Assignment_3_SPAE

Dr. Paul Gronke

2025-10-07

Step 1: Read in the SPAE

1. The SPAE Report is available here https://electionlab.mit.edu/research/projects/survey-performance-american-elections
2. The SPAE Data are avaiable at Datavers https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/GWBAP5

I encourage you to learn how to download data yourself from dataverse, but for the purposes of this lab, I have stored a Stata and SPSS version of the SPAE in the Assignment folder.

To read the SPAE from either Stata or SPSS, you need to have the haven package. We will also load the tidyverse package.

Make sure you make the working directory so that it is "looking" in the "Assignment_3_SPAE" folder.

```
library(tidyverse)
library(haven)
library(janitor) # creates nice tables using the tabyl command and supports piping
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
# Read the SPAE data from Stata
spae_spss <- read_sav("../../data/SPAE/MITU0051_OUTPUT.sav")
```

Step 3: Learn how to take advantage of SPSS labels using the "sjlabelled" package.

You will want to install the "sjlabelled" package if it is not already installed.

```
library(sjlabelled)
```

```
##
## Attaching package: 'sjlabelled'

## The following objects are masked from 'package:haven':
##
##     as_factor, read_sas, read_spss, read_stata, write_sas, zap_labels

## The following object is masked from 'package:forcats':
##
##     as_factor

## The following object is masked from 'package:dplyr':
##
##     as_label

## The following object is masked from 'package:ggplot2':
##
##     as_label

## The following object is masked from 'package:usethis':
##
##     tidy_labels
```

```
table(as_label(spae_spss$Q1))
```

```
##
##            I did not vote in the election this November
##                                                     428
##            I thought about voting this time, but didn't
##                                                     122
##                    I usually vote, but didn't this time
##                                                     130
```

```
##    I tried to vote, but was not allowed to when I tried
##                                                      34
## I tried to vote, but it ended up being too much trouble
##                                                      54
##     I definitely voted in the November General Election
##                                                    9432
##                                                 skipped
##                                                       0
##                                               not asked
##                                                       0
```

```
table(as_label(spae_spss$pid3))
```

```
##
##    Democrat  Republican Independent       Other    Not sure     skipped
##        3945        2832        2678         443         302           0
##   not asked
##           0
```

```
# spae_spss %>%
#   tabyl(as_label(spae_spss$Q1))
#
# spae_spss %>%
#   tabyl(as_label(spae_spss$pid3))
```

Step 4: Learn how to use survey weights in R

R has a package called "survey" that allows you to use survey weights. There is also a package called "srcvyr" that allows you to use the "dplyr" syntax with survey weights.

Finally, there is a package called "gtsummary" that recognizes "survey weighted objects", which means you can make nice tables with survey weighted data.

You should install these if you do not already have them.

Documentation for using survey data in R is provided at https://tidy-survey-r.github.io/tidy-survey-book/

```
library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
library(srvyr)
```

```
##
## Attaching package: 'srvyr'

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following object is masked from 'package:stats':
##
##     filter
```

```
library(gtsummary)

# Do some recodes, theb convert all columns which are "labeled" as factors since these will display
# better in tables in R.

spae_spss <- spae_spss %>%
  mutate(across(
    where(haven::is.labelled),
    ~ haven::as_factor(.x, levels = "labels")
    )
  ) %>%
  mutate(
    pid3 = fct_recode(pid3,
      "Independent" = "Independent",
      NULL = "Other",
      NULL = "Not sure",
      "Democrat"   = "Democrat",
      "Republican" = "Republican"),
    Q39 = case_when(
      Q39 == "I don't know" ~ NA,
      TRUE ~ Q39),
    Q42 = case_when(
      Q42 == "I don't know" ~ NA,
      TRUE ~ Q42),
    Q39_2 = case_when(
      Q39 %in% c("Very confident", "Somewhat confident") ~ "Confident",
      Q39 %in% c("Not too confident", "Not at all confident") ~ "Not confident",
      Q39 == "I don't know" ~ NA),
    Q42_2 = case_when(
      Q42 %in% c("Very confident", "Somewhat confident") ~ "Confident",
      Q42 %in% c("Not too confident", "Not at all confident") ~ "Not confident",
      Q42 == "I don't know" ~ NA)
```

Characteristic	N = 10,200[1]
Voted	
I did not vote in the election this November	444 (4.4%)
I thought about voting this time, but didn't	119 (1.2%)
I usually vote, but didn't this time	131 (1.3%)
I tried to vote, but was not allowed to when I tried	41 (0.4%)
I tried to vote, but it ended up being too much trouble	61 (0.6%)
I definitely voted in the November General Election	9,404 (92%)

[1] n (%)

```
    ) %>%
  mutate(across(where(is.factor), forcats::fct_drop))

# This alternative method converts only specific columns to factors

# spae_spss <- spae_spss %>%
#   mutate(
#     Q1   = haven::as_factor(Q1,   levels = "labels"),
#     pid3 = haven::as_factor(pid3, levels = "labels")
#   )

# Create a survey design object


spae_spss_design <- spae_spss %>%
  as_survey_design(weights = weight_final)

# Create a survey weighted table using gtsummary

# spae_spss_design %>%
#   tabyl(., Q1)
#
spae_spss_design %>%
  select(Q1) %>%
  tbl_svysummary(.)
```

```
spae_spss_design %>%
  select(pid3) %>%
  tbl_svysummary(.)
```

```
# Now show how you can do this in one pipe without creating a separate object

spae_spss_design %>%
  tbl_svysummary(
    include = Q1,
    missing = "no") %>%
  modify_header(stat_0 ~ "**Voted**") %>%
  modify_caption("**Voting Turnout**")
```

Characteristic	N = 10,200[1]
pid3	
Democrat	3,774 (40%)
Republican	3,222 (34%)
Independent	2,489 (26%)
Unknown	714

[1] n (%)

Table 1: **Voting Turnout**

Characteristic	Voted[1]
Voted	
I did not vote in the election this November	444 (4.4%)
I thought about voting this time, but didn't	119 (1.2%)
I usually vote, but didn't this time	131 (1.3%)
I tried to vote, but was not allowed to when I tried	41 (0.4%)
I tried to vote, but it ended up being too much trouble	61 (0.6%)
I definitely voted in the November General Election	9,404 (92%)

[1] n (%)

Step 5: Create a table of voter confidence by Party ID

Examine the Voter Confidence items, calculate differences by PID

```
# Unweighted data.

spae_spss %>%
 tabyl(Q39)
```

```
##                   Q39    n     percent valid_percent
##        Very confident 6844 0.670980392    0.73718225
##    Somewhat confident 2107 0.206568627    0.22694959
##     Not too confident  242 0.023725490    0.02606635
##  Not at all confident   91 0.008921569    0.00980181
##                 <NA>  916 0.089803922            NA
```

```
spae_spss_design %>%
  filter(!is.na(pid3), !is.na(Q39)) %>%
  tbl_svysummary(include = Q39, by = pid3) %>%
  modify_header(label ~ "") %>%
  modify_caption("**Voter Confidence (own vote) by PID**")
```

```
spae_spss_design %>%
  filter(!is.na(pid3), !is.na(Q39)) %>%
  tbl_svysummary(include = Q39_2, by = pid3) %>%
```

Table 2: **Voter Confidence (own vote) by PID**

	Democrat N = 3,505[1]	**Republican** N = 2,991[1]	**Independent** N = 2,184[1]
Confidence (your vote)			
Very confident	2,485 (71%)	2,173 (73%)	1,451 (66%)
Somewhat confident	879 (25%)	760 (25%)	619 (28%)
Not too confident	112 (3.2%)	45 (1.5%)	68 (3.1%)
Not at all confident	28 (0.8%)	13 (0.4%)	45 (2.1%)

[1] n (%)

Table 3: **Voter Confidence (own vote) by PID**

	Democrat N = 3,505[1]	**Republican** N = 2,991[1]	**Independent** N = 2,184[1]
Q39_2			
Confident	3,364 (96%)	2,933 (98%)	2,071 (95%)
Not confident	141 (4.0%)	58 (1.9%)	113 (5.2%)

[1] n (%)

```
  modify_header(label ~ "") %>%
  modify_caption("**Voter Confidence (own vote) by PID**")
```

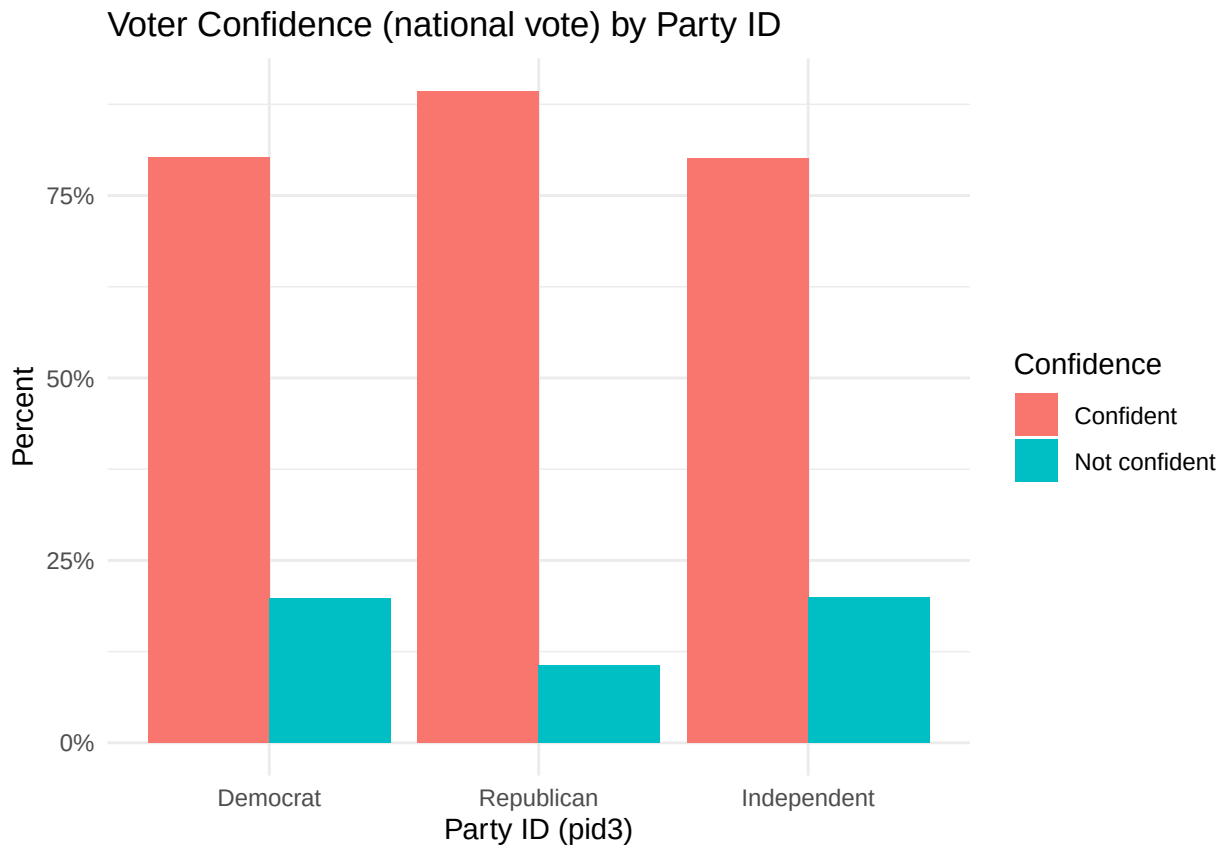
```
spae_spss_design %>%
  filter(!is.na(pid3), !is.na(Q39)) %>%
  tbl_svysummary(include = Q42_2, by = pid3) %>%
  modify_header(label ~ "") %>%
  modify_caption("**Voter Confidence (national vote) by PID**")
```

```
spae_spss_design %>%
  filter(!is.na(pid3), !is.na(Q42_2)) %>%
  group_by(pid3, Q42_2) %>%
  summarize(pct = survey_mean(vartype = NULL, proportion = TRUE)) %>%
  ungroup() %>%
  ggplot(aes(x = pid3, y = pct, fill = Q42_2)) +
  geom_col(position = "dodge") +   # stacked 100% bars
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    x = "Party ID (pid3)",
    y = "Percent",
    fill = "Confidence",
    title = "Voter Confidence (national vote) by Party ID"
  ) +
  theme_minimal()
```

Table 4: **Voter Confidence (national vote) by PID**

	Democrat N = 3,505[1]	**Republican** N = 2,991[1]	**Independent** N = 2,184[1]
Q42_2			
Confident	2,788 (81%)	2,618 (90%)	1,757 (82%)
Not confident	641 (19%)	304 (10%)	380 (18%)
Unknown	75	69	46

[1] n (%)



Voter confidence by states

```
spae_spss_design %>%
  filter(!is.na(inputstate), !is.na(Q42_2)) %>%
  mutate(confident = Q42_2 == "Confident") %>%
  group_by(inputstate) %>%
  summarize(
    pct_confident = survey_mean(confident, vartype = NULL),  # This will only count the confident respo
    .groups = "drop"
  ) %>%
  mutate(`% Confident` = percent(pct_confident, accuracy = 0.1)) %>%
```

```
  select(inputstate, `% Confident`) %>%
  arrange(inputstate) %>%
  kable(
    caption = "Voter Confidence (National Vote) by State",
    booktabs = TRUE,
    align = c("l", "r")
  ) %>%
  kable_styling(full_width = FALSE, latex_options = c("striped", "hold_position")) %>%
  footnote(
    general = "Source: 2024 Survey of the Performance of American Elections (SPAE)",
    threeparttable = TRUE
  )
```

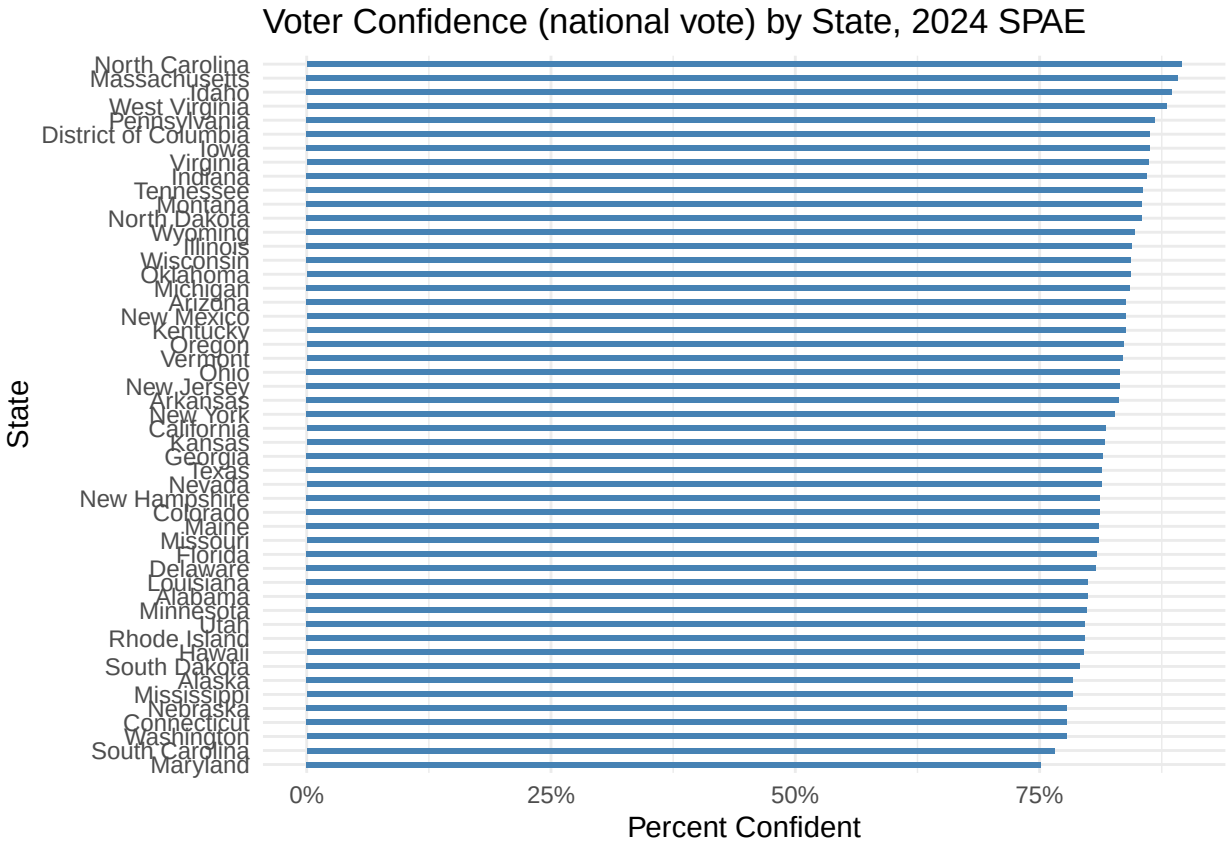
Table 5: Voter Confidence (National Vote) by State

inputstate	% Confident
Alabama	80.0%
Alaska	78.4%
Arizona	83.8%
Arkansas	83.1%
California	81.7%
Colorado	81.2%
Connecticut	77.8%
Delaware	80.8%
District of Columbia	86.3%
Florida	80.9%
Georgia	81.5%
Hawaii	79.5%
Idaho	88.6%
Illinois	84.5%
Indiana	86.0%
Iowa	86.3%
Kansas	81.6%
Kentucky	83.8%
Louisiana	80.0%
Maine	81.1%
Maryland	75.1%
Massachusetts	89.1%
Michigan	84.3%
Minnesota	79.8%
Mississippi	78.3%
Missouri	81.0%
Montana	85.5%
Nebraska	77.8%
Nevada	81.3%
New Hampshire	81.2%
New Jersey	83.2%
New Mexico	83.8%
New York	82.7%
North Carolina	89.5%

North Dakota	85.5%
Ohio	83.2%
Oklahoma	84.3%
Oregon	83.6%
Pennsylvania	86.8%
Rhode Island	79.6%
South Carolina	76.5%
South Dakota	79.1%
Tennessee	85.6%
Texas	81.4%
Utah	79.7%
Vermont	83.5%
Virginia	86.2%
Washington	77.7%
West Virginia	88.0%
Wisconsin	84.4%
Wyoming	84.8%

Note:
Source: 2024 Survey of the Performance of American Elections (SPAE)

```
spae_spss_design %>%
  filter(!is.na(inputstate), !is.na(Q42_2)) %>%
  mutate(confident = Q42_2 == "Confident") %>%
  group_by(inputstate) %>%
  summarize(
    pct_confident = survey_mean(confident, vartype = NULL),  # This will only count the confident respo
    .groups = "drop"
  ) %>%
#  mutate(`% Confident` = percent(pct_confident, accuracy = 0.1)) %>%
  select(inputstate, pct_confident) %>%
   ggplot(aes(x = reorder(inputstate, pct_confident), y = pct_confident)) +
   geom_col(fill = "steelblue", width = 0.4) +
  scale_y_continuous(labels = scales::percent_format()) +
   labs(
     x = "State",
     y = "Percent Confident",
     title = "Voter Confidence (national vote) by State, 2024 SPAE"
   ) +
   theme_minimal() +
   coord_flip()
```


Voter Confidence (national vote) by State, 2024 SPAE
State
North Carolina
Massachusetts
Idaho
West Virginia
Pennsylvania
District of Columbia
Iowa
Virginia
Indiana
Tennessee
Montana
North Dakota
Wyoming
Illinois
Wisconsin
Oklahoma
Michigan
Arizona
New Mexico
Kentucky
Oregon
Vermont
Ohio
New Jersey
Arkansas
New York
California
Kansas
Georgia
Texas
Nevada
New Hampshire
Colorado
Maine
Missouri
Florida
Delaware
Louisiana
Alabama
Minnesota
Utah
Rhode Island
Hawaii
South Dakota
Alaska
Mississippi
Nebraska
Connecticut
Washington
South Carolina
Maryland
0%
25%
50%
75%
Percent Confident