US Presidential Election Project

Ian Bogley

7/16/2020

```
library(pacman)
p_load(broom,sf,tidyverse,ggspatial,ggplot2,future.apply,here,usmap,gganimate,magick,plotly)
```

The United States is at a political inflection point in our national history. With the intense polarization that has grown over the past 20 years, I was inspired to create my own humble set of graphics attempting to study the phenomena present in voting behavior.

The process by which we continue through this report shall follow this roadmap: Read in two sets of distinct data. One, a csv file containing raw voter levels for each party in each county for each year during the presidential election. The other, a set of shapefiles providing the boundary lines for every county for each year during the presidential election. It's important to include the different shapefiles because county lines have changed over the past 20 years, oftentimes not at a convenient time for our purposes. Take Broomfield County in Colorado for instance. Having been created in November of 2001, the decennial census of 2000 would fail to accurately represent this change for the 2004 election.

This explains why I decided to utilize multiple sources for our shapefile data. The census shapefiles were useful for the 2000 and 2016 years, but didn't record the 2004, 2008, or 2012 information. As such, I got the 3 remaining years from the "Newman Library" online database (Sources will be explicitly included at the end of the report for citation). Interestingly enough, they actually come with the voter data by county for the presidential election! In order to reduce the chance of discrepencies, however, I will be using the same source for all 5 iterations. Specifically, a dataset from Harvard Dataverse with voter data for each county in the US for every presidential election since 2000.

Let's begin by reading in the voter data now.

```
vote_county_00_16 <-
  here("election/countypres_2000-2016.csv") %>%
  read_csv() %>%
  mutate(
    NAME = gsub(" County","",county),
    party = substr(party,1,3),
    percent = candidatevotes/totalvotes,
    STATE = fips(state),
    county = NULL
  ) %>%
  mutate(
    NAME = gsub("^St. ","Saint ", NAME)
  ) %>% mutate(
    NAME = gsub(" City","", NAME)
  )
```

```
## Parsed with column specification:
```

```
## cols(
##   year = col_double(),
##   state = col_character(),
##   state_po = col_character(),
##   county = col_character(),
##   FIPS = col_double(),
##   office = col_character(),
##   candidate = col_character(),
##   party = col_character(),
##   candidatevotes = col_double(),
##   totalvotes = col_double(),
##   version = col_double()
## )
```

```
vote_county_00_16$NAME[vote_county_00_16$NAME=="Ste. Genevieve"] <- "Sainte Genevieve"
vote_county_00_16$NAME[vote_county_00_16$NAME=="Oglala Lakota"] <- "Shannon"
```

We will be forced to deal with multiple issues where counties require renaming or reformatting. As a default rule I write each abbreviation out, and remove and following descriptors such as "County" or "City".

Now we will read in the shapefiles for drawing out the maps. It was somewhat difficult to track down accurate shapefiles, since the census only provided shapefiles from 2000,2010, and 2013-2018. As such we will reformat non-census shapefiles to be compatible with later functions. This reformatting includes applying the same standards as our voter data above.

The goal is to create 5 shapefiles, 1 for each year. Each will be a multipolygon shapefile with three columns: STATE: The fips code of the state a particular county resides in. This eliminates issues with counties of the same name. NAME: The name of the county following the guidelines above (no abbreviations or following descriptors). Note that the capitalization nor the spacing of the strings matter. This will be addressed later on.

```
us2000_sf_dir  <- here("election/2000")
county_2000_shapefile <- read_sf(
  dsn = us2000_sf_dir
) %>%
  select(STATE,NAME,geometry) %>%
  st_cast("MULTIPOLYGON") %>%
  mutate(
    NAME = gsub("^St. ","Saint ",NAME)
  )
county_2000_shapefile$NAME <- gsub(" County", "",county_2000_shapefile$NAME)
county_2000_shapefile$NAME[county_2000_shapefile$NAME=="Ste. Genevieve"] <- "Sainte Genevieve"
st_crs(county_2000_shapefile) <- "NAD83"

us2004_sf_dir <- here("election/2004")
county_2004_shapefile <- read_sf(
  dsn = us2004_sf_dir
) %>%
  mutate(
    STATE = STATE_FIPS,
    NAME = gsub(" County", "", COUNTY)
  ) %>%
  mutate(
    NAME = gsub(" Parish","",NAME)
```

```
  ) %>%
  mutate(
    NAME = gsub("^St. ","Saint ",NAME)
  ) %>%
  mutate(
    NAME = gsub(" City","", NAME)
  ) %>%
  select(STATE,NAME,geometry)


us2008_sf_dir <- here("election/2008")
county_2008_shapefile <- read_sf(
  dsn = us2008_sf_dir
) %>%
  mutate(
    STATE = STATE_FIPS,
    NAME = gsub(" County", "", COUNTY)
  ) %>%
  mutate(
    NAME = gsub(" Parish","",NAME)
  ) %>%
  mutate(
    NAME = gsub(" City","", NAME)
  ) %>%
  select(STATE,NAME,geometry)
county_2008_shapefile$NAME[county_2008_shapefile$NAME == "DeBaca"] <- "De Baca"

us2012_sf_dir <- here("election/2012")
county_2012_shapefile <- read_sf(
  dsn = us2012_sf_dir
) %>%
  mutate(
    STATE = STATE_FIPS,
    NAME = gsub(" County", "", COUNTY)
  ) %>%
  mutate(
    NAME = gsub(" Parish","",NAME)
  ) %>%
  mutate(
    NAME = gsub("^St. ", "Saint ", NAME)
  ) %>%
  mutate(
    NAME = gsub("^St ", "Saint ", NAME)
  ) %>%
  mutate(
    NAME = gsub(" City","", NAME)
  ) %>%
  select(STATE,NAME,geometry)
county_2012_shapefile$NAME[county_2012_shapefile$NAME == "Gd. Traverse"] <- "Grand Traverse"
county_2012_shapefile$NAME[county_2012_shapefile$NAME == "Lewis & Clark"] <- "Lewis and Clark"
county_2012_shapefile$NAME[county_2012_shapefile$NAME == "King & Queen"] <- "King and Queen"
county_2012_shapefile$NAME[county_2012_shapefile$NAME == "Ste. Genevieve"] <- "Sainte Genevieve"
```

```
us2016_sf_dir <- here("election/2016")
county_2016_shapefile <- read_sf(
  dsn = us2016_sf_dir
) %>%
  mutate(STATE = STATEFP) %>%
  mutate(
    NAME = gsub(" Parish","",NAME)
  ) %>%
  mutate(
    NAME = gsub("^St. ", "Saint ", NAME)
  ) %>%
  mutate(
    NAME = gsub("^St ", "Saint ", NAME)
  ) %>%
  mutate(
    NAME = gsub(" City","", NAME)
  ) %>%
  select(STATE,NAME,geometry)
county_2016_shapefile$NAME[county_2016_shapefile$NAME == "Doña Ana"] <- "Dona Ana"
county_2016_shapefile$NAME[county_2016_shapefile$NAME=="Oglala Lakota"] <- "Shannon"
county_2016_shapefile$NAME[county_2016_shapefile$NAME == "Ste. Genevieve"] <- "Sainte Genevieve"
```

Now we will write a function to create a final version of each year's shapefile, which will be used to plot the final visuals. These 'complete shapefiles' will restructure the data to be more compatible with our end goal. For example, while in the source data used to track voting levels, the votes from different parties are seperated into rows. To help with the continuity of the animations and for our own organization, we will create seperate columns denoting the ratios between the raw level of voters for a single party against the total votes cast in the said county.

Now on to how my beautiful contraption works: The inputs required is a set of voting data for a period of time denoted by the name of the shapefile. For example, the current sourced voter data contains a range from 2000-2016. This means that the function will be applicable between the sourced data and any shapefile from 2000-2016.

It begins by extracting the year we will be plotting. Since the voter data we have contains statistics from multiple years, I narrow down the year by the name of the shapefile. This is possible because we named each shapefile in part by the year it denotes: i.e. county_2000_shapefile is the shapefile for 2000.

After filtering the voter data to the year required, we then create new columns in both the shapefile and voter data. Within, we include the lowercase and collapsed version of the string present in the NAME columns. This eliminates issues with capitalization or spacing if we join the datasets through these columns.

Next, we split the data into "party_data_list". This objects splits the voter data by party into seperate dataframes, which is useful for combining the different parties in the same dataframe. We continue towards this goal by isolating columns from each of the datasets denoting the county ratio for the party. Having successfully renamed the columns we wish to join, as well as retaining their respective counties, we merge each of the columns together ysing a stack overflow submission "my_merge" function.

We are left with 'year_data_final', the final version of our voter data. It includes each county in the mainland US and a single statistic for each party, denoting the ratio between presidential votes for their party versus the total votes cast. Finally, we merge these with our shapefile, creating a final, complete_shapefile.]

The intention behind creating this as a function, is to enable future expansion of the graphics, so long as they are reformatted similarly to these original five shapefiles and single aggregated voter dataset.

```
sf_completion_function_11 <- function(vote_county,shapefile) {
  shapefile_name <- deparse(substitute(shapefile))
  sf_year <- gsub("\\D","",shapefile_name) %>% as.numeric()
  shapefile$county_name <- gsub(" ","",tolower(shapefile$NAME))

  vote_county$party[is.na(vote_county$party)] <- "na"
  vote_county$candidatevotes[is.na(vote_county$candidatevotes)] <- 0
  vote_county$county_name <- gsub(" ","",tolower(vote_county$NAME))

  vote_year_data <- vote_county %>%
    filter(year == sf_year)

  party_data_list <- vote_year_data %>%
    split(vote_year_data$party)

  column_isolate <- function(data_list) {
    new_column <- data_list %>%
      select(year,STATE,county_name,percent)

    colnames(new_column)[4] <- paste(data_list$party[1])

    final_column <- cbind(new_column)

    return(final_column)
  }

  party_columns <- lapply(
    party_data_list[1:length(party_data_list)],
    FUN = column_isolate
  )

  my_merge <- function(df1, df2){
    merge(df1, df2, by = c("year","county_name","STATE"))
  }

  year_data_final <- Reduce(my_merge,party_columns) %>% tibble
  year_data_final[is.na(year_data_final)] <- 0

  complete_shapefile <- left_join(shapefile,year_data_final, by = c("county_name","STATE")) %>%
    na.omit() %>% st_as_sf()
  return(complete_shapefile)
}
```

Now we will create the final shapefiles used in the visuals by running the shapefiles and our voter data through the completion function written above.

Notice that I also disregard the green party in these visuals. Having only been a part of the 2000 election in this sample, I felt it valid to only consider the changes in democratic and republican votes.

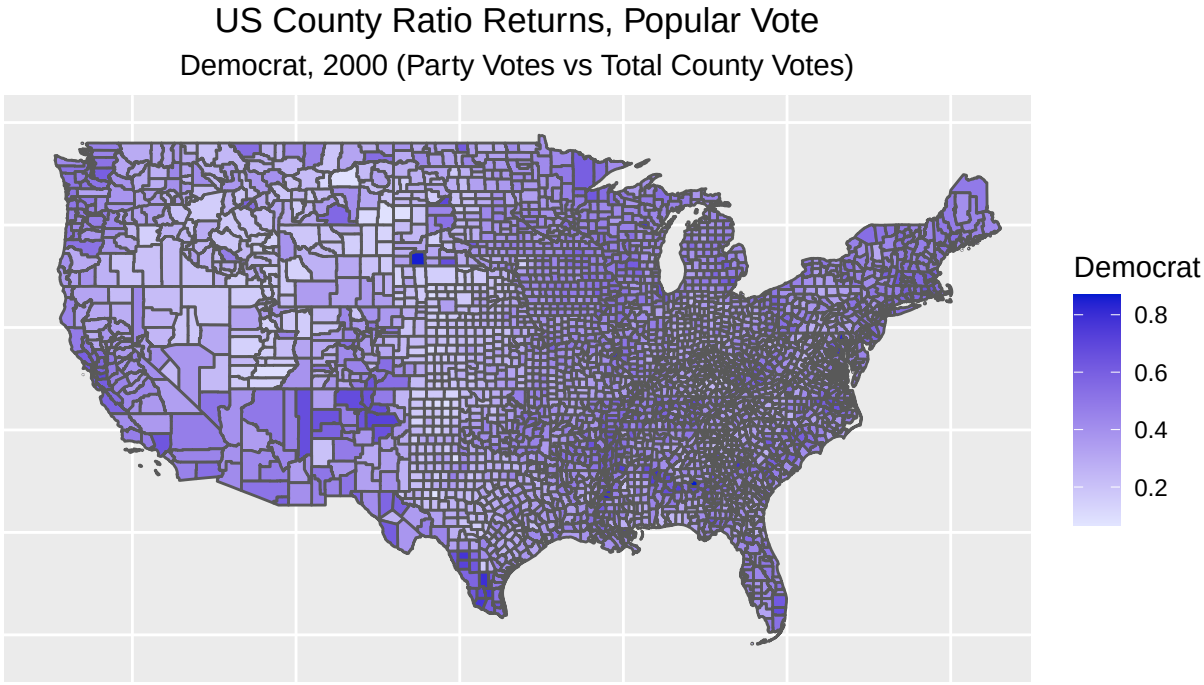
Shapefiles from 2000, 2016 are from the US Census, while 2004-2012 are from: https://www.baruch.cuny.edu/confluence/pages/viewpage.action?pageId=35442824

```
vote_county_00_total <- sf_completion_function_11(vote_county_00_16,county_2000_shapefile)
vote_county_00_final <- vote_county_00_total %>% select(-gre)
```

```
vote_county_04_final <- sf_completion_function_11(vote_county_00_16,county_2004_shapefile)
vote_county_08_final <- sf_completion_function_11(vote_county_00_16,county_2008_shapefile)
vote_county_12_final <- sf_completion_function_11(vote_county_00_16,county_2012_shapefile)
vote_county_16_final <- sf_completion_function_11(vote_county_00_16,county_2016_shapefile)
```

To see what a single graph looks like, lets plot the democratic ratios by county in 2000. Let's also define our map boundaries by the variable Mainland_US.

```
Mainland_US <- coord_sf(xlim = c(-125,-68), ylim = c(24,50))

plot_2000_dem <- vote_county_00_final %>%
  ggplot() +
  geom_sf(aes(fill = dem)) +
  Mainland_US +
  scale_fill_gradient(low = "#E2E5FF",high = "#011AD0") +
  labs(
    title = "US County Ratio Returns, Popular Vote",
    subtitle = "Democrat, 2000 (Party Votes vs Total County Votes)",
    fill = "Democrat"
    ) +
  theme(
    plot.title = element_text(hjust = .5),
    plot.subtitle = element_text(hjust = .5),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  )
plot_2000_dem
```

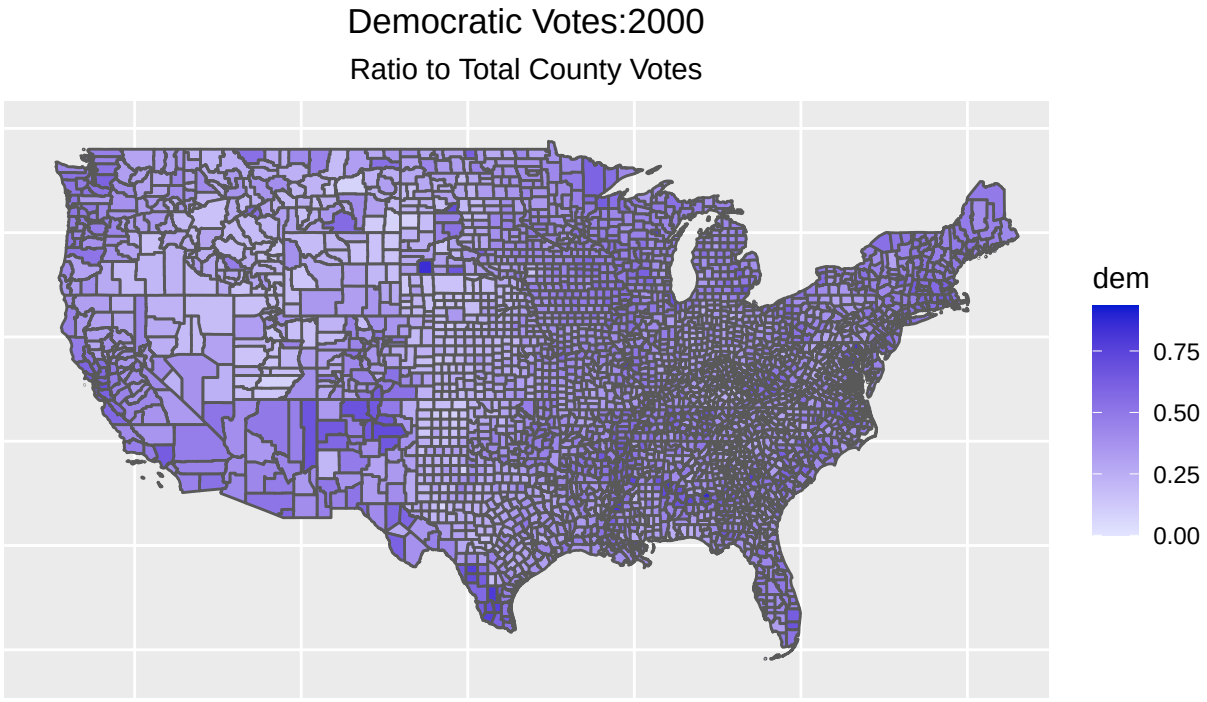


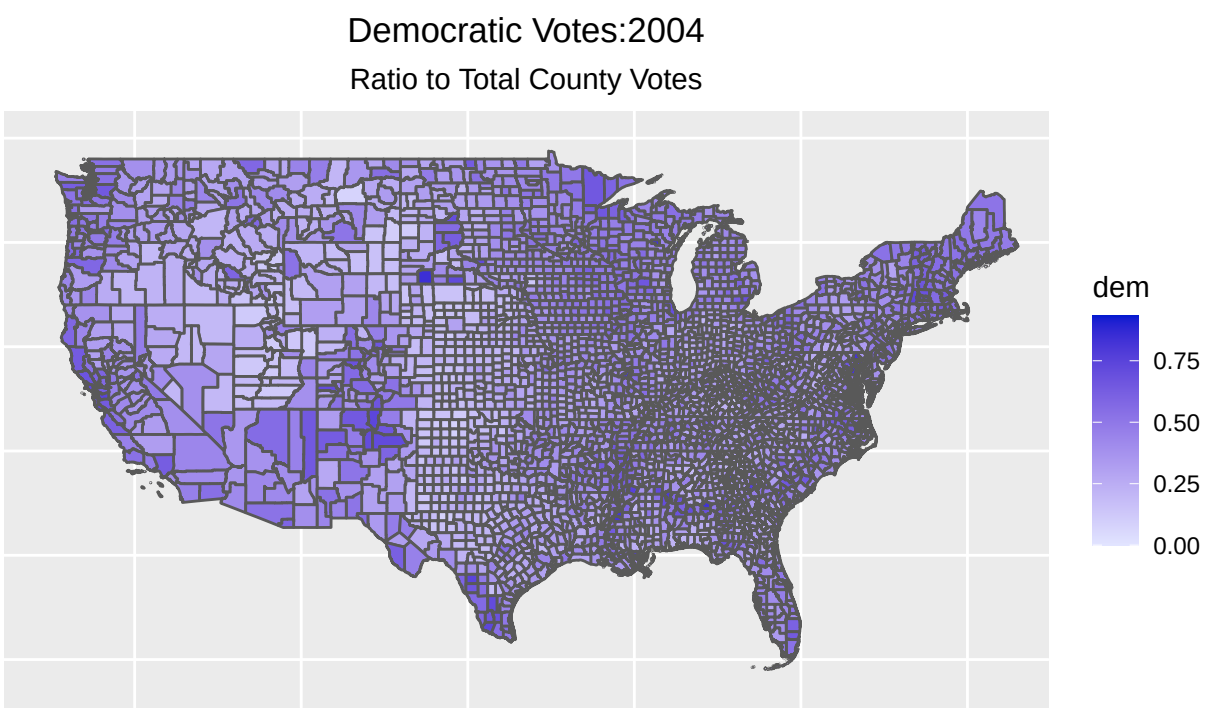
Now we will attempt to combine the different graphs into one figure. To do so, we will join all the final shapefiles we created into a single aggregated database. Then we will use the gganimate package to generate gifs of each year's voting patterns. This first animation will simply be showing the county ratio between total votes and democratic votes.

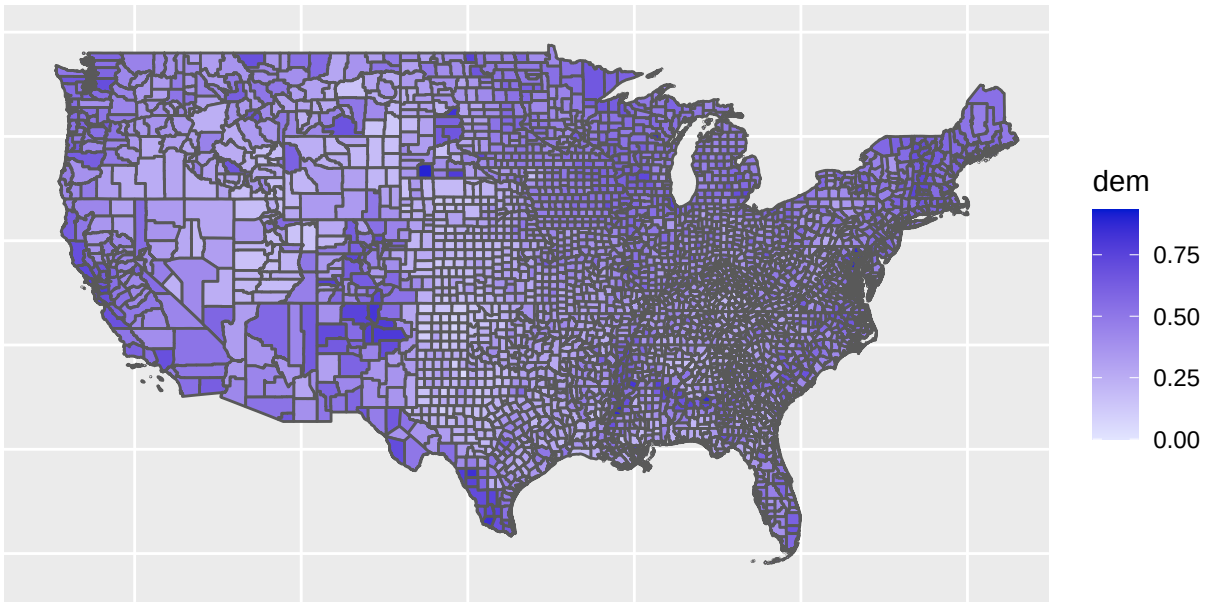
```
animation_data <- rbind(vote_county_00_final,vote_county_04_final,vote_county_08_final,vote_county_12_f

dem_animation <- animation_data %>%
  ggplot() +
  geom_sf(aes(fill = dem)) +
  Mainland_US +
  transition_manual(year) +
  labs(
    title = "Democratic Votes:{current_frame}",
    subtitle = "Ratio to Total County Votes"
    ) +
  theme(
    plot.title = element_text(hjust = .5),
    plot.subtitle = element_text(hjust = .5),
    axis.text = element_blank(),
    axis.ticks = element_blank()
    ) +
  scale_fill_gradient(
    low = "#E2E5FF",
    high = "#011AD0"
  )
```

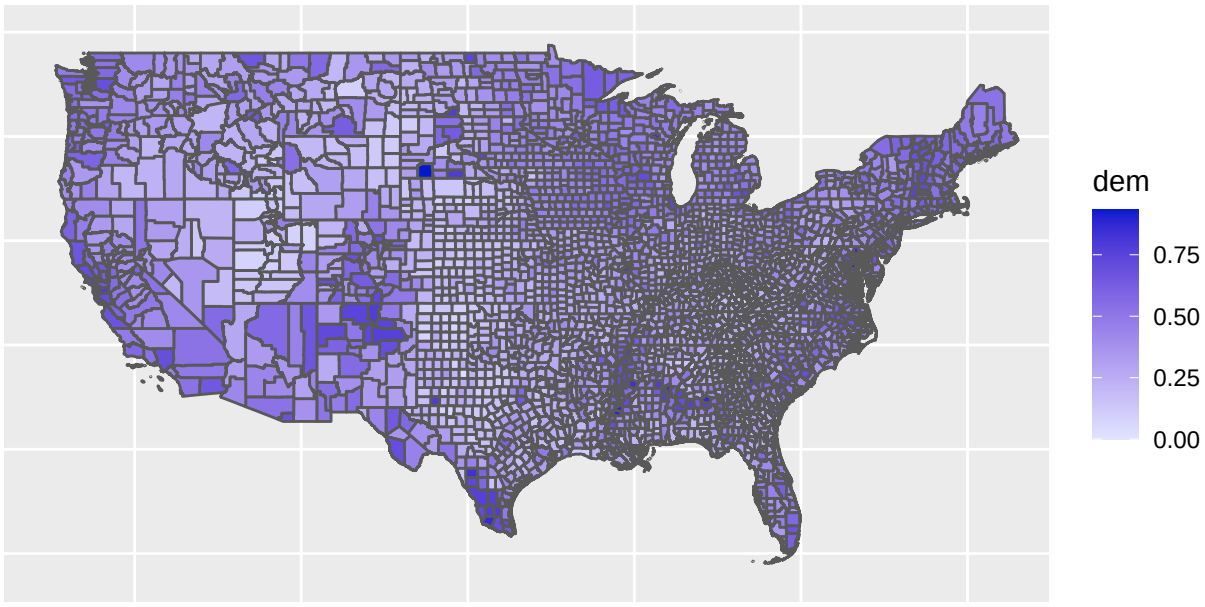
```
dem_animation
```

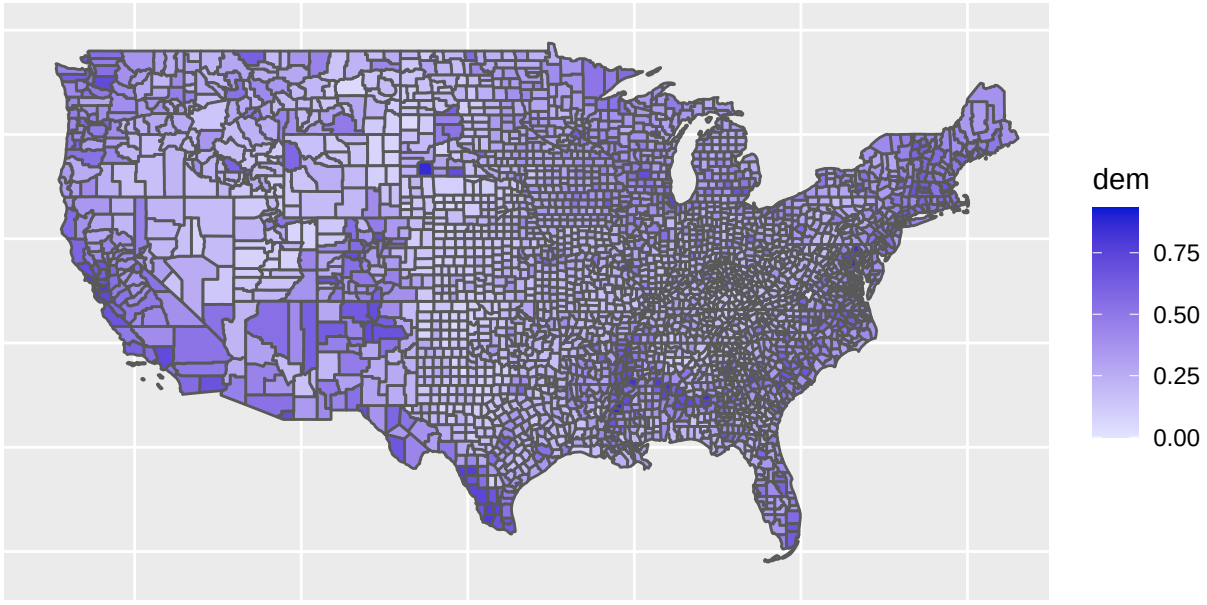
```
## nframes and fps adjusted to match transition
```




Democratic Votes:2004
Ratio to Total County Votes
dem
0.75
0.50
0.25
0.00


Democratic Votes:2008
Ratio to Total County Votes
dem
0.75
0.50
0.25
0.00

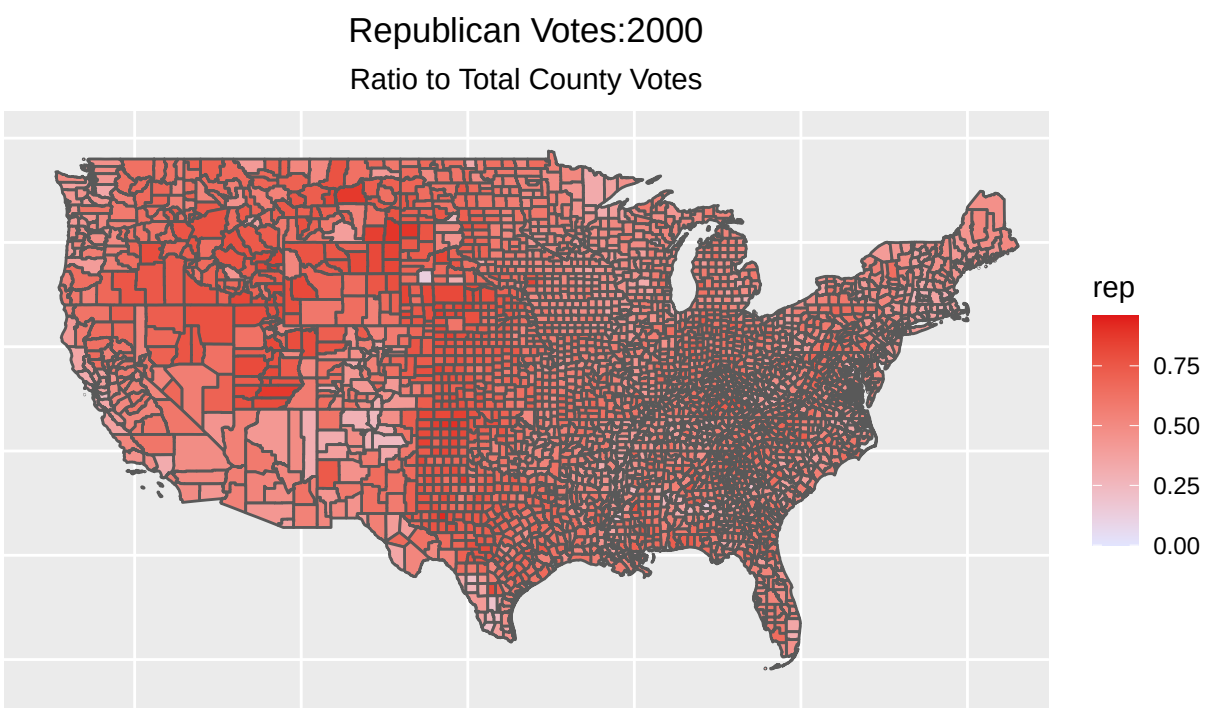

Democratic Votes:2012
Ratio to Total County Votes
dem
0.75
0.50
0.25
0.00

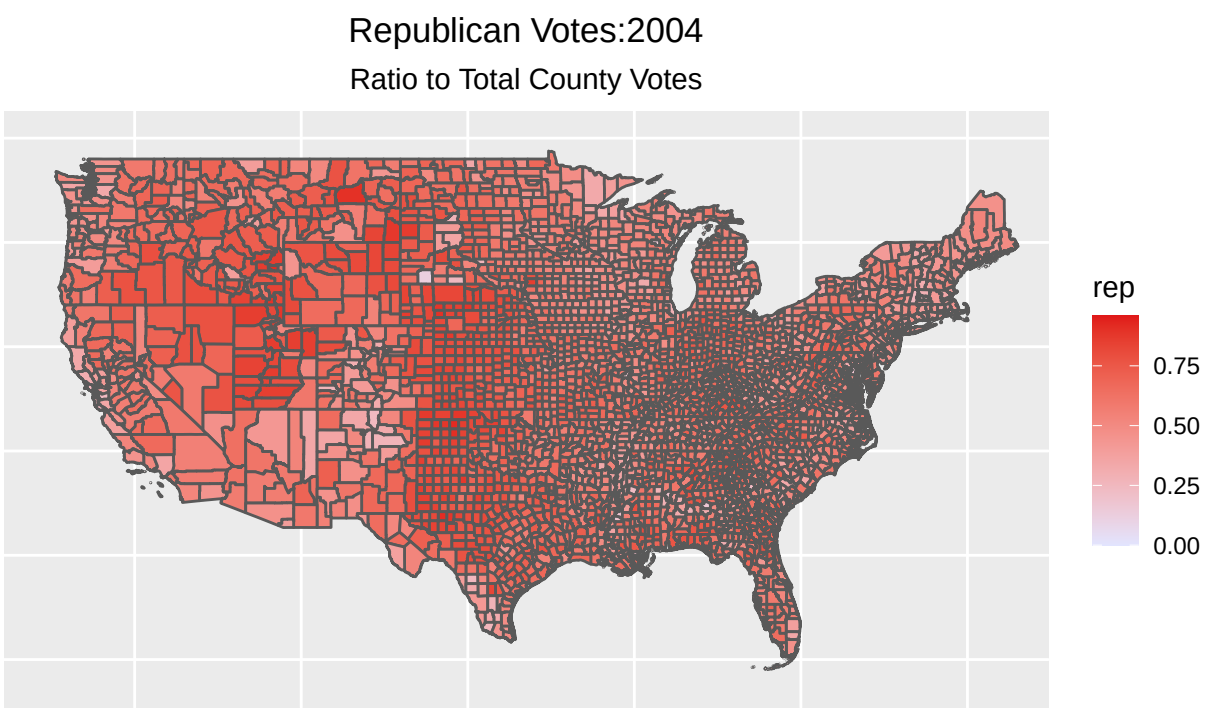


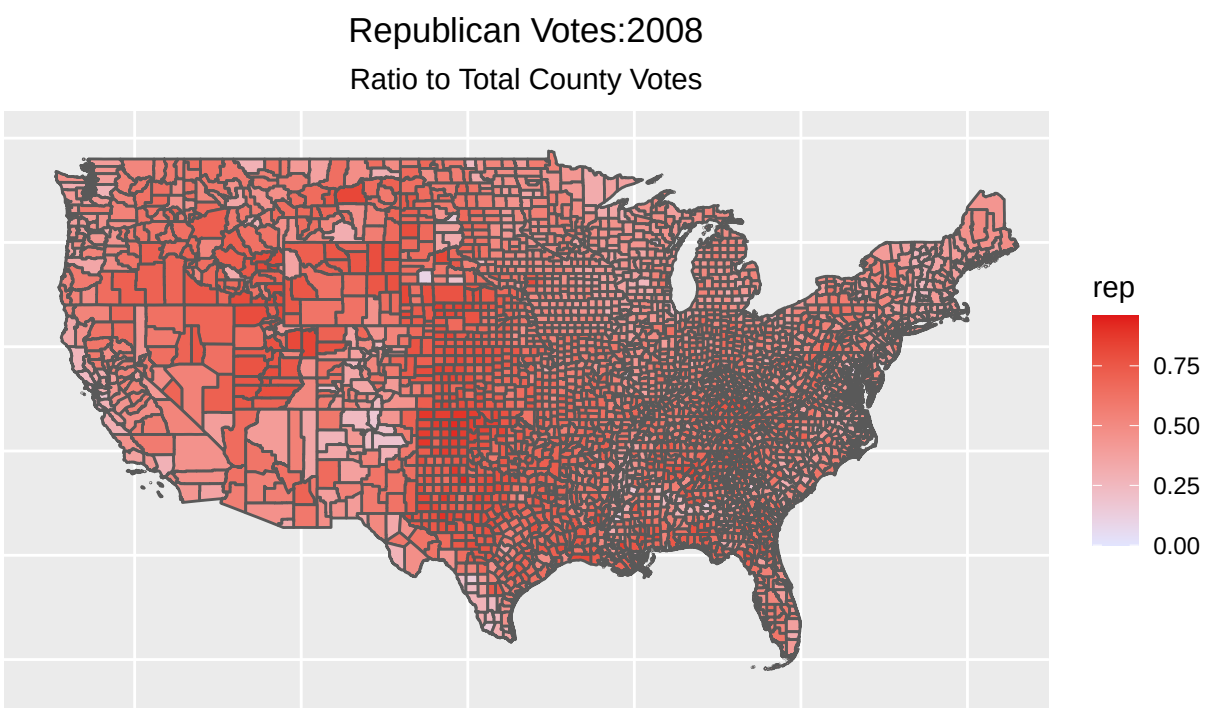
Now we will do the same for republican votes.

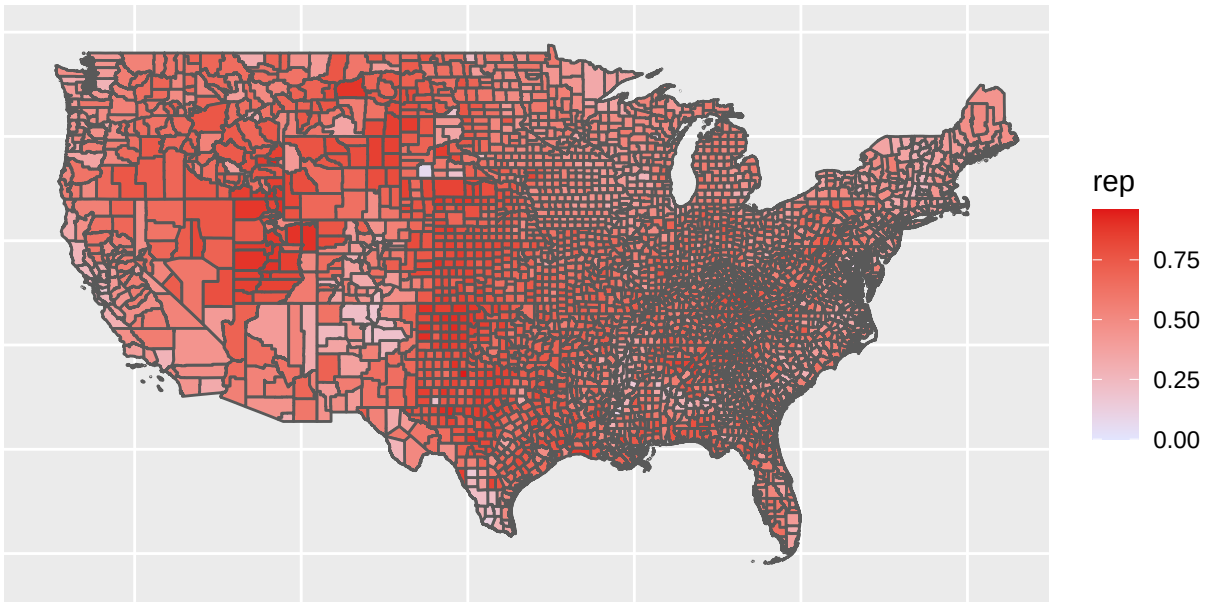
```
rep_animation <- animation_data %>%
  ggplot() +
  geom_sf(aes(fill = rep)) +
  Mainland_US +
  transition_manual(year) +
  labs(
    title = "Republican Votes:{current_frame}",
    subtitle = "Ratio to Total County Votes"
  ) +
  theme(
    plot.title = element_text(hjust = .5),
    plot.subtitle = element_text(hjust = .5),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  ) +
  scale_fill_gradient(
    low = "#E2E5FF",
    high = "#E01818"
  )

rep_animation
```

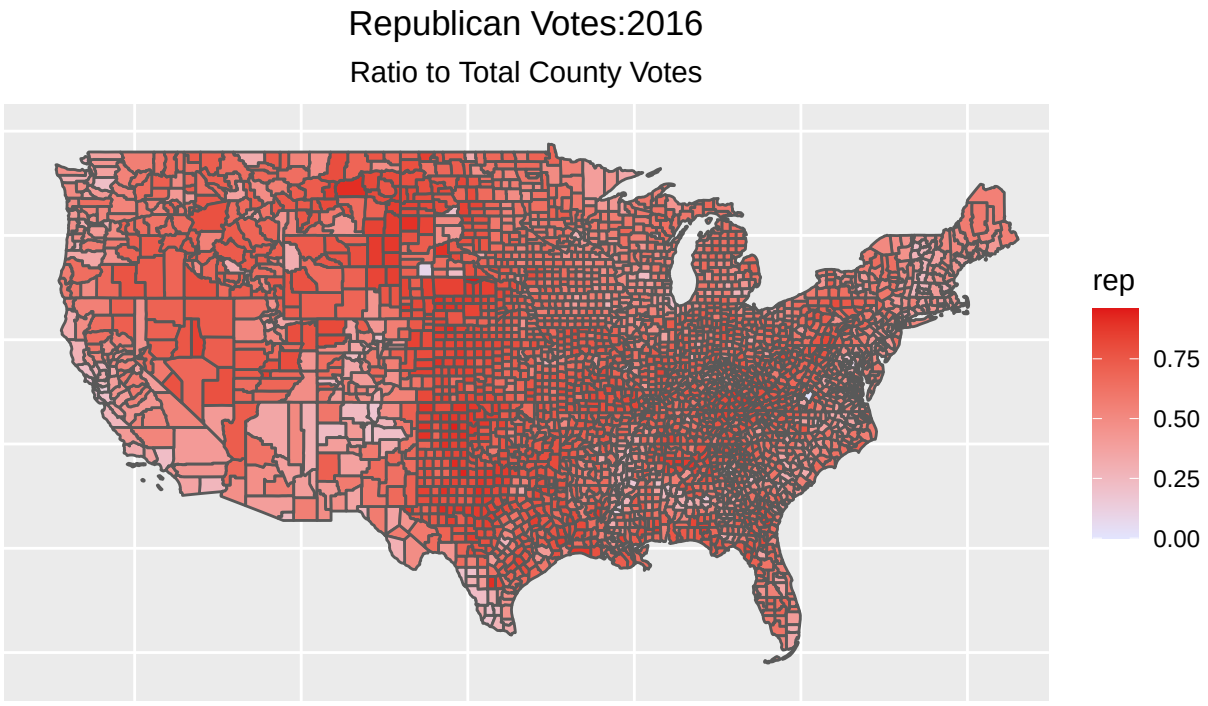
```
## nframes and fps adjusted to match transition
```


Republican Votes:2000
Ratio to Total County Votes
rep
0.75
0.50
0.25
0.00


Republican Votes:2004
Ratio to Total County Votes
rep
0.75
0.50
0.25
0.00


Republican Votes:2008
Ratio to Total County Votes
rep
0.75
0.50
0.25
0.00


Republican Votes:2012
Ratio to Total County Votes
rep
0.75
0.50
0.25
0.00



For now, this concludes the objective I set out for myself. I hope to add on to these graphics in several ways, including:

1. I want to create categorical variables setting up the different ratios into bins, so as to more effectively denote large changes in local voter behavior, instead of the continuous color scheme provided above. I'd also like to create distinct outlines for counties in which a party won.
2. I want to create a single interactive figure, where a viewer can select which party and year to view.
3. I'd like to add in miniature plots of Alaska and Hawaii, so as to not leave them out of the graphics.

If you have advice, feedback, or comments you'd like me to hear, please message me on LinkedIn! Thank you for taking the time to read my work!