Problem set 2

Siwei Dai

October 8, 2021

NOTE: Start with the file `ps2_2021.Rmd` (available from the github repository at https://github.com/UChicago-pol-methods/IntroQSS-F21/tree/main/assignments). Modify that file to include your answers. Make sure you can "knit" the file (e.g. in RStudio by clicking on the `Knit` button). Submit both the Rmd file and the knitted PDF via Canvas

Question 1: US presidential election results (again)

Download the file "tidy_county_pres_results.csv.zip" from the repository (https://github.com/UChicago-pol-methods/IntroQSS-F21/tree/main/data), unzip it, and put the CSV file in the same directory as your Rmd file.

Then load the data:

```
library(tidyverse)
df <- read_csv("./../data/tidy_county_pres_results.csv")
```

For each US county (uniquely identified by `FIPS` and labeled with `county` and `state`) in each presidential election `year`, we have the total number of votes cast (`total_vote`), number of votes for the Democratic candidate (`dem_vote`), and number of votes for the Republican candidate (`rep_vote`).

(1a) Add a variable called `other_vote_share`, which is the proportion of votes cast for candidates other than the Democratic and the Republican.

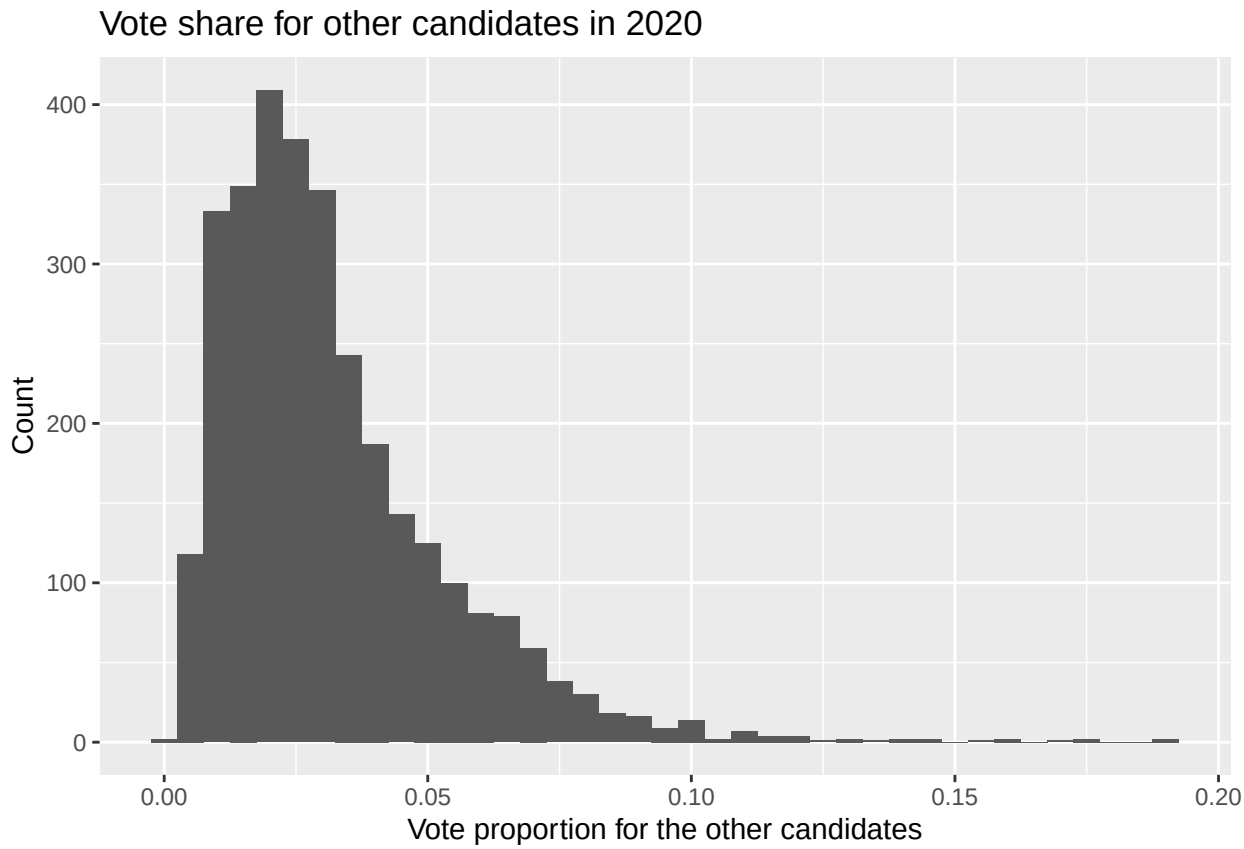
```
# your code here
df <- df %>%
    mutate(other_vote_share = (total_vote - dem_vote - rep_vote) / total_vote)
head(df)
```

```
## # A tibble: 6 x 8
##    FIPS county  state  year total_vote dem_vote rep_vote other_vote_share
##   <dbl> <chr>   <chr> <dbl>      <dbl>    <dbl>    <dbl>            <dbl>
## 1  1001 Autauga AL     1960       2538     1324     1149           0.0256
## 2  1001 Autauga AL     1964       3459        0     2969           0.142
## 3  1001 Autauga AL     1968       7776     1553      606           0.722
## 4  1001 Autauga AL     1972       7140     1593     5367           0.0252
## 5  1001 Autauga AL     1976       9338     4640     4512           0.0199
## 6  1001 Autauga AL     1980      11063     4295     6292           0.0430
```

(1b) Show a histogram of `other_vote_share` in 2000.

```r
# histogram code here
plot1b <- df %>%
    filter(year == 2000) %>%
    ggplot(aes(x = other_vote_share)) +
    geom_histogram(binwidth = 0.005) +
    labs(x = 'Vote proportion for the other candidates', y = 'Count',
         title = 'Vote share for other candidates in 2020')
plot1b
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



(1c) Identify the counties with the highest `other_vote_share` in 2000. Output a table showing the county name, state, and `other_vote_share` for the six counties with the highest `other_vote_share` in 2000. (Don't worry about making the table look nice; just produce the raw `R` output.)

```r
# table code here
tb_1c <- df %>%
    filter(year == 2000) %>%
    select(county, state, other_vote_share) %>%
    arrange(desc(other_vote_share)) %>%
    slice_head(n=6)
tb_1c
```

```
## # A tibble: 6 x 3
```

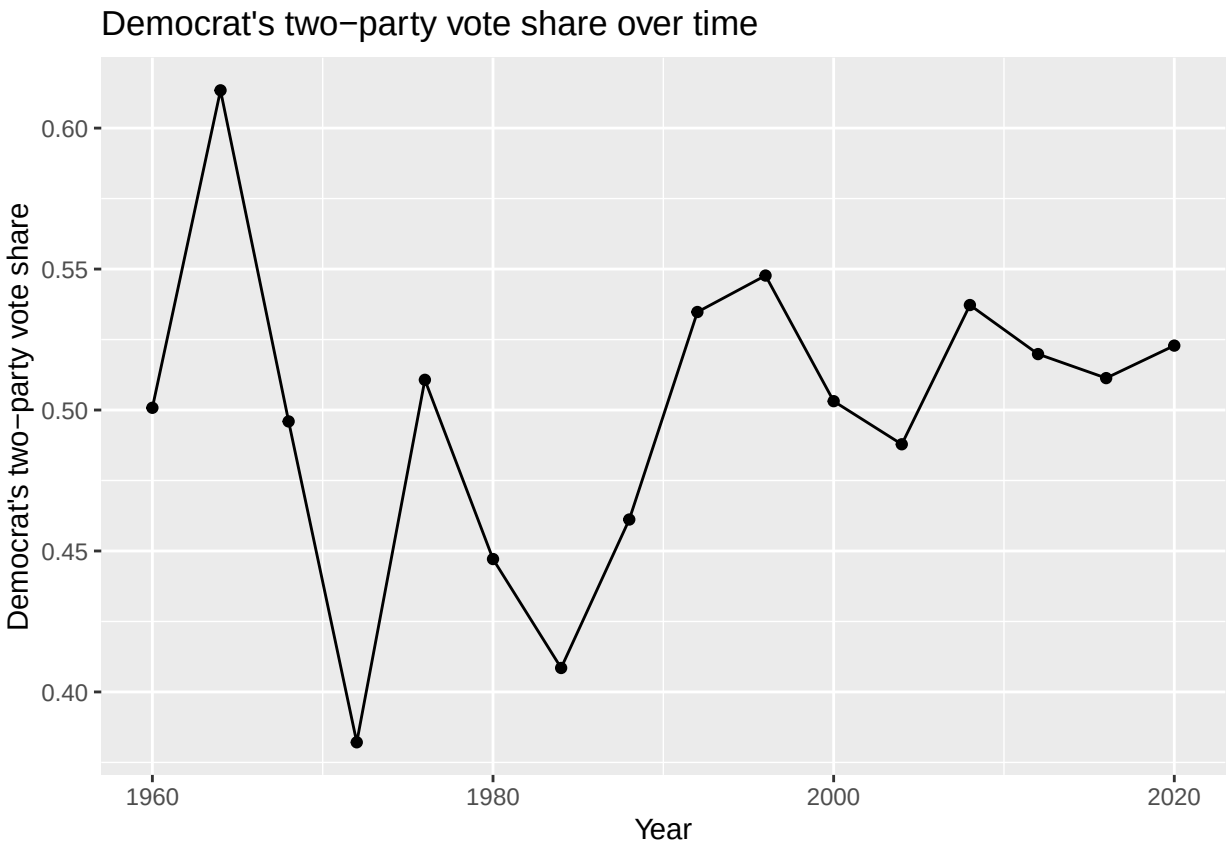
```
##   county     state other_vote_share
##   <chr>      <chr>            <dbl>
## 1 Jefferson  IA               0.190
## 2 San Miguel CO               0.189
## 3 San Juan   CO               0.177
## 4 Grand      UT               0.175
## 5 Missoula   MT               0.169
## 6 Mendocino  CA               0.160
```

(1d) Using `group_by()` and `summarize()`, produce and store a new `tibble` showing the two-party vote share for the Democrat in each election year. ("Two-party vote share for the Democrat" is the votes for the Democrat divided by the votes for either the Democrat or the Republican.) Use it to make a plot showing the Democrats' two-party vote share (vertical axis) across years (horizontal axis).

```
# your code here
tb_1d <- df %>%
    group_by(year) %>%
    summarise('two_party_vote_share' = sum(dem_vote, na.rm = TRUE) /
                  (sum(dem_vote, na.rm = TRUE) + sum(rep_vote, na.rm = TRUE)))

plot_1d <- tb_1d %>%
    ggplot(aes(x = year, y = two_party_vote_share)) +
    geom_point() +
    geom_line() +
    labs(x = "Year", y = "Democrat's two-party vote share",
         title = "Democrat's two-party vote share over time")

tb_1d
```

```
## # A tibble: 16 x 2
##     year two_party_vote_share
##    <dbl>                <dbl>
##  1  1960                0.501
##  2  1964                0.613
##  3  1968                0.496
##  4  1972                0.382
##  5  1976                0.511
##  6  1980                0.447
##  7  1984                0.408
##  8  1988                0.461
##  9  1992                0.535
## 10  1996                0.548
## 11  2000                0.503
## 12  2004                0.488
## 13  2008                0.537
## 14  2012                0.520
## 15  2016                0.511
## 16  2020                0.523
```

```
plot_1d
```



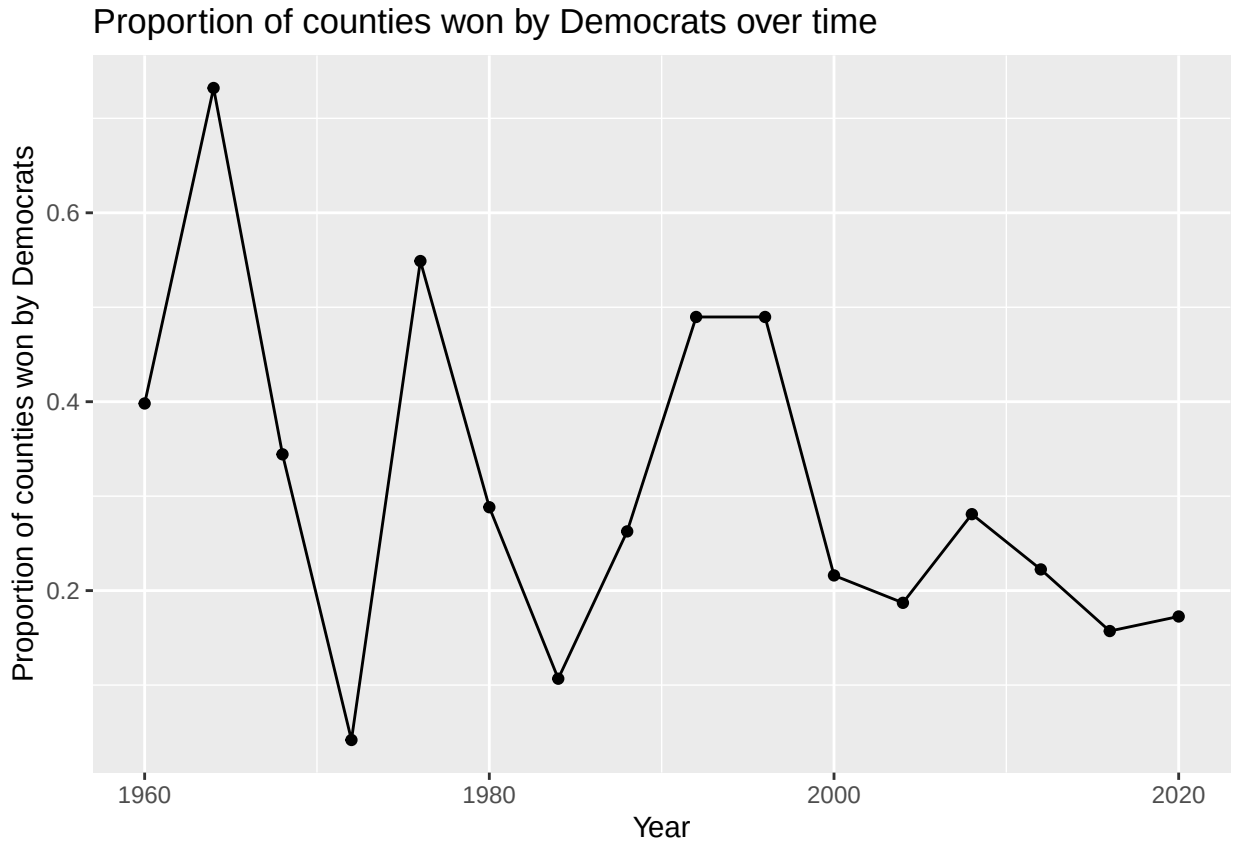
(1e) Using `group_by()` and `summarize()`, produce and store a new `tibble` showing the proportion of counties in which the Democrat got more votes than the Republican in each election year. Use it to make plot showing the share of counties won by the Democrat (vertical axis) across years (horizontal axis).

```
# your code here
tb_1e <- df %>%
    mutate(demwin = (dem_vote > rep_vote)) %>%
    group_by(year) %>%
    summarize(prop_dem_county = mean(demwin, na.rm = TRUE))

plot_1e <- tb_1e %>%
    ggplot(aes(x = year, y = prop_dem_county)) +
    geom_point() +
    geom_line() +
    labs(x = "Year", y = "Proportion of counties won by Democrats",
         title = "Proportion of counties won by Democrats over time")

tb_1e
```

```
## # A tibble: 16 x 2
##     year prop_dem_county
##    <dbl>           <dbl>
##  1  1960           0.398
##  2  1964           0.732
##  3  1968           0.344
```

```
##  4  1972          0.0419
##  5  1976          0.549
##  6  1980          0.288
##  7  1984          0.107
##  8  1988          0.263
##  9  1992          0.490
## 10  1996          0.490
## 11  2000          0.216
## 12  2004          0.187
## 13  2008          0.281
## 14  2012          0.222
## 15  2016          0.157
## 16  2020          0.173
```

```
plot_1e
```



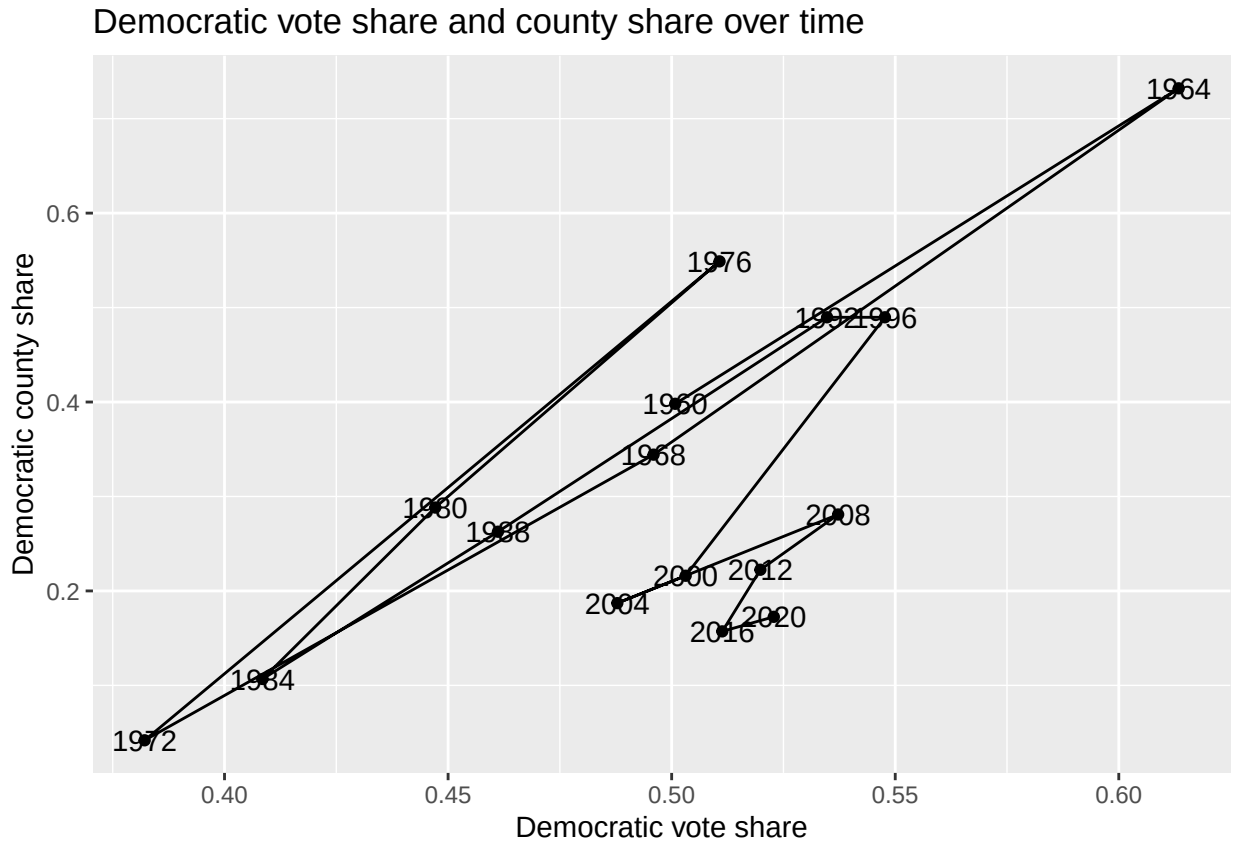
(1f) Use `left_join()` to merge the two tibbles (one with county share, the other with vote share) and store the result. Use this new tibble to plot the Democratic county share (vertical axis) against the Democratic vote share (horizontal axis) over time, as in the last problem set.

```
# your code here
tb_1f <- tb_1d %>%
    left_join(tb_1e, by = 'year')

plot_1f <- tb_1f %>%
    ggplot(aes(x = two_party_vote_share, y = prop_dem_county)) +
```

```
    geom_point() +
    geom_path() +
    geom_text(aes(label = year)) +
    labs(x = 'Democratic vote share', y = 'Democratic county share',
         title = 'Democratic vote share and county share over time')

tb_1f
```

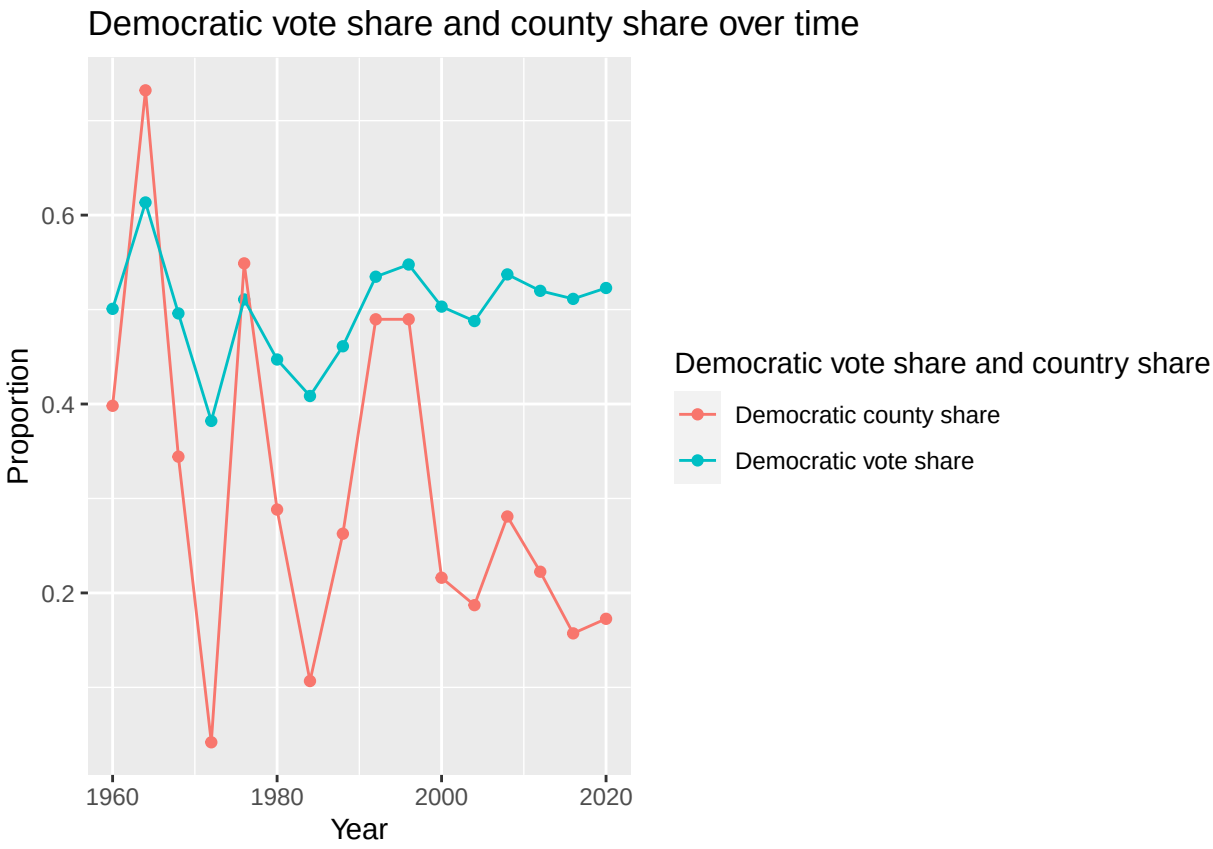
```
## # A tibble: 16 x 3
##     year two_party_vote_share prop_dem_county
##    <dbl>                <dbl>           <dbl>
##  1  1960                0.501          0.398
##  2  1964                0.613          0.732
##  3  1968                0.496          0.344
##  4  1972                0.382          0.0419
##  5  1976                0.511          0.549
##  6  1980                0.447          0.288
##  7  1984                0.408          0.107
##  8  1988                0.461          0.263
##  9  1992                0.535          0.490
## 10  1996                0.548          0.490
## 11  2000                0.503          0.216
## 12  2004                0.488          0.187
## 13  2008                0.537          0.281
## 14  2012                0.520          0.222
## 15  2016                0.511          0.157
## 16  2020                0.523          0.173
```

```
plot_1f
```



(1g) Use `pivot_longer()` to convert the tibble created in the last question to a format appropriate for plotting both the Democratic vote share and the Democratic county share (vertical axis) against the year (horizontal axis) as on the last problem set. Make that plot.

```
# your code here
tb_1g <- tb_1f %>%
    rename('Democratic vote share' = two_party_vote_share,
           'Democratic county share' = prop_dem_county) %>%
    pivot_longer(cols = 2:3,
                 names_to = 'vote_county_share',
                 values_to = 'value')

plot_1g <- tb_1g %>%
    ggplot(aes(x = year, y = value, color = vote_county_share)) +
    geom_point() +
    geom_line() +
    labs(x = 'Year', y = "Proportion",
         title = 'Democratic vote share and county share over time',
         color = 'Democratic vote share and country share')

plot_1g
```

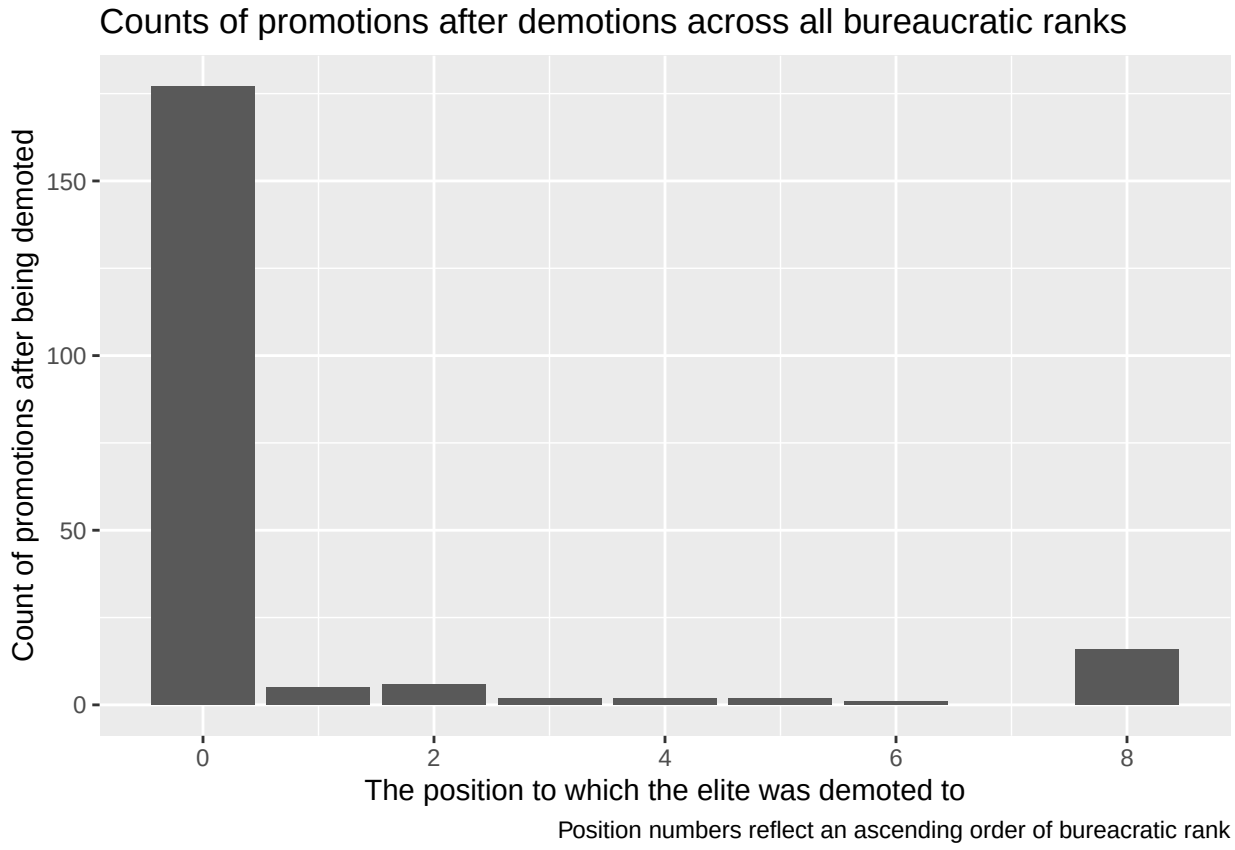


Question 2: independent project data

Choose a dataset that you will use for your independent project. As noted last week, it should have many observations (but not too many) and many variables. And it should be interesting to you! Load the data and make a figure using tools we have learned in class.

```
# Objective: plot the distribution of events in which a Chinese political
# elite who got demoted yet promoted afterwards (vertical axis) against
# the position to which the elite was demoted to (horizontal axis)
# The original dataset is in Simplified Chinese, so I have added a renaming step
# to translate
datapath <- './ps2_data/FullData.csv'
df_ori <- read.csv(datapath)
df_tidy <- df_ori %>%
    as_tibble() %>%
    # select and rename the variables describing the elite's name,
    # position on the hierarchy, and the province he's in
    select(姓名, 级别, 经历序号, 地方一级关键词) %>%
    rename('name' = 姓名,
           'position_ori' = 级别,
           'exper_num' = 经历序号,
           'province' = 地方一级关键词) %>%
    # recode the position from strings into numeric
    mutate('position_numeric' = case_when(position_ori == ' 无级别' ~ 0,
                                          position_ori == ' 小于副处' ~ 1,
```

```
                                        position_ori == ' 副处' ~ 2,
                                        position_ori == ' 正处' ~ 3,
                                        position_ori == ' 副厅' ~ 4,
                                        position_ori == ' 正厅' ~ 5,
                                        position_ori == ' 副部' ~ 6,
                                        position_ori == ' 正部' ~ 7,
                                        position_ori == ' 副国' ~ 8,
                                        position_ori == ' 正国' ~ 9)) %>%
    group_by(name) %>%
    # rank the experiences in the temporal order
    arrange(exper_num, by_group = TRUE) %>%
    # create a new variable to indicate if the elite has been demoted and
    # promoted afterwards
    mutate(demprom = (lag(position_numeric) > position_numeric) &
               (lead(position_numeric) > position_numeric))

plot_2 <- df_tidy %>%
    filter(demprom) %>%
    # support for simplified Chinese is not very well, so I have to use
    # `position_numeric` to represent the rank in the numeric form;
    # the attached caption explains the ranking
    ggplot(aes(x = position_numeric)) +
    geom_bar() +
    labs(x = 'The position to which the elite was demoted to',
         y = 'Count of promotions after being demoted',
         caption = 'Position numbers reflect an ascending order of bureacratic rank',
         title = 'Counts of promotions after demotions across all bureaucratic ranks')

plot_2
```


Counts of promotions after demotions across all bureaucratic ranks
Count of promotions after being demoted
150
100
50
0
0
2
4
6
8
The position to which the elite was demoted to
Position numbers reflect an ascending order of bureacratic rank