Section 01 Introduction

Gov 1347: Election Analytics

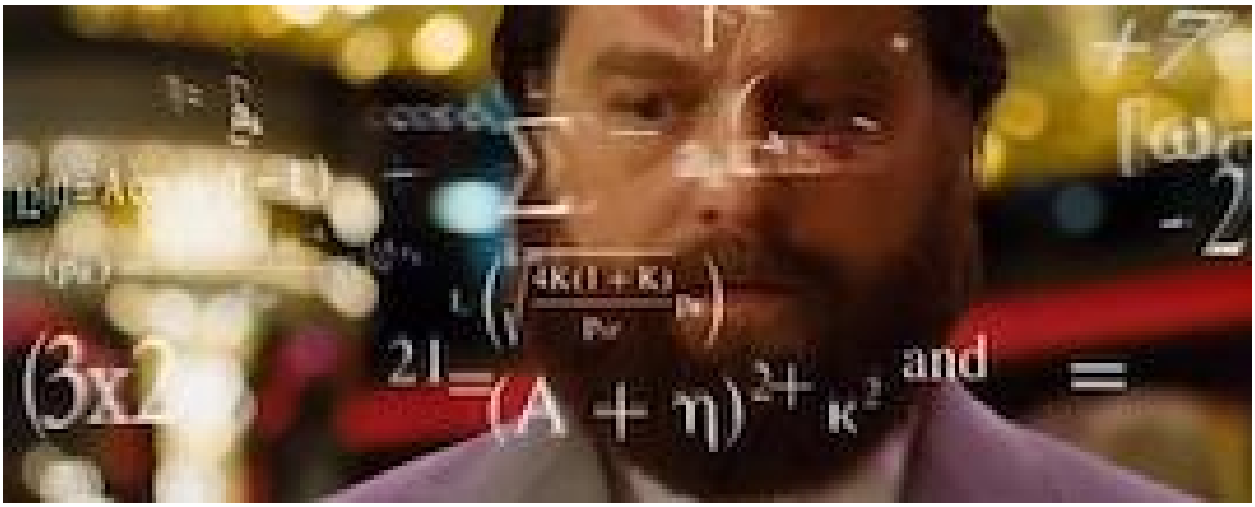
Kiara Hernandez

September 8, 2022

(Re-)Introductions!

*** Icebreaker: name, hometown, favorite piece of media consumed this summer and why**

Today's agenda

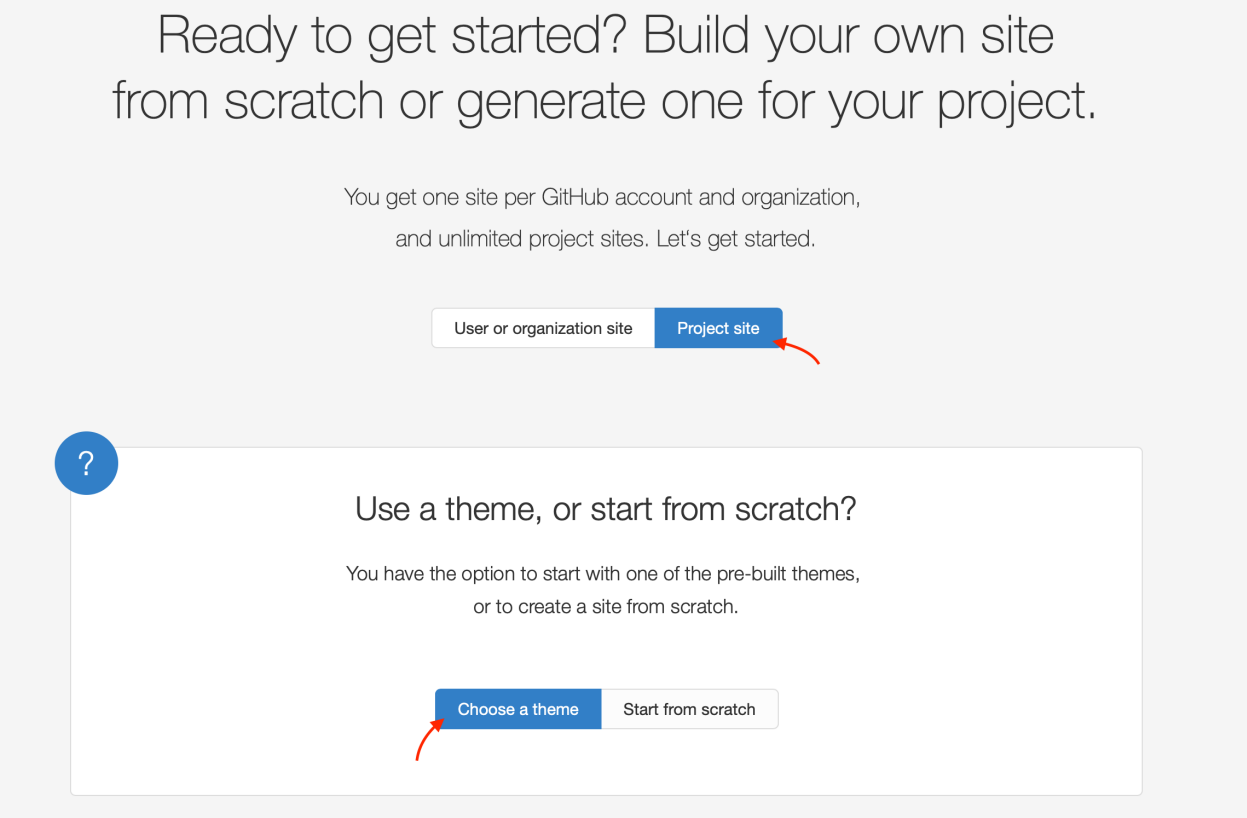


1. **Set-up:** Kick off your career as a *professional* election analyst:
    - Set up your own blog on GitHub
    - Review some key data analysis tools
    - Review what makes a good data visualization
2. **Describing trends in the popular vote:** Build your *first visualizations* to answer two foundational questions:
    - How competitive are midterm elections in the United States?
    - Which states and districts vote blue/red and how consistently?

1. Set-Up

Setting up your election analysis blog

1. Make a *GitHub* account (if you don't have one)
2. Go to https://pages.github.com
3. Select the options below and follow the instructions:



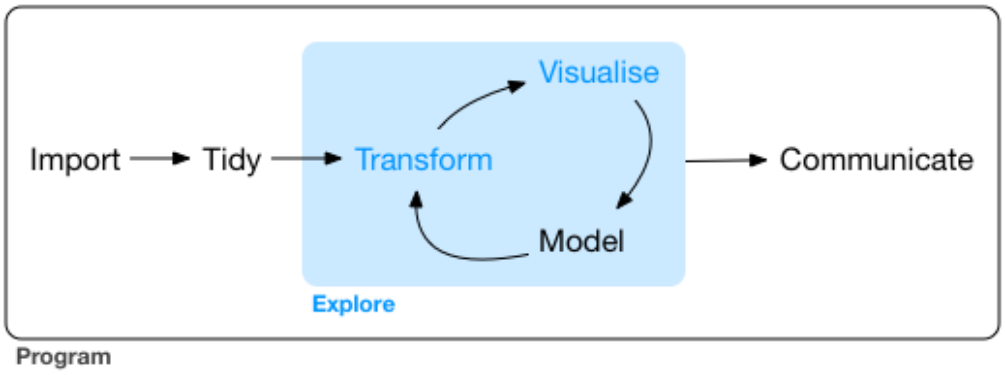
Our (actual) workflow for this class

1. Set your working directory in R to your Github repository for this class.
2. Download the weekly section data from the Google drive (~/Lab sessions/01-Intro (9/8)/Section data https://drive.google.com/drive/folders/1fOPjqzf6d4rVjewC0qP6HTgZKoQEsV47) to your Gov 1347 Github repository.
3. Describe, visualize, and otherwise work with the section data in Rmarkdown.

Our workflow for today

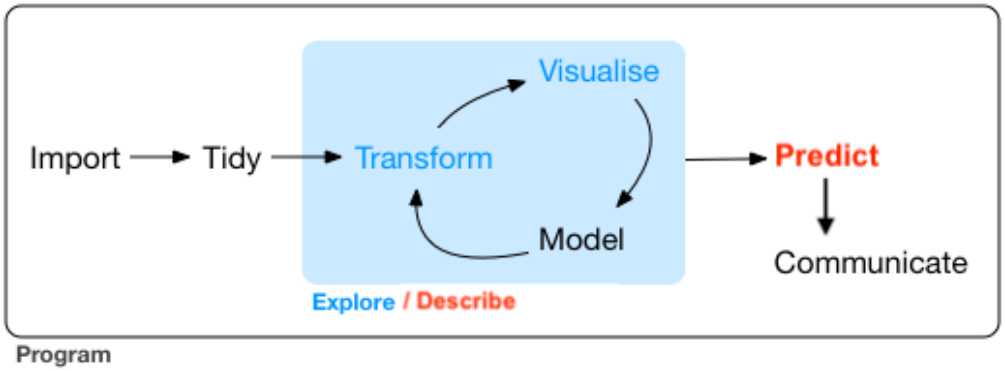
1. Create folder in your file management system for this class and set this as your working directory.
2. Download the weekly section data from the Google drive (~/Lab sessions/01-Intro (9/8)/Section data https://drive.google.com/drive/folders/1fOPjqzf6d4rVjewC0qP6HTgZKoQEsV47) to your Gov 1347 folder.
3. Describe, visualize, and otherwise work with the section data in Rmarkdown.

The R analyst's workflow



- Descriptive inference (vs) Causal inference (vs) Predictive inference

The R election analyst's workflow



Caveat: **descriptive** models aren't always useful for **predictive** inference and vice versa. We will try to be *principled* in how we approach both tasks ... and be careful not to automatically conflate either with **causal** inference.

Fundamentals of analyzing and visualizing election data

What you want to do	R packages	Example functions
Load data	`tidyverse`	`read_csv`
Subset rows or columns	`tidyverse`	`filter`, `select`
Format data (wide or long)	`tidyverse`	`spread`, `gather`
Create new columns	`tidyverse`	`mutate`
Summarise data	`tidyverse`	`group_by`, `summarise`
Merge datasets	`tidyverse`	`join`
Visualizations	`ggplot2`	`geom_line`
Statistical analysis	base R	`lm`, `glm`

We're assuming you have some familiarity with this workflow using `R Studio` from Gov 50 or elsewhere. In this section's analysis, we will review these tools and establish some good practices.

2. Describing Trends in the Popular Vote

Load data (`read_csv`)

We're going to analyze major-party midterm candidates' popular votes from 1948-2020:

First, download the data from the section folder and move it your folder for this class.

```
# set working directory first always
setwd("~/Dropbox/ElectionAnalytics_Midterms/Lab sessions/Intro")
library(tidyverse)
popvote_df <- read_csv("house nationwide vote and seat share by party 1948-2020.csv")
```

```
colnames(popvote_df)
```

```
##  [1] "year"                "AreaAll"             "R_seats"
##  [4] "D_seats"             "Other_seats"         "total_votes"
##  [7] "R_votes"             "D_votes"             "Other_votes"
## [10] "winning_vote_margin" "winner_party"        "R_totalvote_pct"
## [13] "D_totalvote_pct"     "Other_totalvote_pct" "R_majorvote_pct"
## [16] "D_majorvote_pct"
```

```
head(popvote_df[c("year", "winner_party", "winning_vote_margin")])
```

```
## # A tibble: 6 x 3
##    year winner_party winning_vote_margin
##   <dbl> <chr>                      <dbl>
## 1  1948 D                        3149893
## 2  1950 R                         243541
## 3  1952 D                        1395522
## 4  1954 D                        2130550
## 5  1956 D                        1525491
## 6  1958 D                        4508241
```

Subsetting data (`filter`, `select`)

It's always a good idea to spot-check your dataset: `filter` on specific rows and `select` specific columns.

```
popvote_df %>%
  filter(year == 2018) %>%
  select(D_seats, D_majorvote_pct, winner_party)
```

```
## # A tibble: 1 x 3
##   D_seats D_majorvote_pct winner_party
##     <dbl>           <dbl> <chr>
## 1     235            54.4 D
```

Formatting data to wide or long (`spread`, `gather`)

Depending on what you want the unit of analysis to be, you might want to change how the data look.

Currently, long format (the unit of analysis: party-race)

```
popvote_df %>%
    select(year, winner_party, winning_vote_margin) %>%
    filter(year %in% c(1948, 1952, 1956))
```

```
## # A tibble: 3 x 3
##    year winner_party winning_vote_margin
##   <dbl> <chr>                      <dbl>
## 1  1948 D                        3149893
## 2  1952 D                        1395522
## 3  1956 D                        1525491
```

Formatting data to wide or long (spread, gather)

Long to wide format (unit of analysis: race)

```
popvote_wide_df <- popvote_df %>%
  select(year, winner_party, winning_vote_margin) %>%
  spread(key = winner_party, value = winning_vote_margin)

head(popvote_wide_df, 3)
```

```
## # A tibble: 3 x 3
##    year       D      R
##   <dbl>   <dbl>  <dbl>
## 1  1948 3149893     NA
## 2  1950      NA 243541
## 3  1952 1395522     NA
```

Formatting data to wide or long (spread, gather)

Wide to long format (unit of analysis: candidate-race)

```
popvote_wide_df %>%
  gather(key = "winner_party", value = "winning_vote_margin",
         D, R) %>%
  filter(year %in% c(1948, 1952, 1956))
```

```
## # A tibble: 6 x 3
##    year winner_party winning_vote_margin
##   <dbl> <chr>                      <dbl>
## 1  1948 D                        3149893
## 2  1952 D                        1395522
## 3  1956 D                        1525491
## 4  1948 R                             NA
## 5  1952 R                             NA
## 6  1956 R                             NA
```

Modifying our data (many options)

With our wide race-level dataframe, we can code who wins each race:

- With base `R`:

```
popvote_wide_df$winner <- ifelse(
  is.na(popvote_wide_df$D == TRUE),
  "Republican", "Democrat")
```

- With `tidyverse`:

```
popvote_wide_df <- popvote_wide_df %>%
  mutate(winner = case_when(D != "NA" ~ "Democrat",
                            TRUE ~ "Republican"))
```

Summaries of our data (`summarise`)

Now we can ask: from 1948 - 2016, how many races were won by Democrats?

```
popvote_wide_df %>%
  group_by(winner) %>%
  summarise(races = n())
```

```
## # A tibble: 2 x 2
##   winner     races
##   <chr>      <int>
## 1 Democrat      28
## 2 Republican     9
```

Merging datasets together (`*_join`)

Most interesting datasets are actually merges of individual datasets. A hypothetical example is merging in yearly GDP with our race-level data:

```
popvote_and_economy_df <- popvote_wide_df %>%
  left_join(economy_df, by = "year")
```

There are different types of joins depending on how you want to deal with un-merged rows. Ex:

- `left_join` discards un-merged rows originally in `economy_df`
- `inner_join` discards un-merged rows in both dataframes
- `full_join` keeps all the rows in both dataframes

Visualising our data in ggplot2

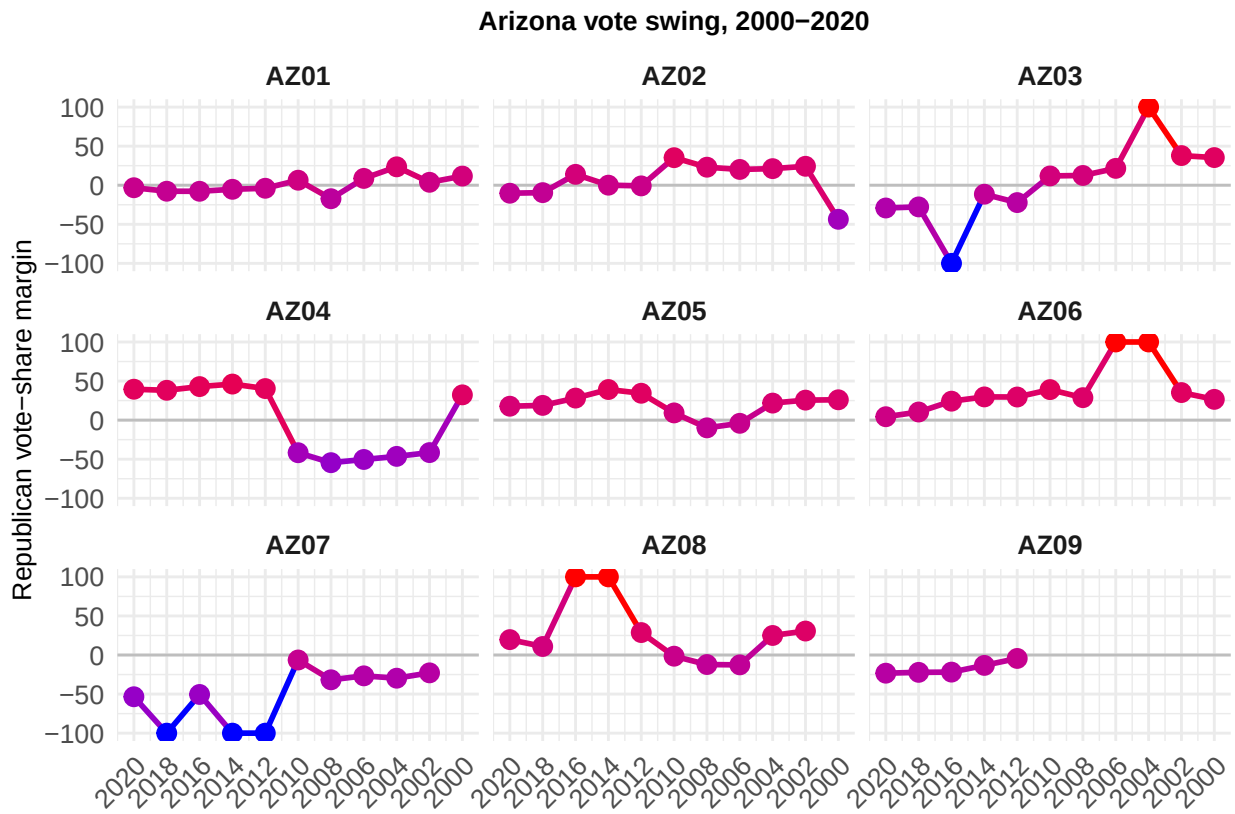
Review: After passing data into `ggplot` command and selecting the type of `geom_*` plot, you must gradually "layer' ' on the plot attributes:

```
library(ggplot2)
ggplot(data = df, mapping = aes(x = x_col, y = y_col)) +
  geom_bar()
```

```
library(ggplot2)
ggplot(data = df, mapping = aes(x = x_col, y = y_col)) +
  geom_bar() +
  xlab("x axis label") +
  ylab("y axis label") +
  ggtitle("my plot")
```

```
library(ggplot2)
ggplot(data = df, mapping = aes(x = x_col, y = y_col)) +
  geom_bar() +
  xlab("x axis label") +
  ylab("y axis label") +
  ggtitle("my plot") +
  theme_classic() +
  theme(axis.text = element_text(size = 10))
```

Example: FiveThirtyEight-esque replication in ggplot2



Interactive Session in R Studio

Geographic and temporal trends in midterm elections?

```
## make map of vote share by state and CD

# start with 114th congress - 2014 election
# required packages
require(tidyverse)
require(ggplot2)
require(sf)

# load geographic data
get_congress_map <- function(cong=114) {
  tmp_file <- tempfile()
  tmp_dir  <- tempdir()
  zp <- sprintf("https://cdmaps.polisci.ucla.edu/shp/districts114.zip",cong)
  download.file(zp, tmp_file)
  unzip(zipfile = tmp_file, exdir = tmp_dir)
  fpath <- paste(tmp_dir, sprintf("districtShapes/districts114.shp",cong), sep = "/")
  st_read(fpath)
}

# load 114th congress
cd114 <- get_congress_map(114)
```

```
## Reading layer `districts114' from data source
##   `/private/var/folders/rm/t20t932n257f8ysr6s45ngdw0000gr/T/RtmpQ2r7iv/districtShapes/districts114.sh
##   using driver `ESRI Shapefile'
## Simple feature collection with 436 features and 15 fields (with 1 geometry empty)
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -179.1473 ymin: 18.91383 xmax: 179.7785 ymax: 71.35256
## Geodetic CRS:  NAD83
```

```
# select specific state
cd114_nj <- cd114 %>%
            filter(STATENAME=="New Jersey") %>%
            mutate(DISTRICT = as.character(DISTRICT))%>%
            select(DISTRICT)

# add data to plot - 2014 GOP party seat share
# reload election data - h from previous exercise
h <- house_party_vote_share_by_district_1948_2020

# filter for 2014 election and state
R_nj_2014 <- h %>%
    filter(raceYear == 2014, State == "New Jersey") %>%
    select(raceYear, State, district_num, RepVotesMajorPercent, DemVotesMajorPercent) %>%
  # summarize party vote share by district
    group_by(district_num) %>%
    summarise(Rep_votes_pct = RepVotesMajorPercent) %>%
```

```
  # rename district variable name to match shapefile
    rename(DISTRICT = district_num)

# before joining dfs, check classes of variable to be merged on
class(R_nj_2014$DISTRICT)
```

```
## [1] "numeric"
```

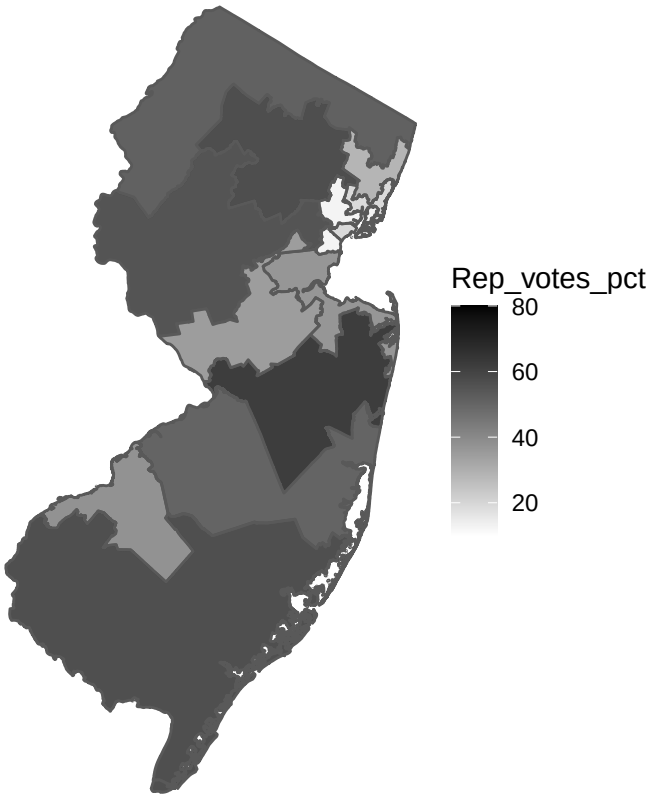
```
class(cd114_nj$DISTRICT)
```

```
## [1] "character"
```

```
# change class
cd114_nj$DISTRICT <- as.numeric(cd114_nj$DISTRICT)

# join election returns with shapefiles
cd114_nj <- cd114_nj %>% left_join(R_nj_2014, by="DISTRICT")
cd114_nj
```

```
## Simple feature collection with 12 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -75.55979 ymin: 38.92852 xmax: -73.90267 ymax: 41.35742
## Geodetic CRS:  NAD83
## First 10 features:
##    DISTRICT Rep_votes_pct                       geometry
## 1         1         40.71 MULTIPOLYGON (((-75.02587 4...
## 2         2         62.25 MULTIPOLYGON (((-75.54269 3...
## 3         3         54.90 MULTIPOLYGON (((-75.031 39....
## 4         4         68.59 MULTIPOLYGON (((-74.75268 4...
## 5         5         56.13 MULTIPOLYGON (((-75.13552 4...
## 6         6         39.38 MULTIPOLYGON (((-74.51921 4...
## 7         7         60.45 MULTIPOLYGON (((-75.20371 4...
## 8         8         19.75 MULTIPOLYGON (((-74.25407 4...
## 9         9         30.52 MULTIPOLYGON (((-74.2062 40...
## 10       10         12.88 MULTIPOLYGON (((-74.30501 4...
```

```
# time to map!
ggplot() +
  geom_sf(data=cd114_nj,aes(fill=Rep_votes_pct),
          inherit.aes=FALSE,alpha=0.9) +
  scale_fill_gradient(low = "white", high = "black", limits=c(10,80)) +
  theme_void() +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
```

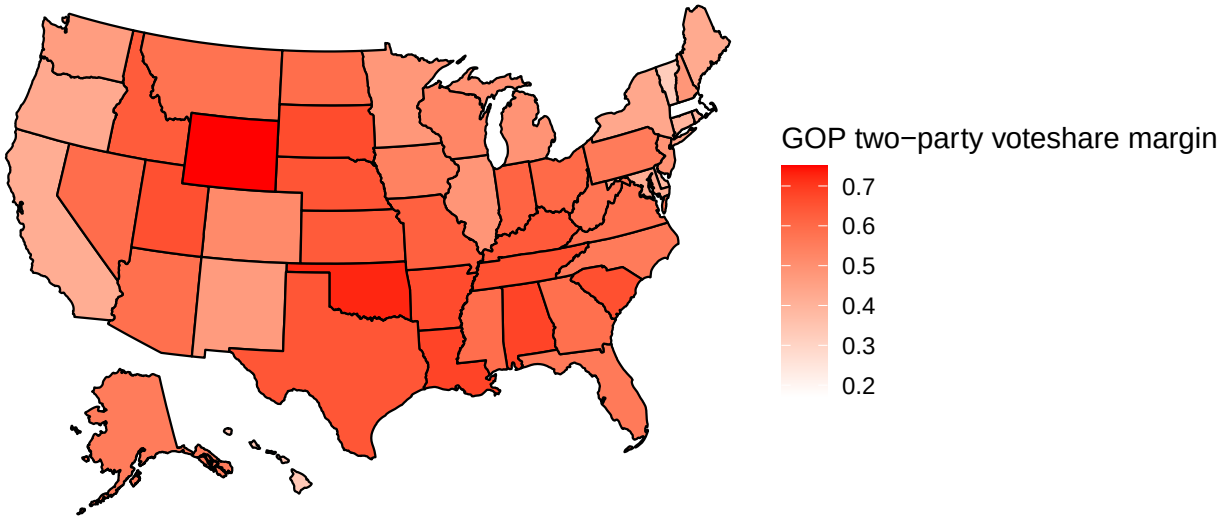



```
## make map of GOP vote share by state (national) - 2014
# use h dataset from earlier
# house_party_vote_share_by_district_1948_2020 <-
  # read_csv("house party vote share by district 1948-2020.csv")
# h <- house_party_vote_share_by_district_1948_2020

# filter for relevant variables
R_2014 <- h %>%
    filter(raceYear == 2014) %>%
    select(raceYear, State, district_num, district_id, RepVotes, DemVotes) %>%
  # summarize party vote share by state
    group_by(State) %>%
  # mutate Rep vote margin by state %>%
    mutate(R_votemargin_st = (sum(RepVotes))/
                             sum(RepVotes + DemVotes),
           D_votemargin_st = (sum(DemVotes))/
                             sum(RepVotes + DemVotes)) %>%
    rename(state = State)

# load usmap
# install.packages('plot_usmap')
library(usmap)

states_map <- usmap::us_map()
unique(states_map$abbr)
```

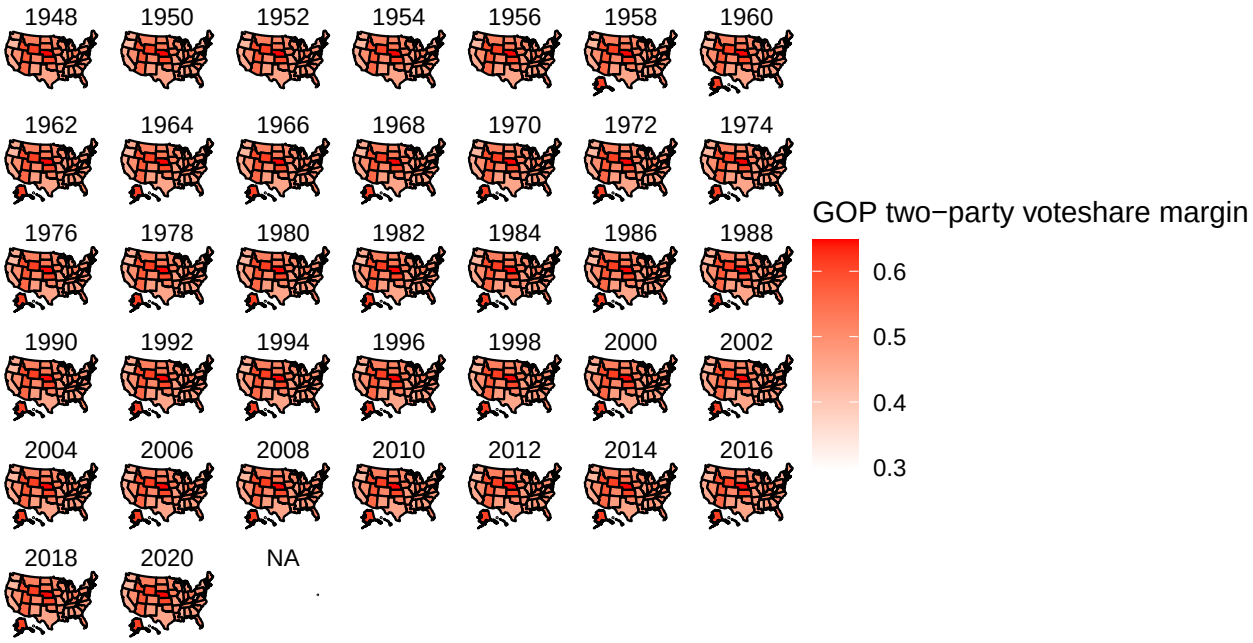
```
##  [1] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "DC" "FL" "GA" "HI" "ID" "IL" "IN"
## [16] "IA" "KS" "KY" "LA" "ME" "MD" "MA" "MI" "MN" "MS" "MO" "MT" "NE" "NV" "NH"
```

```
## [31] "NJ" "NM" "NY" "NC" "ND" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT"
## [46] "VT" "VA" "WA" "WV" "WI" "WY"
```

```
# plot
plot_usmap(data = R_2014, regions = "states", values = "R_votemargin_st") +
  scale_fill_gradient(low = "white", high = "red", name = "GOP two-party voteshare margin") +
  theme_void()
```




```
## map across all election years
# filter for relevant variables
R_all <- h %>%
    select(raceYear, State, district_num, district_id, RepVotes, DemVotes) %>%
  # summarize party vote share by state
    group_by(State) %>%
  # mutate Rep vote margin by state %>%
    mutate(R_votemargin_st = (sum(RepVotes))/
                             sum(RepVotes + DemVotes),
           D_votemargin_st = (sum(DemVotes))/
                             sum(RepVotes + DemVotes)) %>%
    rename(state = State)

# plot
plot_usmap(data = R_all, regions = "states", values = "R_votemargin_st") +
  facet_wrap(facets = raceYear ~.) +
  scale_fill_gradient(low = "white", high = "red", name = "GOP two-party voteshare margin") +
  theme_void()
```



A summary of the district-level exercise (and lots of other helpful information about mapping congressional districts in R) can be found here: https://cdmaps.polisci.ucla.edu/tut/mapping_congress_in_R.html

Final thoughts

- Impossible to memorize every single `dplyr` and `ggplot2` command you'll need.
- Recommendation: create R Studio snippets for various plots we've shown today (`Preferences > Code > Snippets`).
- Visualization is about communication, not coding ... points for **clarity** and **engagement**, not technical wizardry.
- Lots of `ggplot2` tools to creatively communicate: exts.ggplot2.tidyverse.org/gallery/

Blog extensions

1. **Visualization customization.** Use the tools from this section to
    1. create a map of Republican/Democrat voteshare margin by state in a year of your choice,
    2. create a map of Republican/Democrat voteshare margin by state and congressional district in 2014,
    3. label each state (e.g. `AZ`) in your map and
    4. create a custom `ggplot2` theme for your blog.
2. **Gerrymandering extension.** So far, we've been looking at the voteshare margin by party. This means we have been working with the popular vote as opposed to the number of seats won in the House (recall our discussion from Tuesday about the inefficient geographic distribution of Democrats as compared to Republicans that results from gerrymandering — the drawing of district boundaries to give parties electoral advantages. Read more here: https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/, Why Cities Lose: The Deep Roots of the Urban-Rural Political Divide. Jonathan A. Rodden. Basic Books. 2019.).
    - Create a map of seat share by party for the entire U.S. in a year of your choice.

- You can find and download the data at this website: https://guides.library.harvard.edu/hks/campaigns_elections>.
- Select "CQ Voting & Elections Collection" > Election Results > Office:
- House Elections > Election Type: General > Region: National >
- Year: [your choice].
- Similar to the map you created of voteshare margin by state, your map should contain all 50 states shaded according the proportion of seats won by Republicans/Democrats. How do seat share and voteshare compare?

3. **Swing state map extension.** Instead of plotting *voteshare-margin* maps for each year y where the color for a state is colored by the quantity $\frac{R_y}{D_y+R_y}$ (or $\frac{D_y}{D_y+R_y}$), plot a *swing map* for each year where each state is colored by the quantity $\frac{R_y}{D_y+R_y} - \frac{R_{y-4}}{D_{y-4}+R_{y-4}}$. Which states are/have been battleground states? Which are no longer?