Alignment Part 1

Peter Jones

2024-05-29

Loading and cleaning data

In the following code, I load in and merge 27 separate data sets, one for each presidential election from 1920 to 2020. Each dataset includes how many votes each candidate received in a given year, but no additional information.

```
#load data (every election by year)
P_1920 <- read.csv("data/State_Presidential_Election_Data_1920_0_0_1.csv")
P_1924 <- read.csv("data/State_Presidential_Election_Data_1924_0_0_1.csv")
P_1928 <- read.csv("data/State_Presidential_Election_Data_1928_0_0_1.csv")
P_1932 <- read.csv("data/State_Presidential_Election_Data_1932_0_0_1.csv")
P_1936 <- read.csv("data/State_Presidential_Election_Data_1936_0_0_1.csv")
P_1940 <- read.csv("data/State_Presidential_Election_Data_1940_0_0_1.csv")
P_1944 <- read.csv("data/State_Presidential_Election_Data_1944_0_0_1.csv")
P_1948 <- read.csv("data/State_Presidential_Election_Data_1948_0_0_1.csv")
P_1952 <- read.csv("data/State_Presidential_Election_Data_1952_0_0_1.csv")
P_1956 <- read.csv("data/State_Presidential_Election_Data_1956_0_0_1.csv")
P_1960 <- read.csv("data/State_Presidential_Election_Data_1960_0_0_1.csv")
P_1964 <- read.csv("data/State_Presidential_Election_Data_1964_0_0_1.csv")
P_1968 <- read.csv("data/State_Presidential_Election_Data_1968_0_0_1.csv")
P_1972 <- read.csv("data/State_Presidential_Election_Data_1972_0_0_1.csv")
P_1976 <- read.csv("data/State_Presidential_Election_Data_1976_0_0_1.csv")
P_1980 <- read.csv("data/State_Presidential_Election_Data_1980_0_0_1.csv")
P_1984 <- read.csv("data/State_Presidential_Election_Data_1984_0_0_1.csv")
P_1988 <- read.csv("data/State_Presidential_Election_Data_1988_0_0_1.csv")
P_1992 <- read.csv("data/State_Presidential_Election_Data_1992_0_0_1.csv")
P_1996 <- read.csv("data/State_Presidential_Election_Data_1996_0_0_1.csv")
P_2000 <- read.csv("data/State_Presidential_Election_Data_2000_0_0_1.csv")
P_2004 <- read.csv("data/State_Presidential_Election_Data_2004_0_0_1.csv")
P_2008 <- read.csv("data/State_Presidential_Election_Data_2008_0_0_1.csv")
P_2012 <- read.csv("data/State_Presidential_Election_Data_2012_0_0_1.csv")
P_2016 <- read.csv("data/State_Presidential_Election_Data_2016_0_0_1_b.csv")
P_2020 <- read.csv("data/State_Presidential_Election_Data_2020_0_0_1.csv")

#list
dfs <- mget(ls(pattern = "^P_"))

# Add a new column for each data frame indicating the year
dfs <- lapply(names(dfs), function(x) {
  df <- dfs[[x]]
  df$Year <- as.numeric(sub("^P_", "", x))
  return(df)
```

```
})

# Merge all data frames into one
elections <- bind_rows(dfs)
#clean out extra rows
elections <- elections %>%
  mutate_all(as.numeric)
```

```
## Warning: There were 314 warnings in `mutate()`.
## The first warning was:
## i In argument: `FIPS = .Primitive("as.double")(FIPS)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 313 remaining warnings.
```

```
elections <- elections |>
  filter((!is.na(Total.Vote)))
#move Year first
elections <- elections |>
  select(Year, everything())
```

Further shaping the data

At present, the data does not include geographical names. It also includes irrelevant candidates (e.g. Kanye West) that do not really match our purposes. In the following block of code, I match state names into the dataset using FIPS numbers and filter out all candidates that did not receive more than 10% of any state during their candidacy. Note that this 10% benchmark is easily adjustable. I also remove "candidates" that do not stand for a real person, such as "unpledged electors."

```
#FIPS to state DF
fips <- read.csv("data/fips.csv")
fips <- fips |>
  select(FIPS, State)

#merge to label elections df
elections <- left_join(elections, fips, by = "FIPS")
elections <- elections |>
  select(FIPS, State, everything()) |>
  select(!Geographic.Name) |>
  select(!Geographic.Subtype)

# create 10 percent benchmark
elections <- elections |>
  mutate(tenp_benchmark = Total.Vote * 0.1) |>
  select(FIPS, State, Year, Total.Vote, tenp_benchmark, everything())
elections$tenp_benchmark <- elections$tenp_benchmark |>
  round()

# Eliminate columns where no candidate received 10% of the vote
candidate_columns <- names(elections)[!names(elections) %in% c("FIPS", "State", "Year", "Total.Vote", "
keep_columns <- sapply(candidate_columns, function(col) {
  any(elections[[col]] >= elections$tenp_benchmark, na.rm = TRUE)
```

```
})

# Add the non-candidate columns to keep_columns
keep_columns <- c(TRUE, TRUE, TRUE, TRUE, TRUE, keep_columns)

# Filter the elections data frame and get rid of "No.Candidate" and "Unpledged.Elector"
elections <- elections[, keep_columns]
elections <- elections |>
  select(!No.Candidate) |>
  select(!Unpledged.Elector) |>
  select(!Unpledged.Electors)
```

A new organizing variable

Here, I create a variable for "state-year", our unit of analysis. This will be helpful for all subsequent analysis.

```
#make state-year variable for subsequent electoral vote merge
elections <- elections %>%
  mutate(State.Year = str_c(State, Year, sep = "_")) |>
  select(State.Year, everything())
```

Adding in electoral college vote tallies for each state-year

Outside of this program, I built a CSV that displays the number of votes under the Electoral College for each state-year unit. I based it on the table available at https://www.270towin.com/state-electoral-vote-history/. Next, I will load this homemade CSV into this program and attach it to my master elections dataset.

```
ecv <- read.csv("data/EC_Votes.csv")
ecv_select <- ecv |>
  select(State.Year, Votes)
elections <- elections |>
  left_join(ecv_select, by = "State.Year")
elections <- elections %>%
  rename(EC_votes = Votes) %>%
  select(State.Year, FIPS, State, Year, EC_votes, everything())
```

Initial Alignment Computation

RJZ's instructions for the first computation on alignment were as follows: First, I would compute the fraction of the electoral votes a state has in a year. Second, I would determine the percentage of the state's voters who voted for the winner. Third, I would conduct a regression where the percent voting for the winner was a function of the % of electoral votes.

In the following sections, I execute these instructions.

Compute percent of electoral votes for each state-year

```r
elections <- elections |>
  group_by(Year) |>
  mutate(total_EC_votes = sum(EC_votes, na.rm = TRUE))

elections <- elections |>
  mutate(EC_percentage = (EC_votes / total_EC_votes)*100)
```

Compute percentage of each state-years's voters who voted for the winner

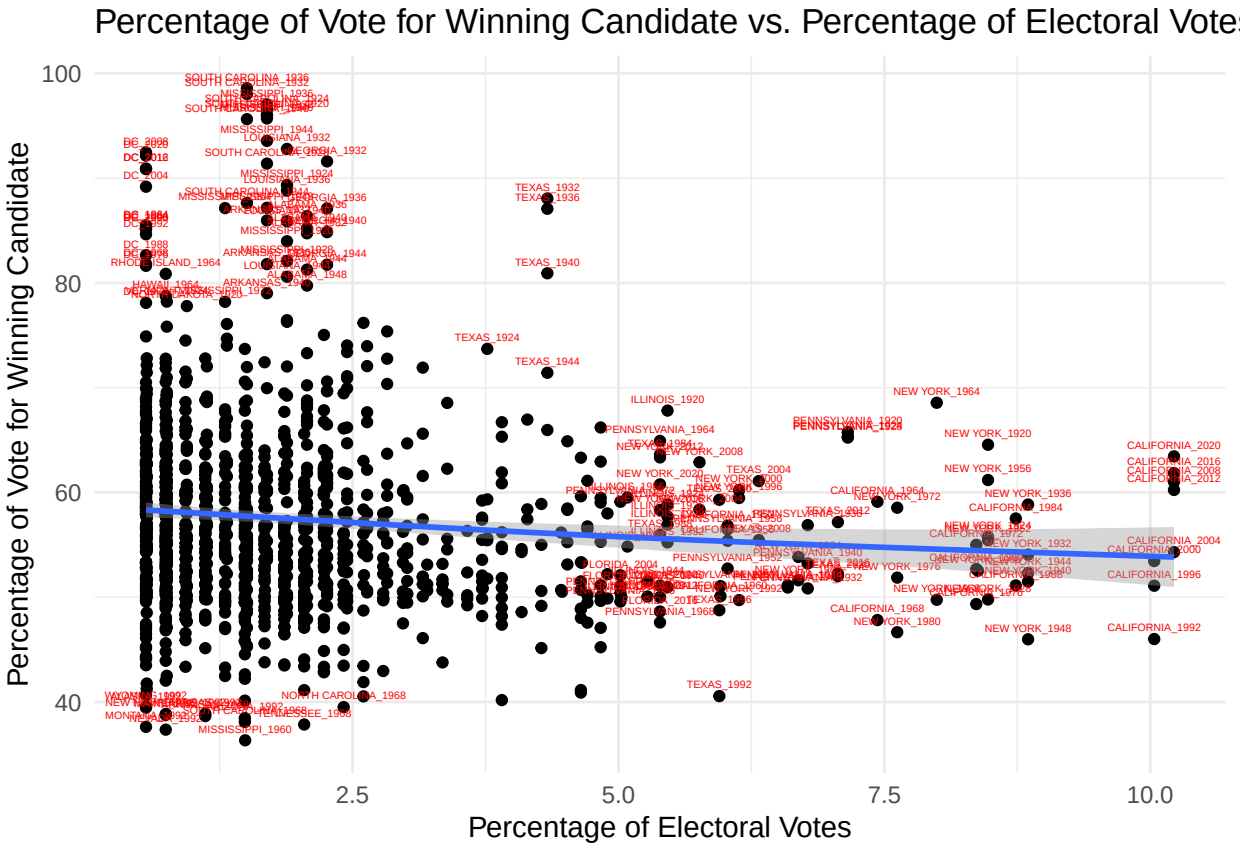
```r
 # Identify the columns with candidate vote counts
candidate_columns <- setdiff(names(elections), c("FIPS", "State", "Year", "Total.Vote", "twentyp_benchma

# Calculate the percentage of the total vote for the winning candidate
elections <- elections %>%
  rowwise() %>%
  mutate(Winner_Vote_Count = max(c_across(all_of(candidate_columns)), na.rm = TRUE),
         Winner_Vote_Percentage = (Winner_Vote_Count / Total.Vote) * 100)
```

Scatterplot and regression for the relationship between these variables

The plot currently only has labels for points outside the "middle" of the diagram, as including labels for all points obscures the line of best fit. However, this can be easily adjusted as well.

```r
#plot
ggplot(data = elections, aes(x = EC_percentage, y = Winner_Vote_Percentage)) +
  geom_point() +
  labs(title = "Percentage of Vote for Winning Candidate vs. Percentage of Electoral Votes",
       x = "Percentage of Electoral Votes",
       y = "Percentage of Vote for Winning Candidate") +
  geom_text(aes(label = ifelse((Winner_Vote_Percentage > 77 | EC_percentage > 5) | Winner_Vote_Percentag
            hjust = 0.5, vjust = -1, size = 1.5, color = "red") +
  theme_minimal() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



The regression below indicates a statistically significant relationship between our variables of interest. Specifically, the slope indicates that for each one percentage point increase in the electoral votes, the percentage of the popular vote for the winning candidate decreases by approximately 0.5603 percentage points. This negative relationship suggests that as the percentage of electoral votes increases, the percentage of the total vote won by the winning candidate tends to slightly decrease. This may indicate that in states with higher electoral vote percentages, the popular vote might be more competitive or closer.

```
# Fit a linear regression model
regression_model <- lm(Winner_Vote_Percentage ~ EC_percentage, data = elections)

# Summarize the regression results
summary(regression_model)
```

```
##
## Call:
## lm(formula = Winner_Vote_Percentage ~ EC_percentage, data = elections)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.428  -5.869  -1.396   3.990  40.814
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    58.6047     0.3971 147.597  < 2e-16 ***
## EC_percentage  -0.5603     0.1512  -3.707 0.000219 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.213 on 1293 degrees of freedom
## Multiple R-squared:  0.01051,    Adjusted R-squared:  0.009748
## F-statistic: 13.74 on 1 and 1293 DF,  p-value: 0.000219
```