**DATA**

Link to the original data in .csv format: <https://github.com/fivethirtyeight/data/blob/76c471a9124d690ba92709ca21cbfcdde226b44e/polls/pres_pollaverages_1968-2016.csv>

**R CODE**

#imports

library(lubridate)

library(tidyverse)

library(repr)

library(infer)

library(cowplot)

library(broom)

library(dplyr)

library(data.table)

library(ggplot2)

library(moments)

options(repr.plot.width = 10, repr.plot.height = 10)

# LOADING DATA AND PERFORMING QUALITY CONTROL

data\_raw <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/76c471a9124d690ba92709ca21cbfcdde226b44e/polls/pres\_pollaverages\_1968-2016.csv")

data\_distribution <- data\_raw |>

ggplot(aes(x = pct\_estimate)) +

geom\_density() +

ylab("Density") +

xlab("Values of pct\_estimate") +

theme(

text = element\_text(size = 19),

plot.title = element\_text(face = "bold"),

axis.title = element\_text(face = "bold")) +

scale\_x\_continuous(breaks = seq(0, 100, 10))

ggsave("data\_distribution.png", plot = data\_distribution, width = 10, height = 10)

distribution\_summary <- summary(data\_raw$pct\_estimate)

# TIDYING DATA

# only keep relevant columns

data\_tidy <- select(data\_raw, cycle:pct\_estimate,election\_date)

# rename modeldate column

data\_tidy <- rename(data\_tidy, model\_date = modeldate)

data\_tidy <- rename(data\_tidy, election\_year = cycle)

# change model\_date and election\_date from char to date format for easier manipulation

data\_tidy$election\_date <- mdy(data\_tidy$election\_date)

data\_tidy$model\_date <- mdy(data\_tidy$model\_date)

# FILTERING DATA BY DATE

# Extract month and day from date column

data\_tidy$month <- month(data\_tidy$model\_date)

data\_tidy$day <- day(data\_tidy$model\_date)

# Filter out rows where the date is before April 9th

filtered\_data <- data\_tidy |>

filter(month > 3 & day > 8)

# show the earliest date with data recorded for each election to ensure the right subset of data was taken

first\_dates\_by\_election\_year <- filtered\_data |>

group\_by(election\_year) |>

slice\_head(n = 1) |>

select(election\_year, model\_date)

# creating a .csv file for the filtered\_data

write.csv(filtered\_data, "../data/filtered\_data.csv", row.names = FALSE)

# GROUPING THE DATA INTO DIVISIONS AND REMOVING UNNEEDED COLUMNS

# Grouping states by division

grouped\_by\_division <- filtered\_data |>

filter(!(state %in% c("ME-1", "ME-2", "NE-1", "NE-2", "NE-3", "National"))) |>

mutate(division = case\_when(

state %in% c("Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island", "Vermont") ~ "New England",

state %in% c("New Jersey", "New York", "Pennsylvania") ~ "Middle Atlantic",

state %in% c("Illinois", "Indiana", "Michigan", "Ohio", "Wisconsin") ~ "East North Central",

state %in% c("Iowa", "Kansas", "Minnesota", "Missouri", "Nebraska", "North Dakota", "South Dakota") ~ "West North Central",

state %in% c("Delaware", "District of Columbia", "Florida", "Georgia", "Maryland", "North Carolina", "South Carolina", "Virginia", "West Virginia") ~ "South Atlantic",

state %in% c("Alabama", "Kentucky", "Mississippi", "Tennessee") ~ "East South Central",

state %in% c("Arkansas", "Louisiana", "Oklahoma", "Texas") ~ "West South Central",

state %in% c("Arizona", "Colorado", "Idaho", "Montana", "Nevada", "New Mexico", "Utah", "Wyoming") ~ "Mountain",

state %in% c("Alaska", "California", "Hawaii", "Oregon", "Washington") ~ "Pacific"

))

# remove any unneeded columns

grouped\_by\_division <- grouped\_by\_division |>

select(election\_year, candidate\_name, pct\_estimate, division,model\_date)

# CALCULATING PER DISTRICT POLLING AVERAGE FOR EACH CANDIDATE IN EACH ELECTION

# Define the function to calculate averages by year (unchanged)

calculate\_average\_by\_year <- function(dataframe, year) {

filtered\_data <- filter(dataframe, election\_year == year)

average\_pct\_estimates <- aggregate(

pct\_estimate ~ candidate\_name + division,

data = filtered\_data,

FUN = mean

)

average\_pct\_estimates$election\_year <- year # Add year information

return(average\_pct\_estimates)}

# Create an empty dataframe to store combined results

all\_year\_averages <- data.frame() # Start with an empty dataframe

# Iterate through years and append results to the dataframe

years <- c(2016, 2012, 2008, 2004, 2000, 1996, 1992, 1988, 1984, 1980, 1976, 1972)

for (year in years) {

year\_df <- calculate\_average\_by\_year(grouped\_by\_division, year)

all\_year\_averages <- rbind(all\_year\_averages, year\_df)}

# creating a .csv file for the computed division averages

write.csv(all\_year\_averages, "../data/division\_averages.csv", row.names = FALSE)

division\_averages <- read.csv("../data/division\_averages.csv")

# CALCULATING A WEIGHTED AVERAGE OF EACH CANDIDATE'S POLLING SCORES BASED ON DIVISION

# Each district's electoral vote total is the sum of all their states' electoral votes

south\_atlantic <- 104

pacific <- 81

east\_north\_central <- 72

mid\_atlantic <- 61

west\_south\_central <- 61

mountain <- 49

west\_north\_central <- 43

east\_south\_central <- 34

new\_england <- 33

total <- 538

# Define weights dictionary

weights <- c(

"South Atlantic" = south\_atlantic / total,

"Pacific" = pacific / total,

"East North Central" = east\_north\_central / total,

"Middle Atlantic" = mid\_atlantic / total,

"West South Central" = west\_south\_central / total,

"Mountain" = mountain / total,

"West North Central" = west\_north\_central / total,

"East South Central" = east\_south\_central / total,

"New England" = new\_england / total)

# Calculate weighted average per candidate and year

weighted\_averages <- division\_averages %>%

group\_by(candidate\_name, election\_year) %>%

summarize(weighted\_pct\_estimate = sum(pct\_estimate \* weights[division]) )

weighted\_averages$weighted\_pct\_estimate <- round(weighted\_averages$weighted\_pct\_estimate, digits = 2)

weighted\_averages <- weighted\_averages |> arrange(election\_year)

# creating a .csv file for the weighted averages

write.csv(weighted\_averages, "../data/weighted\_averages.csv", row.names = FALSE)

# Calculate summary statistics for comparison

means <- mean(weighted\_averages$weighted\_pct\_estimate)

median <- median(weighted\_averages$weighted\_pct\_estimate)

std <- sd(weighted\_averages$weighted\_pct\_estimate)

variance <- var(weighted\_averages$weighted\_pct\_estimate)

kurtosis <- kurtosis(weighted\_averages$weighted\_pct\_estimate)

data <- list(means, median, std, variance,kurtosis)

headers <- c("Mean", "Median", "Standard Deviation", "Variance", "Kurtosis")

summary\_stats <- data.frame(data, stringsAsFactors = FALSE)

colnames(summary\_stats) <- headers

# creating a .csv file for the summary statistics

write.csv(summary\_stats, "../data/summary\_stats.csv", row.names = FALSE)