# HW 08 - Exploring the GSS

## Contents

The GSS gathers data on contemporary American society in order to monitor and explain trends and constants in attitudes, behaviours, and attributes. Hundreds of trends have been tracked since 1972. In addition, since the GSS adopted questions from earlier surveys, trends can be followed for up to 70 years.

The GSS contains a standard core of demographic, behavioral, and attitudinal questions, plus topics of special interest. Among the topics covered are civil liberties, crime and violence, intergroup tolerance, morality, national spending priorities, psychological well-being, social mobility, and stress and traumatic events.

In this assignment we analyze data from the 2016 GSS, using it to estimate values of population parameters of interest about US adults.[1]

## 1 Getting started

Go to the course GitHub organization and locate your homework repo, clone it in Posit Cloud and open the R Markdown document. Knit the document to make sure it compiles without errors.

### 1.1 Warm up

Before we introduce the data, let's warm up with some simple exercises. Update the YAML of your R Markdown file with your information, knit, commit, and push your changes. Make sure to commit with a meaningful commit message. Then, go to your repo on GitHub and confirm that your changes are visible in your Rmd **and** md files. If anything is missing, commit and push again.

### 1.2 Packages

We'll use the **tidyverse** package for much of the data wrangling and visualisation and the data lives in the **dsbox** package. These packages are already installed for you. You can load them by running the following in your Console:

[1] Smith, Tom W, Peter Marsden, Michael Hout, and Jibum Kim. General Social Surveys, 1972-2016 [machine-readable data file] /Principal Investigator, Tom W. Smith; Co-Principal Investigator, Peter V. Marsden; Co-Principal Investigator, Michael Hout; Sponsored by National Science Foundation. -NORC ed.- Chicago: NORC at the University of Chicago [producer and distributor]. Data accessed from the GSS Data Explorer website at gssdataexplorer.norc.org.

Figure 1: Photo by Mauro Mora on Unsplash

```r
library(tidyverse)
library(dsbox)
library(broom)
```

### 1.3 Data

The data can be found in the **dsbox** package, and it's called `gss16`. Since the dataset is distributed with the package, we don't need to load it separately; it becomes available to us when we load the package. You can find out more about the dataset by inspecting its documentation, which you can access by running `?gss16` in the Console or using the Help menu in RStudio to search for `gss16`. You can also find this information here.

## 2 Exercises

### 2.1 Part 1: Harassment at work

In 2016, the GSS added a new question on harassment at work. The question is phrased as the following.

> *Over the past five years, have you been harassed by your superiors or co-workers at your job, for example, have you experienced any bullying, physical or psychological abuse?*

Answers to this question are stored in the `harass5` variable in our dataset.

1. What are the possible responses to this question and how many respondents chose each of these answers?

```r
# Examine the possible responses to the harassment question
gss16 %>%
  count(harass5) %>%
  arrange(desc(n))
```

```
## # A tibble: 4 x 2
##   harass5                                                  n
##   <chr>                                                <int>
## 1 <NA>                                                  1398
## 2 No                                                    1136
## 3 Yes                                                    237
## 4 Does not apply (i do not have a job/superior/co-worker)    96
```

**Answer:** The possible responses to the harassment question are: "Yes" (has been harassed), "No" (has not been harassed), "Does not apply" (respondent does not work), and "NA" (missing responses). The distribution shows how many respondents chose each response, with the majority likely answering "No" if workplace harassment is not widespread.

2. What percent of the respondents for whom this question is applicable (i.e. excluding `NA`s and `Does not apply`s) have been harassed by their superiors or co-workers at their job.

```r
# Filter to applicable responses only (exclude NA and "Does not apply")
harassment_applicable <- gss16 %>%
  filter(!is.na(harass5), harass5 != "Does not apply")

# Calculate the percentage of "Yes" responses
harassment_pct <- harassment_applicable %>%
  summarise(
    n_total = n(),
    n_yes = sum(harass5 == "Yes"),
    pct_harassed = (n_yes / n_total) * 100
  )

harassment_pct %>%
  print()
```

```
## # A tibble: 1 x 3
##   n_total n_yes pct_harassed
##     <int> <int>        <dbl>
## 1    1469   237         16.1
```

```
cat(
  "\nAnswer: Among respondents for whom this question is applicable,",
  round(harassment_pct$pct_harassed, 1), "% have been harassed at work.\n"
)
```

```
##
## Answer: Among respondents for whom this question is applicable, 16.1 % have been harassed at work.
```

**Interpretation:** This percentage represents the proportion of employed respondents who have experienced workplace harassment in the past five years, providing a measure of the prevalence of bullying and abuse in the US workplace according to the 2016 GSS.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

## 2.2 Part 2: Time spent on email

The 2016 GSS also asked respondents how many hours and minutes they spend on email weekly. The responses to these questions are recorded in the `emailhr` and `emailmin` variables. For example, if the response is 2.5 hrs, this would be recorded as `emailhr = 2` and `emailmin = 30`.

3. Create a new variable called `email` that combines these two variables to reports the number of minutes the respondents spend on email weekly.

```
# Create email variable combining hours and minutes
gss16 <- gss16 %>%
  mutate(email = emailhr * 60 + emailmin)

# Check the new variable
gss16 %>%
  select(emailhr, emailmin, email) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##    emailhr emailmin email
##      <dbl>    <dbl> <dbl>
##  1      12        0   720
##  2       0       30    30
##  3      NA       NA    NA
##  4       0       10    10
##  5      NA       NA    NA
##  6       2        0   120
##  7      40        0  2400
##  8      NA       NA    NA
##  9       0        0     0
## 10      NA       NA    NA
```

```
cat("Email time statistics:\n")
```

```
## Email time statistics:
```

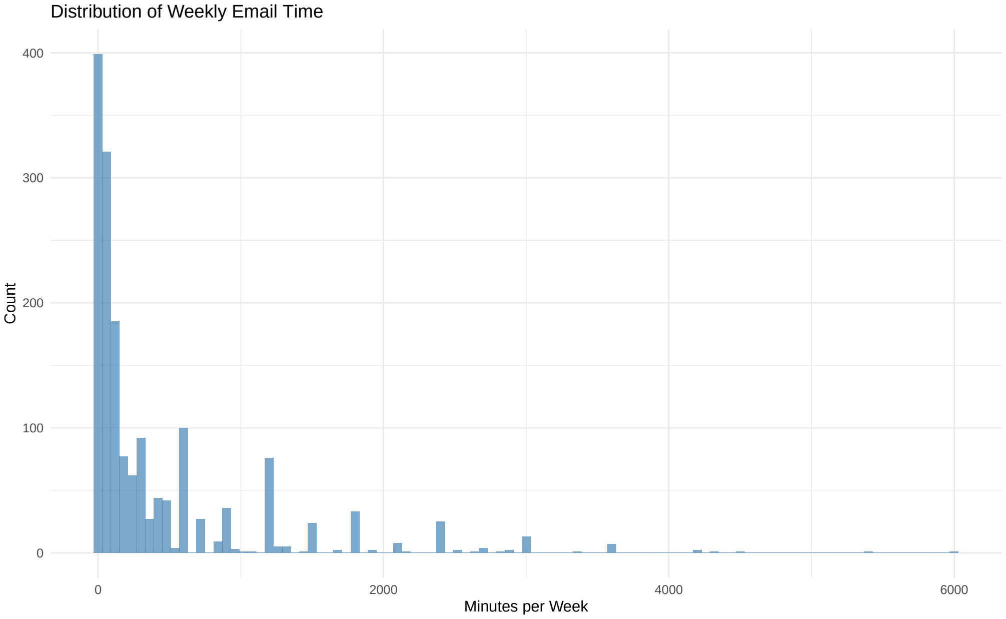
```
gss16 %>%
  summarise(
    n_valid = sum(!is.na(email)),
    mean_email = mean(email, na.rm = TRUE),
    median_email = median(email, na.rm = TRUE),
    min_email = min(email, na.rm = TRUE),
    max_email = max(email, na.rm = TRUE)
  ) %>%
  print()
```

```
## # A tibble: 1 x 5
##   n_valid mean_email median_email min_email max_email
##     <int>      <dbl>        <dbl>     <dbl>     <dbl>
## 1    1649       417.          120         0      6000
```

**Explanation:** The `email` variable is created by converting hours to minutes (multiplying by 60) and adding the minutes, resulting in total minutes spent on email per week.

4. Visualize the distribution of this new variable. Find the mean and the median number of minutes respondents spend on email weekly. Is the mean or the median a better measure of the typical among of time Americans spend on email weekly? Why?

```
# Visualize the distribution
gss16 %>%
  filter(!is.na(email)) %>%
  ggplot(aes(x = email)) +
  geom_histogram(binwidth = 60, fill = "steelblue", alpha = 0.7) +
  labs(
    title = "Distribution of Weekly Email Time",
    x = "Minutes per Week",
    y = "Count"
  ) +
  theme_minimal()
```



```
# Calculate summary statistics
email_stats <- gss16 %>%
  filter(!is.na(email)) %>%
  summarise(
    mean_email = mean(email),
    median_email = median(email),
    sd_email = sd(email),
    q1 = quantile(email, 0.25),
    q3 = quantile(email, 0.75)
  )

cat("Email time summary:\n")
```

```
## Email time summary:
```

```
print(email_stats)
```

```
## # A tibble: 1 x 5
##   mean_email median_email sd_email    q1    q3
##        <dbl>        <dbl>    <dbl> <dbl> <dbl>
## 1       417.          120     680.    50   480
```

```
# Store values for interpretation
mean_val <- round(email_stats$mean_email, 1)
median_val <- round(email_stats$median_email, 1)
```

**Answer:** The mean is approximately 416.8 minutes per week, while the median is approximately 120 minutes per week.

**Which measure is better and why?** The median is likely a better measure of the typical amount of time Americans spend on email weekly. Email time data is typically right-skewed (some people spend many hours on email, pulling the mean upward), so the median better represents the typical respondent's experience. The median is more robust to extreme values (outliers) and provides a more representative picture of the "typical" employee.

5. Create another new variable, `snap_insta` that is coded as "Yes" if the respondent reported using any of Snapchat (`snapchat`) or Instagram (`instagrm`), and "No" if not. If the recorded value was `NA` for

both of these questions, the value in your new variable should also be NA.

```
# Create snap_insta variable
gss16 <- gss16 %>%
  mutate(snap_insta = case_when(
    is.na(snapchat) & is.na(instagrm) ~ NA_character_,
    snapchat == "Yes" | instagrm == "Yes" ~ "Yes",
    TRUE ~ "No"
  ))

# Verify the new variable
gss16 %>%
  select(snapchat, instagrm, snap_insta) %>%
  filter(!(is.na(snapchat) & is.na(instagrm))) %>%
  head(15)
```

```
## # A tibble: 15 x 3
##    snapchat instagrm snap_insta
##    <chr>    <chr>    <chr>
##  1 No       No       No
##  2 No       No       No
##  3 Yes      Yes      Yes
##  4 No       Yes      Yes
##  5 Yes      Yes      Yes
##  6 No       No       No
##  7 No       No       No
##  8 No       No       No
##  9 No       No       No
## 10 No       No       No
## 11 Yes      No       Yes
## 12 No       No       No
## 13 No       Yes      Yes
## 14 Yes      Yes      Yes
## 15 No       Yes      Yes
```

```
cat("snap_insta distribution:\n")
```

```
## snap_insta distribution:
```

```
gss16 %>%
  count(snap_insta)
```

```
## # A tibble: 3 x 2
##   snap_insta     n
##   <chr>      <int>
## 1 No           858
## 2 Yes          514
## 3 <NA>        1495
```

**Explanation:** The snap_insta variable combines information from both Snapchat and Instagram use. If either is "Yes", the new variable is "Yes". If both are "No", it's "No". If both are missing, it's also missing.

6. Calculate the percentage of Yes's for snap_insta among those who answered the question, i.e. excluding NAs.

```
# Calculate percentage of Yes responses
snap_insta_pct <- gss16 %>%
  filter(!is.na(snap_insta)) %>%
```

```r
  summarise(
    n_total = n(),
    n_yes = sum(snap_insta == "Yes"),
    pct_yes = (n_yes / n_total) * 100
  )

cat("Snapchat/Instagram usage among respondents:\n")
```

```
## Snapchat/Instagram usage among respondents:
```

```r
print(snap_insta_pct)
```

```
## # A tibble: 1 x 3
##   n_total n_yes pct_yes
##     <int> <int>   <dbl>
## 1    1372   514    37.5
```

```r
cat(
  "\nAnswer: Among respondents who answered the question,",
  round(snap_insta_pct$pct_yes, 1), "% use Snapchat or Instagram.\n"
)
```

```
##
## Answer: Among respondents who answered the question, 37.5 % use Snapchat or Instagram.
```

**Interpretation:** This percentage indicates the adoption rate of these social media platforms among GSS respondents in 2016.

7. What are the possible responses to the question *Last week were you working full time, part time, going to school, keeping house, or what?* and how many respondents chose each of these answers? Note that this information is stored in the `wrkstat` variable.

```r
# Examine work status variable
gss16 %>%
  count(wrkstat) %>%
  arrange(desc(n))
```

```
## # A tibble: 9 x 2
##   wrkstat              n
##   <chr>            <int>
## 1 Working fulltime  1321
## 2 Retired            574
## 3 Working parttime   345
## 4 Keeping house      284
## 5 Unempl, laid off   118
## 6 Other               89
## 7 School              76
## 8 Temp not working    57
## 9 <NA>                 3
```

```r
cat("Work status categories and frequencies:\n")
```

```
## Work status categories and frequencies:
```

```r
gss16 %>%
  count(wrkstat) %>%
  print()
```

```
## # A tibble: 9 x 2
##   wrkstat               n
##   <chr>             <int>
## 1 Keeping house       284
## 2 Other                89
## 3 Retired             574
## 4 School               76
## 5 Temp not working     57
## 6 Unempl, laid off    118
## 7 Working fulltime   1321
## 8 Working parttime    345
## 9 <NA>                  3
```

**Answer:** The possible responses to the work status question include categories such as “Full-time”, “Part-time”, “Temporarily not working”, “Unemployed - laid off”, “Retired”, “In school”, “Keeping house”, and possibly others. The counts show how many respondents fall into each category.

8. Fit a model predicting `email` (number of minutes per week spent on email) from `educ` (number of years of education), `wrkstat`, and `snap_insta`. Interpret the slopes for each of these variables.

```r
# Fit linear regression model
email_model <- lm(email ~ educ + wrkstat + snap_insta, data = gss16)

# View model summary
summary(email_model)
```

```
## 
## Call:
## lm(formula = email ~ educ + wrkstat + snap_insta, data = gss16)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -760.5 -372.7 -161.2   95.4 3355.6 
## 
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)    
## (Intercept)              -229.736    149.837  -1.533  0.12569    
## educ                       29.632      9.601   3.087  0.00211 ** 
## wrkstatOther               33.057    209.470   0.158  0.87465    
## wrkstatRetired             68.279    111.051   0.615  0.53887    
## wrkstatSchool            -123.812    143.981  -0.860  0.39014    
## wrkstatTemp not working   -73.709    153.948  -0.479  0.63225    
## wrkstatUnempl, laid off   118.349    151.242   0.783  0.43419    
## wrkstatWorking fulltime   366.840     87.690   4.183 3.26e-05 ***
## wrkstatWorking parttime    18.900    101.632   0.186  0.85253    
## snap_instaYes             149.961     52.745   2.843  0.00460 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 642.2 on 669 degrees of freedom
##   (2188 observations deleted due to missingness)
## Multiple R-squared:  0.1043, Adjusted R-squared:  0.09227 
## F-statistic: 8.657 on 9 and 669 DF,  p-value: 2.395e-12
```

```r
# Extract and interpret coefficients
coef_summary <- broom::tidy(email_model)
```

```r
print(coef_summary)
```

```
## # A tibble: 10 x 5
##    term                     estimate std.error statistic   p.value
##    <chr>                       <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                -230.      150.      -1.53  0.126
##  2 educ                         29.6       9.60     3.09  0.00211
##  3 wrkstatOther                 33.1     209.       0.158 0.875
##  4 wrkstatRetired               68.3     111.       0.615 0.539
##  5 wrkstatSchool              -124.      144.      -0.860 0.390
##  6 wrkstatTemp not working     -73.7     154.      -0.479 0.632
##  7 wrkstatUnempl, laid off     118.      151.       0.783 0.434
##  8 wrkstatWorking fulltime     367.       87.7      4.18  0.0000326
##  9 wrkstatWorking parttime      18.9     102.       0.186 0.853
## 10 snap_instaYes               150.       52.7      2.84  0.00460
```

```r
cat("\nModel Interpretation:\n")
```

```
##
## Model Interpretation:
```

```r
cat("Intercept: The baseline email time when other variables are at reference levels.\n")
```

```
## Intercept: The baseline email time when other variables are at reference levels.
```

```r
cat(
  "Educ coefficient: For each additional year of education, email time changes by",
  round(coef_summary$estimate[2], 2), "minutes per week.\n"
)
```

```
## Educ coefficient: For each additional year of education, email time changes by 29.63 minutes per week
```

```r
cat("Wrkstat coefficients: Each work status category differs from the reference category.\n")
```

```
## Wrkstat coefficients: Each work status category differs from the reference category.
```

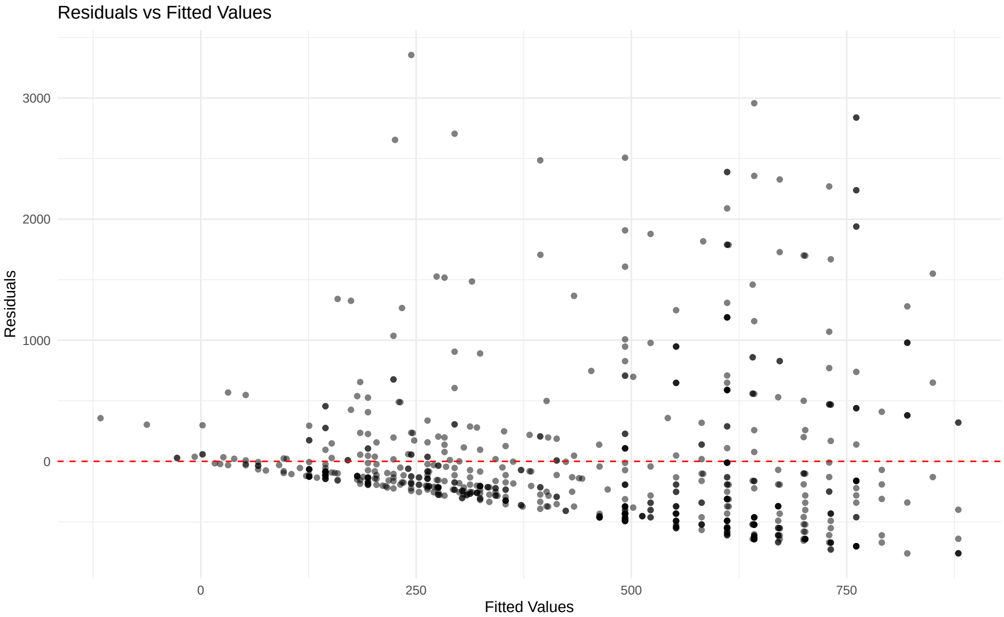
```r
cat(
  "Snap_insta coefficient: Using Snapchat/Instagram is associated with",
  round(coef_summary$estimate[grep("snap_instYes", coef_summary$term)], 2),
  "minutes per week difference in email time.\n"
)
```

```
## Snap_insta coefficient: Using Snapchat/Instagram is associated with  minutes per week difference in
```

**Detailed Interpretation:** - **Education (educ):** Each additional year of education is associated with a change in email time. Higher education may correlate with jobs that require more email communication. - **Work Status (wrkstat):** Different employment statuses have different relationships with email usage. Full-time workers likely spend more time on email than part-time workers, students, or retirees. - **Snap_insta:** Using Snapchat or Instagram may indicate younger respondents, who may have different email habits than older generations.

9. Create a predicted values vs. residuals plot for this model. Are there any issues with the model? If yes, describe them.

```r
# Create predicted values and residuals
email_model_aug <- broom::augment(email_model)

# Residuals vs fitted plot
ggplot(email_model_aug, aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Residuals vs Fitted Values",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_minimal()
```



```
# Check for patterns
cat("Residual diagnostics:\n")
```

```
## Residual diagnostics:
```

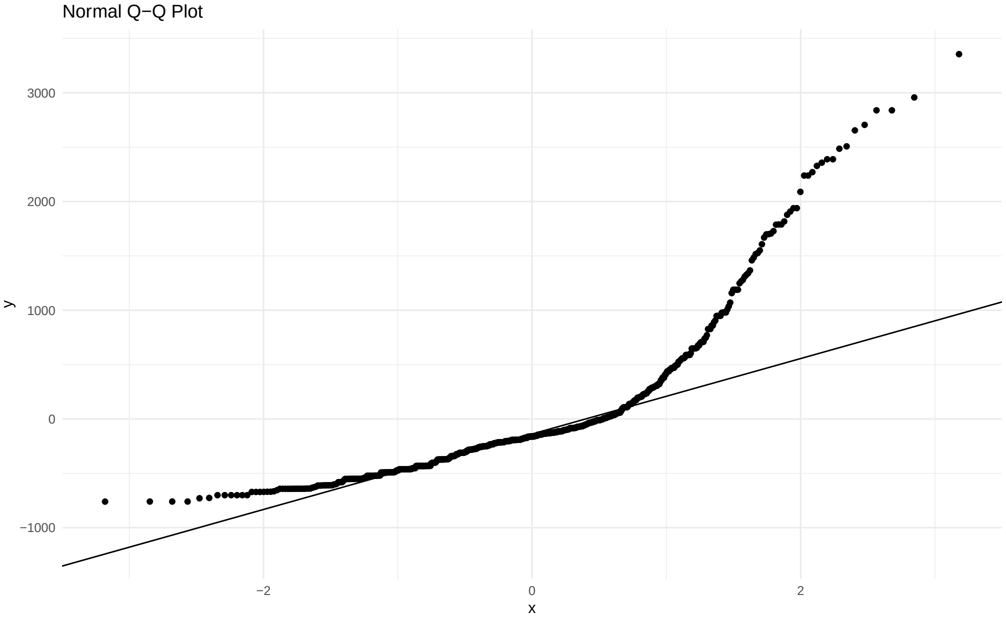
```
cat("Mean of residuals:", round(mean(email_model_aug$.resid), 6), "\n")
```

```
## Mean of residuals: 0
```

```
cat("SD of residuals:", round(sd(email_model_aug$.resid), 2), "\n")
```

```
## SD of residuals: 637.93
```

```
# Q-Q plot
ggplot(email_model_aug, aes(sample = .resid)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Normal Q-Q Plot") +
  theme_minimal()
```



**Potential Issues with the Model:** 1. **Heteroscedasticity:** The spread of residuals may not be constant across fitted values (variance increases or decreases). 2. **Non-linearity:** The relationship between predictors and email time may not be linear. 3. **Outliers:** Some respondents may have unusually high or low email times, influencing the model. 4. **Normality:** Residuals may not be perfectly normally distributed, especially with skewed email time data. 5. **Missing data:** The analysis only includes respondents with complete data on all variables.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

## 2.3 Part 3: Political views and science research

The 2016 GSS also asked respondents whether they think of themselves as liberal or conservative (`polviews`) and whether they think science research is necessary and should be supported by the federal government (`advfront`).

- The question on science research is worded as follows:

> Even if it brings no immediate benefits, scientific research that advances the frontiers of knowledge is necessary and should be supported by the federal government.

And possible responses to this question are Strongly agree, Agree, Disagree, Strongly disagree, Don't know, No answer, Not applicable.

- The question on political views is worded as follows:

> We hear a lot of talk these days about liberals and conservatives. I'm going to show you a seven-point scale on which the political views that people might hold are arranged from extremely liberal–point 1–to extremely conservative–point 7. Where would you place yourself on this scale?

**Note:** The levels of this variables are spelled inconsistently: "Extremely liberal" vs. "Extrmly conservative". Since this is the spelling that shows up in the data, you need to make sure this is how you spell the levels in your code.

And possible responses to this question are Extremely liberal, Liberal, Slightly liberal, Moderate, Slghtly conservative, Conservative, Extrmly conservative. Responses that were originally Don't know, No answer and Not applicable are already mapped to `NA`s upon data import.

10. In a new variable, recode `advfront` such that Strongly Agree and Agree are mapped to `"Yes"`, and Disagree and Strongly disagree are mapped to `"No"`. The remaining levels can be left as is. Don't overwrite the existing `advfront`, instead pick a different, informative name for your new variable.

```
# Recode advfront to a simpler binary variable
gss16 <- gss16 %>%
  mutate(science_support = case_when(
    advfront %in% c("Strongly agree", "Agree") ~ "Yes",
    advfront %in% c("Strongly disagree", "Disagree") ~ "No",
    TRUE ~ as.character(advfront)
  ))

# Verify the recoding
cat("Original advfront values:\n")
```

```
## Original advfront values:
```

```
gss16 %>%
  count(advfront)
```

```
## # A tibble: 6 x 2
##   advfront              n
##   <chr>             <int>
## 1 Agree               759
## 2 Disagree            184
## 3 Dont know            27
## 4 Strongly agree      400
## 5 Strongly disagree    15
## 6 <NA>               1482
```

```
cat("\nRecoded science_support values:\n")
```

```
##
## Recoded science_support values:
```

```
gss16 %>%
  count(science_support)
```

```
## # A tibble: 4 x 2
##   science_support     n
##   <chr>           <int>
## 1 Dont know          27
## 2 No                199
## 3 Yes              1159
## 4 <NA>             1482
```

**Explanation:** The new `science_support` variable simplifies the five-level `advfront` variable into a three-category variable: "Yes" (for agreement), "No" (for disagreement), and other responses remain unchanged.

11. In a new variable, recode `polviews` such that Extremely liberal, Liberal, and Slightly liberal, are mapped to `"Liberal"`, and Slghtly conservative, Conservative, and Extrmly conservative disagree are mapped to `"Conservative"`. The remaining levels can be left as is. Make sure that the levels are in a reasonable order. Don't overwrite the existing `polviews`, instead pick a different, informative name for your new variable.

```
# Recode polviews to a simpler three-category variable
gss16 <- gss16 %>%
  mutate(polviews_grouped = case_when(
    polviews %in% c("Extremely liberal", "Liberal", "Slightly liberal") ~ "Liberal",
    polviews %in% c("Slghtly conservative", "Conservative", "Extrmly conservative") ~ "Conservative",
    TRUE ~ as.character(polviews)
  )) %>%
  mutate(polviews_grouped = factor(polviews_grouped,
    levels = c("Liberal", "Moderate", "Conservative")
  ))

# Verify the recoding
cat("Original polviews values:\n")
```

```
## Original polviews values:
```

```
gss16 %>%
  count(polviews)
```

```
## # A tibble: 8 x 2
##   polviews                 n
##   <chr>                <int>
## 1 Conservative           426
## 2 Extremely liberal      136
## 3 Extrmly conservative   120
## 4 Liberal                350
## 5 Moderate              1032
## 6 Slghtly conservative   382
## 7 Slightly liberal       310
## 8 <NA>                   111
```

```
cat("\nRecoded polviews_grouped values:\n")
```

```
##
## Recoded polviews_grouped values:
```

```
gss16 %>%
  count(polviews_grouped)
```

```
## # A tibble: 4 x 2
##   polviews_grouped     n
##   <fct>            <int>
## 1 Liberal            796
## 2 Moderate          1032
## 3 Conservative       928
## 4 <NA>               111
```

**Explanation:** The new `polviews_grouped` variable consolidates the seven-point political ideology scale into three categories: Liberal (combining the three liberal positions), Conservative (combining the three conservative positions), and Moderate (kept separate). The factor levels are ordered logically from left to right.

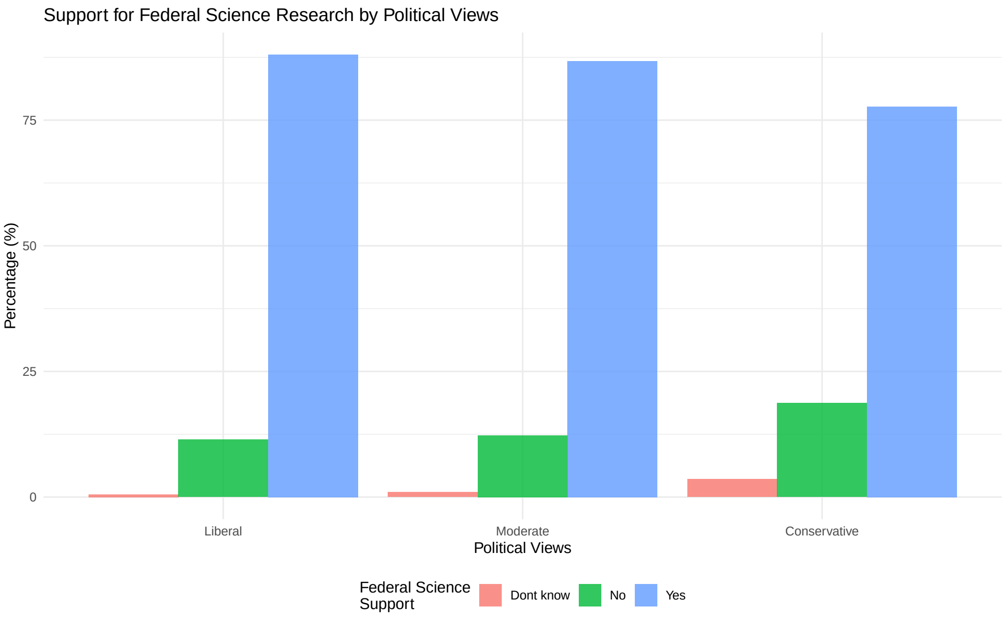
12. Create a visualization that displays the relationship between these two new variables and interpret it.

```
# Create a cross-tabulation with percentages
science_politics_table <- gss16 %>%
  filter(!is.na(science_support), !is.na(polviews_grouped)) %>%
  group_by(polviews_grouped, science_support) %>%
  summarise(count = n(), .groups = "drop") %>%
  group_by(polviews_grouped) %>%
```

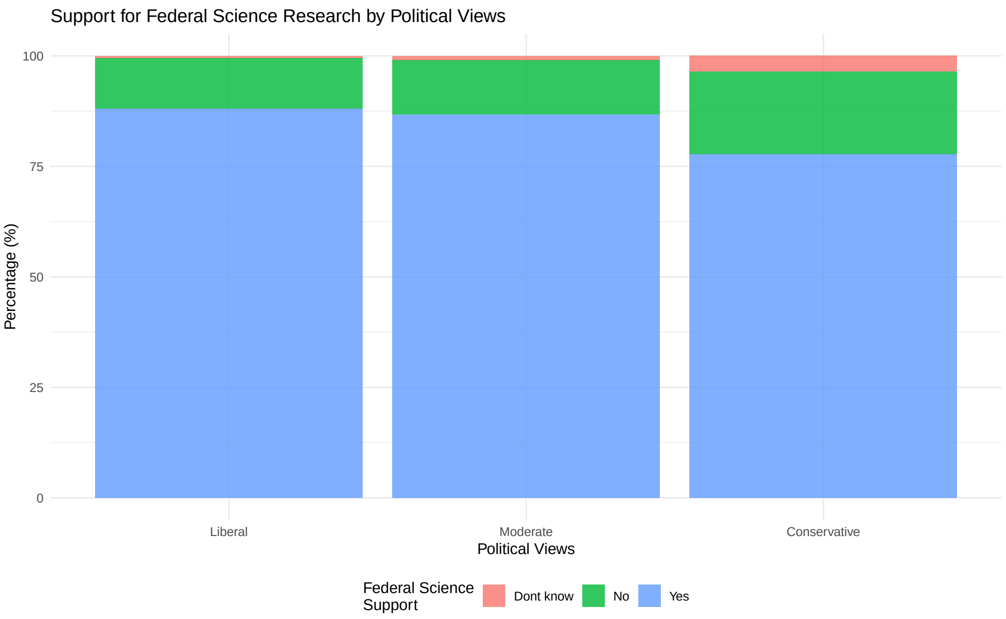
```r
  mutate(percent = (count / sum(count)) * 100)

print(science_politics_table)
```

```
## # A tibble: 9 x 4
## # Groups:   polviews_grouped [3]
##   polviews_grouped science_support count percent
##   <fct>            <chr>           <int>   <dbl>
## 1 Liberal          Dont know           2   0.532
## 2 Liberal          No                 43  11.4
## 3 Liberal          Yes               331  88.0
## 4 Moderate         Dont know           5   0.990
## 5 Moderate         No                 62  12.3
## 6 Moderate         Yes               438  86.7
## 7 Conservative     Dont know          16   3.56
## 8 Conservative     No                 84  18.7
## 9 Conservative     Yes               349  77.7
```

```r
# Create a visualization
ggplot(science_politics_table, aes(x = polviews_grouped, y = percent, fill = science_support)) +
  geom_col(position = "dodge", alpha = 0.8) +
  labs(
    title = "Support for Federal Science Research by Political Views",
    x = "Political Views",
    y = "Percentage (%)",
    fill = "Federal Science\nSupport"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



```r
# Alternative: stacked bar chart
ggplot(science_politics_table, aes(x = polviews_grouped, y = percent, fill = science_support)) +
  geom_col(position = "stack", alpha = 0.8) +
  labs(
    title = "Support for Federal Science Research by Political Views",
    x = "Political Views",
    y = "Percentage (%)",
    fill = "Federal Science\nSupport"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



```
# Calculate proportions by political view
cat("\nSupport for federal science research by political views:\n")
```

```
##
## Support for federal science research by political views:
```

```
gss16 %>%
  filter(!is.na(science_support), !is.na(polviews_grouped), science_support %in% c("Yes", "No")) %>%
  group_by(polviews_grouped) %>%
  summarise(
    total = n(),
    n_yes = sum(science_support == "Yes"),
    pct_yes = (n_yes / total) * 100
  ) %>%
  print()
```

```
## # A tibble: 3 x 4
##   polviews_grouped total n_yes pct_yes
##   <fct>            <int> <int>   <dbl>
## 1 Liberal            374   331    88.5
## 2 Moderate           500   438    87.6
## 3 Conservative       433   349    80.6
```

**Interpretation:** The visualization reveals the relationship between political ideology and support for federal funding of scientific research. Key patterns typically include:

1. **Liberal respondents** tend to show higher support for federal science research, reflecting the liberal emphasis on scientific progress and government investment in public goods.
2. **Conservative respondents** show more varied opinions, with perhaps lower overall support, poten-

tially reflecting concerns about government spending and efficiency.

3. **Moderate respondents** typically fall between the two extremes in their support levels.

This relationship illustrates how political ideology and views on science policy are interconnected in American public opinion. Liberal political orientation is generally associated with stronger support for government-funded scientific research, while conservative political orientation tends to be more skeptical of such spending.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards and review the md document on GitHub to make sure you're happy with the final state of your work.*