= c("18-25", "26-35", "36-45", "46-55", "56-65", "65-75",
14      "<75")
15    )
16  )
17 table(ces2015$age) #check age range looks accurate
18 table(ces2015$cat_age) #view categories

```

## Data Visualization

Create random palette to use for some of the graphs:

```

1 scale_fill_eb <- function(...) {
2   scale_fill_manual(
3     values = c(
4       "#fd3838", "#ff8c02", "#7cb04b", "#7792ff", "#b960da", "#da5799", "#"
5       ff7257, "#c83131", "#e17421", "#399e72", "#5e72c5", "#9b5ab2", "#a72264"
6       , "#ff7257", "#ffb050", "#a4eac2", "#b6c3e8", "#e3aaaf", "#ed90af", "#fd3838"
7       , "#ff8c02", "#7cb04b", "#7792ff", "#b960da"
8     ),
9     ...
10   )
11 }

```

1. Plot turnout rate by age group. First need to create statistics summary for turnout for each age group created earlier.

```

1 # DV 1: Turnout rate by age group
2 turnout_agegroup<- ces2015 %>%
3   filter(!is.na(cat_age)) %>% #filter na age categories
4   group_by(cat_age) %>% #group within age category to
5   summarise(turnout = mean(p_voted == "Yes", na.rm = TRUE)) #turnout rate
       as yes voted vs no (filtered earlier)

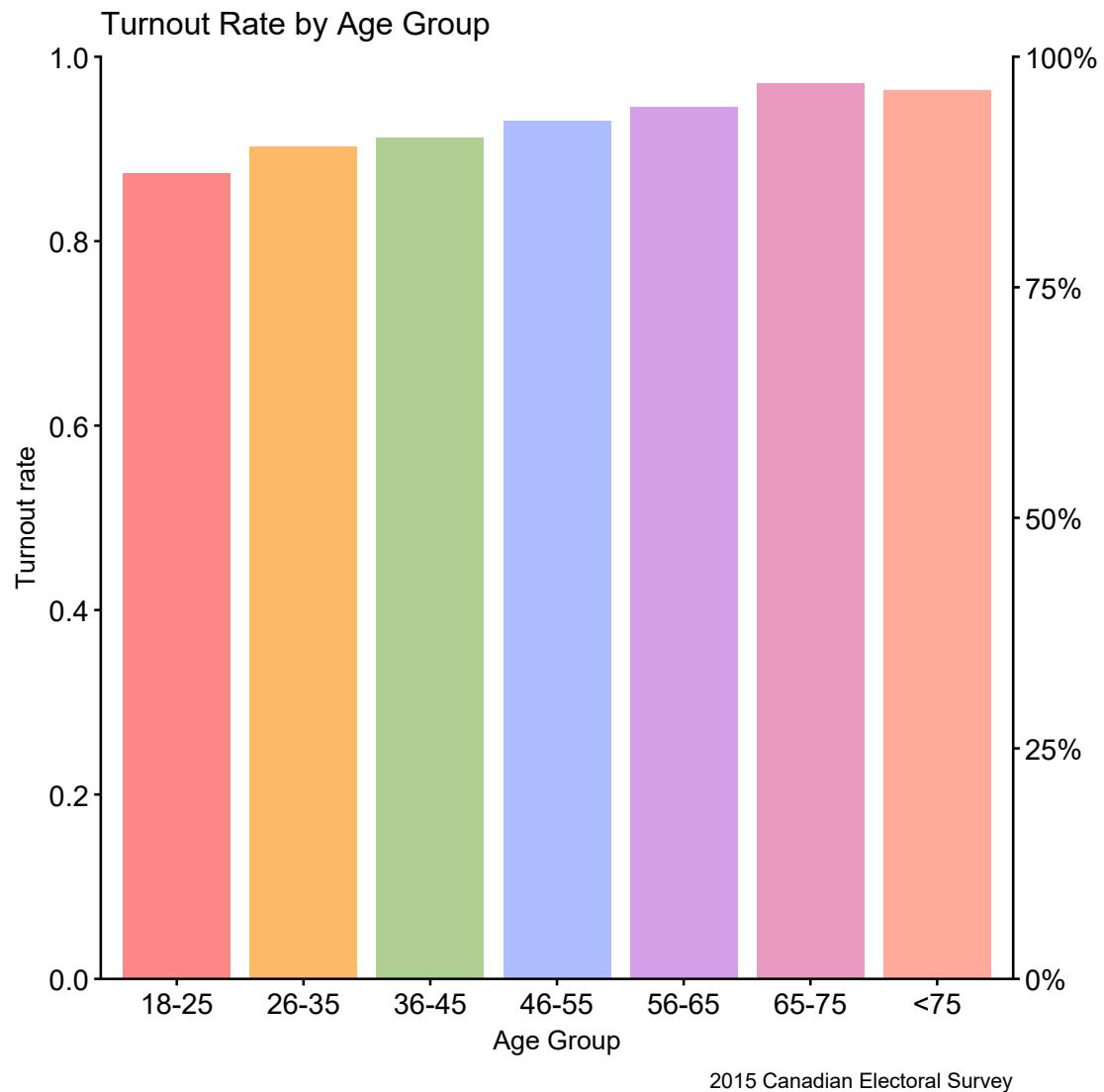
```

Plot using a bar chart. As age categories were already in ascending order when plotted they remain so. Add a second y-axis for fun to see both percentage and proportion forms.

```

1 plot1 <- ggplot(data = turnout_agegroup, aes(x = cat_age, y = turnout,
      fill = cat_age)) +
2   geom_col(alpha = 0.6, width = 0.85) +
3   scale_y_continuous(
4     limits = c(0,1),
5     breaks = seq(0, 1, by = 0.2),
6     expand = c(0,0),
7     sec.axis = sec_axis(~ . * 100, #add second axis for fun transform to
      percentage
                           labels = function(x) paste0(x, "%")) #add % label
8   ) +
9   scale_fill_eb() +
10  theme_classic(base_size = 12, base_family = "Arial") +
11  labs(x = "Age Group", y = "Turnout rate",
12        title = "2015 Canada Election Turnout Rate by Age Group", caption
13        = "2015 Canadian Electoral Survey") +
14  guides(fill = "none") +
15  theme(
16    axis.text.y = element_text(size = 13),
17    axis.text.x = element_text(size = 13)

```



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).

Data exploration on the structure and values within key variables. Transform relevant variables into correct form before removing missing values and specifying only the main parties to be kept. Then ordering the party intentions by the mean value of the left-right placement so when plotted it is easier to see the pattern.

```

1 # DV 2: Density Plot of ideology by party
2 table(is.na(ces2015$p_selfplace)) #number missing values for selfplace
3 unique(ces2015$p_selfplace) #can see NA and 1000 out of range
4 unique(ces2015$vote_for) #party intend to vote for, see spellings/missing
   values
5 table(ces2015$vote_for) #baseline to check for after
6
7
8 ideo_by_party <- ces2015 %>%
9   mutate(
10     p_selfplace = as.integer(p_selfplace), #turn to integer
11     vote_for = as.factor(vote_for) #transform to factor
12   ) %>%
13   filter(!is.na(p_selfplace), p_selfplace != "1000") %>% #remove na and
   1000 values
14   filter(vote_for %in% c("Liberal", "ndp", "Green Party", "Conservatives",
   , "Bloc Quebecois")) %>%
15   mutate(vote_for = recode(factor(vote_for), "ndp" = "NDP"))
16
17 table(ideo_by_party$vote_for) #check looks correct and parties
18 table(ideo_by_party$p_selfplace)
19 unique(ideo_by_party$vote_for)
20
21 #Order by mean left-right
22 ideo_by_party$vote_for <- reorder(ideo_by_party$vote_for, ideo_by_party$p
   _selfplace, FUN = mean)
```

Plot using faceted density plots with only one column to line up the x-axis for comparison. Looks similar to ggridge density plots but the facet title is in the center of each for a cleaner look. The hexcode colours for each party were extracted from the Wikipedia page. The bin-width was selected to see a semi-smooth density (due to integer scores) but enough to see the general trend. As adding annotations would appear for every facet, using spaces within the x-axis title (that appears once) to orient the left-right labels on the axis.

```

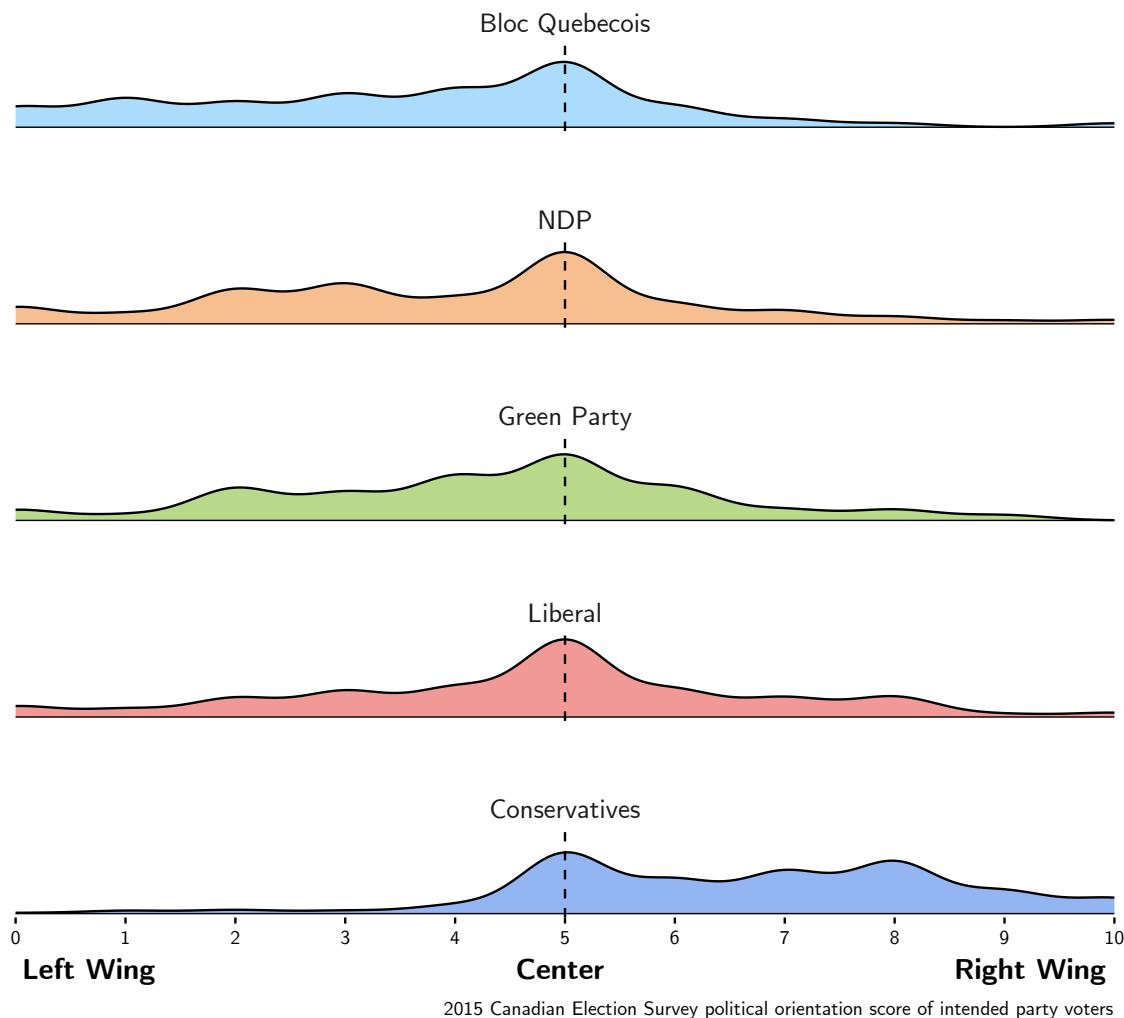
1 plot2 <- ggplot(ideo_by_party, aes(x= p_selfplace, fill= vote_for)) +
2   geom_density(alpha = 0.7, bw = 0.4) + #scale height of factors
3   geom_vline(xintercept = 5, linetype = "dashed") +
4   geom_hline(yintercept = 0, linewidth = 0.3) +
5   facet_wrap(vars(vote_for), nrow = 5) +
6   coord_cartesian(xlim = c(0,10)) +
7   scale_x_continuous(limits = c(0,10),
```

```

8           breaks = 0:10 ,
9           expand = c(0,0)) +
10      scale_fill_manual(values = c( #manually set colours to hexcodes
11        "Liberal" = "#EA6D6A", #correct party colours for clarity
12        "NDP" = "#F4A460",
13        "Green Party" = "#99C955",
14        "Conservatives" = "#6495ED",
15        "Bloc Quebecois" = "#87CEFA")) +
16      labs(x = "Left Wing"                               Center
17          Right Wing",
18          title = "Left–Right Self–Placement of Individuals Indending to
19          Vote for the Main Parties\n",
20          caption = "2015 Canadian Election Survey political orientation
21          score of intended party voters") +
22      guides(fill = "none") +
23      theme_classic(base_family = "CMU Sans Serif") +
24      theme(
25        axis.text.y = element_blank() , #rmv y axis
26        axis.title.y = element_blank() ,
27        axis.line.y = element_blank() ,
28        axis.line.x = element_blank() ,
29        axis.title.x = element_text(size = 13, face = "bold"),
30        strip.background = element_blank(),
31        axis.ticks.y = element_blank(),
32        plot.title = element_text(hjust = 0.5),
33        strip.text.x = element_text(size = 12, hjust = 0.5),
34        panel.spacing = unit(1, "cm"))

```

Left-Right Self-Placement of Individuals Indending to Vote for the Main Parties



3. Produce histogram counts of turnout by income (`income_full`), faceted by province. Key data manipulation here to clean data and get clear factors for income and province (labelled correctly). Removing missing values, before ordering income by income level (starting at lowest category) and province by size (i.e. by length when grouped) that will order the facets when plotting. Then creating a summary statistic of turnout (using yes from before) with the sum (for raw count rather than average) for income level per province to plot.

```

1 # DV3: Hisogram Counts turnout by income facet province 172
2 table(ces2015$income_full) #see range of factors
3 table(ces2015$province) #see codes used for shorthand of provinces
4 ces2015 <- ces2015 %>%
5   mutate(province = as.factor(province)) %>% #turn into factor
6   mutate(province = recode_factor(province, #change based on codebook
7         bc = "British Columbia",
8         nb = "New Brunswick",
9         ns = "Nova Scotia",
10        nwt = "Northwest Territories",
11        Nfld = "Newfoundland and \n Labrador",
12        pei = "Prince Edward Island",
13        Sask = "Saskatchewan")) %>%
14   mutate(income_full = as.factor(income_full)) %>% #turn income into
15   factor
15 filter(!income_full %in% c(".r", ".d")) %>% droplevels() #remove the
16   missing categories
16
17 table(ces2015$income_full)
18 ces2015$income_full <- relevel(ces2015$income_full,
19         ref = c("less than $29,999")) #specify
20   levels of factor lowest
21   #to highest
21
22
23 ces2015<- ces2015 %>% #reorder for facet placement by size of province
24   mutate(province = fct_reorder(province, province, .fun = length, .desc
25   = TRUE))
25
26
27 turnout_income <- ces2015 %>% #create turnout by income variable
28   group_by(province, income_full) %>% #grouped by province
29   filter(!is.na(income_full)) %>% #remove na
30   summarise(turnout = sum(p_voted == "Yes", na.rm = TRUE))
31
32 class(turnout_income$province)

```

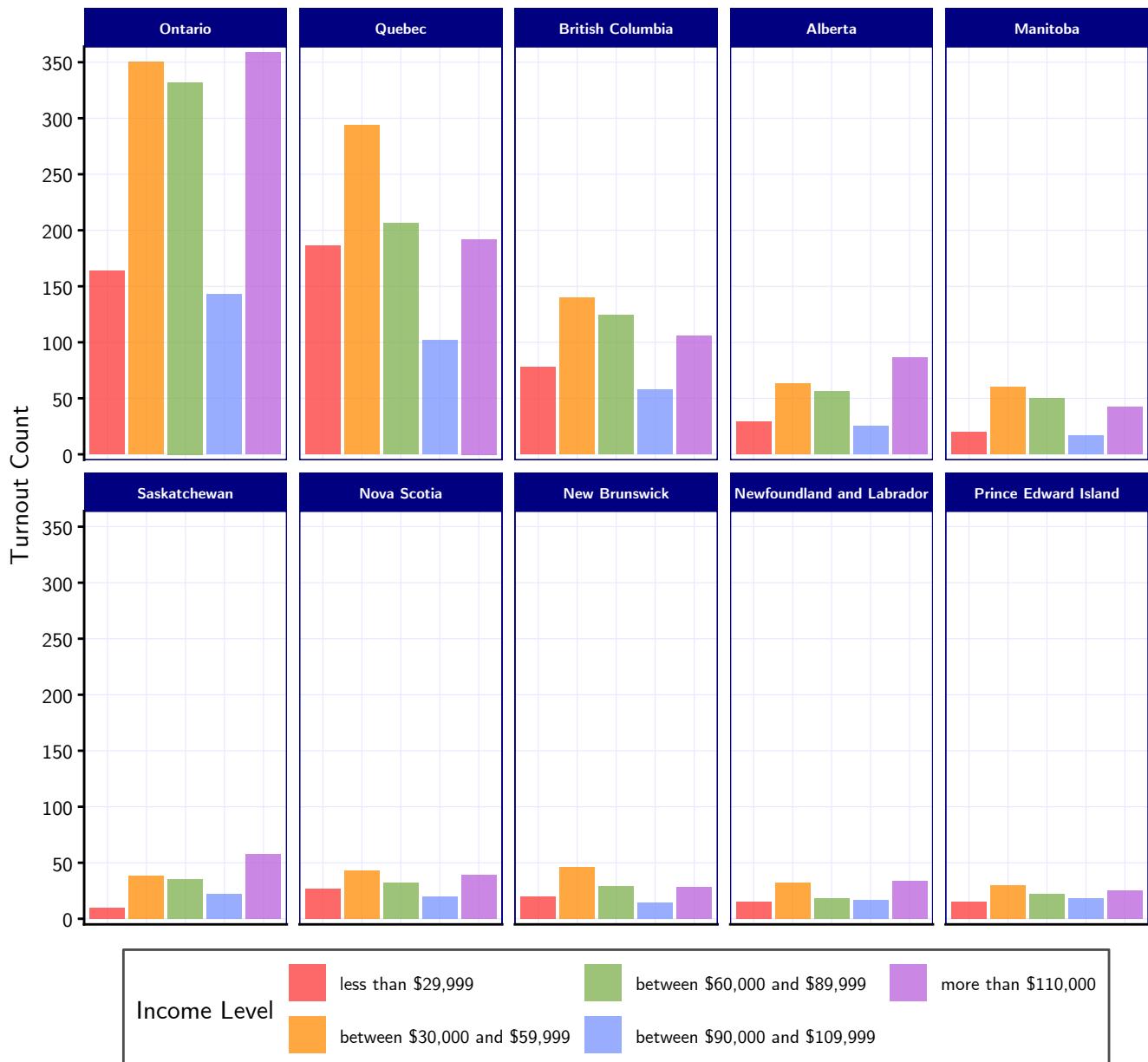
When plotting chose 5 columns as 10 categories for symmetry and ease of comparison, wrapping by province. Removing the messy elements like x-axis labels and relying on the legend for category interpretation was decided as the levels are already ordered. As

```

1 plot3 <- ggplot(turnout_income, aes(x = income_full, y = turnout, fill =
2   income_full)) +
3   geom_col(alpha = 0.75) +
4   facet_wrap(vars(province), ncol = 5) +
5   scale_y_continuous(breaks = seq(0, 400, by = 50), expand = c(0,5)) +
6   theme_classic(base_family = "CMU Sans Serif") +
7   labs(y = "Turnout Count", fill = "Income Level",
8     title = "Provincial turnout counts by income level",
9     caption = "Canadian Electoral Survey 2015. Total number of
10    respondants who said 'Yes' when asked if they voted") +
11   scale_fill_eb() +
12   theme(
13     legend.position = "bottom",
14     legend.text = element_text(size = 8),
15     axis.text.x = element_blank(),
16     axis.title.x = element_blank(),
17     axis.line.x = element_line(colour = "black"),
18     axis.ticks.x = element_blank(),
19     strip.text = element_text(size = 6.5, colour = "white", face = "bold")
20   ),
21   strip.background = element_rect(fill = "navy", colour = "navy"), #
22   panel.border = element_rect(color = "navy", fill = NA, linewidth = 0.4),
23   panel.grid.major.y = element_line(colour = "#e6e9ff", linewidth = 0.2), #very light grid
24   panel.grid.major.x = element_line(colour = "#e6e9ff", linewidth = 0.2),
25   legend.background = element_rect(fill = "white", colour = "grey30",
26     size = 0.5, linetype = "solid"))
27 ) +
28 guides(fill = guide_legend(nrow = 2))
29 ggsave("PS03_p3.pdf", plot3, device = cairo_pdf)

```

## Canadian Province Turnout Counts by Income Level



Canadian Electoral Survey 2015. Total number of respondents who said 'Yes' when asked if they voted

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.

Create work in progress theme.

```

1 # DV 4: Create customisable theme
2 theme_ellen <- function(...) {
3   theme_classic(base_size = 12, base_family = "CMU Sans Serif") +
4     theme(
5       # Titles
6       plot.title = element_text(face = "bold", size = rel(1.3)),
7       plot.subtitle = element_text(face = "plain", size = rel(1.1), color
= "grey70"),
8       plot.caption = element_text(face = "italic", size = rel(0.7),
9                                     color = "grey70", hjust = 0),
10      #Backgrounds
11      # Grid
12      panel.background = element_rect(fill = "white", colour = NA),
13      panel.border = element_blank(),
14      panel.grid = element_blank(),
15      panel.grid.minor = element_blank(),
16
17      #Axis
18      axis.title = element_text(family = "CMU Sans Serif", size = 12), #title font
19      axis.line.x = element_line(linewidth = 0.4), #axis lines
20      axis.line.y = element_line(linewidth = 0.4),
21      axis.text = element_text(colour = "black", size = 10), #axis text change
22      axis.ticks.x = element_line(colour = "black", linewidth = 0.75),
23
24      #Facets
25      strip.background = element_blank(),
26      strip.text = element_text(size = 11),
27
28      #Legend
29      legend.title = element_text(family = 'CMU Sans Serif', size = 12,
30                                   colour = 'black'), #legend
31      legend.text = element_text(family = 'CMU Sans Serif', size = 10,
32                                 colour = 'black'), # font change
33      legend.key.height = unit(1, "lines"), #spacing
34      legend.key.width = unit(1, "lines"),
35      legend.spacing.y = unit(1, "lines"),
36      legend.background = element_rect(
37        fill = NA,
```

```

36     colour = "grey60", #make box around legend to align it
37     size = 0.5,
38     linetype = "solid")
39 )
40 }
```

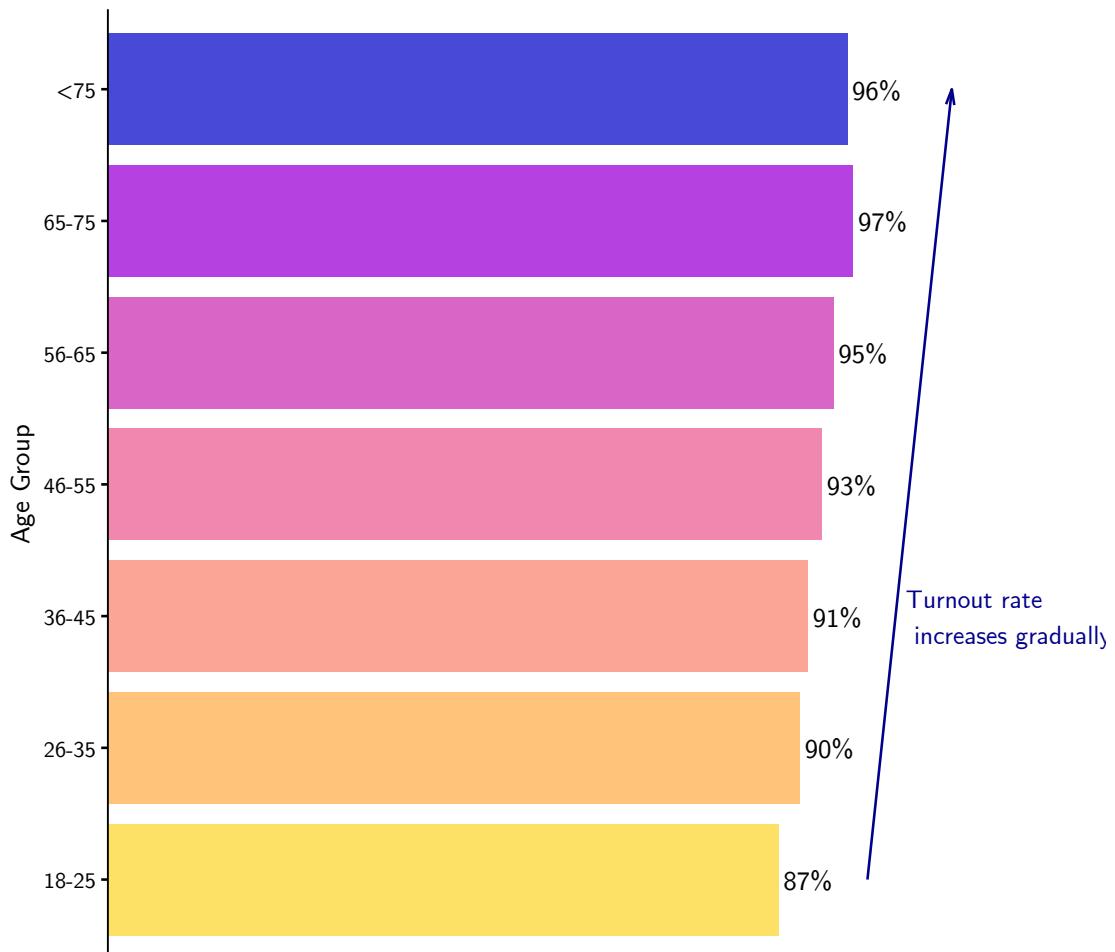
Now add it to first graph and try to make a better visualisation. Flip the coordinates, added more informative titles and captions, selected a more cohesive and appealing palette (slightly gradiating colours is okay as categories are ordered). Rather than having an axis with percentages add the actual average turnout percentage to the right of the horizontal bar to make it clearer. Annotate with arrow and text additionally to explain trend.

```

1 #Apply to previous
2 plot4 <- plot1 + theme_ellen() + #ironically done most of this in first
  plt
3 coord_flip() +
4 labs(title = "Turnout rate increased with Age in the 2015 Canadian
  Election",
5       subtitle = "Percentage of respondents proportioning to have voted in
  the 2015 election by age group.",
6       caption = "Source: Canadian Election Study 2015 Post-Election Poll
  . \n Poor quality participants and those with missing or irregular
  responses were removed.") +
7 theme(
8   axis.text.x = element_blank(),
9   axis.line.x = element_blank(),
10  axis.title.x = element_blank(),
11  axis.ticks.x = element_blank()
12 ) +
13 scale_fill_manual(values = c("#fdd735", "#ffb14e", "#fa8775", "#ea5f94"
  , "#cd34b5", "#9d02d7", "#0d0dca")) +
14 scale_y_continuous(limits = c(0,1.3), expand = c(0,0)) +
15 geom_text(
16   aes(label = scales::percent(turnout, accuracy = 1)),
17   hjust = -0.1) +
18 annotate(geom = "segment", x = 1, xend = 7, y = .99, yend = 1.1, colour
  = "darkblue",
19           arrow = arrow(angle = 15, length = unit(0.5, "lines")))) +
20 annotate(geom = "text", x = 3, y = 1.04, label = "Turnout rate \n
  increases gradually with age", hjust = 0, colour = "darkblue", size =
  4)
21
22 ggsave("PS03_p4.pdf", plot4, device = cairo_pdf)
```

## Turnout rate increased with Age in the 2015 Canadian Election

Percentage of respondents proportioning to have voted in the 2015 election by age group



Source: Canadian Election Study 2015 Post-Election Poll.  
Poor quality participants and those with missing or irregular responses were removed.