OQMSBS - Lab Assignment 1

Michael Sullivan

2025-01-23

```
#Reading in and checking the coffee data
survey_df <- read.csv("GACTT_RESULTS_ANONYMIZED_HW1.csv")

print(head(survey_df,6))
```

```
##   submission_id  zip             age gender cups cups_num
## 1        gMR29l <NA> 18-24 years old   <NA> <NA>       NA
## 2        BkPN0e <NA> 25-34 years old   <NA> <NA>       NA
## 3        W5G8jj <NA> 25-34 years old   <NA> <NA>       NA
## 4        4xWgGr <NA> 35-44 years old   <NA> <NA>       NA
## 5        QD27Q8 <NA> 25-34 years old   <NA> <NA>       NA
## 6        V0LPeM <NA> 55-64 years old   <NA> <NA>       NA
##                                                          home_brew party
## 1                                                              <NA>  <NA>
## 2                         Pod/capsule machine (e.g. Keurig/Nespresso)  <NA>
## 3                                              Bean-to-cup machine  <NA>
## 4                            Coffee brewing machine (e.g. Mr. Coffee)  <NA>
## 5                                                        Pour over  <NA>
## 6 Pod/capsule machine (e.g. Keurig/Nespresso), Espresso, French press  <NA>
```

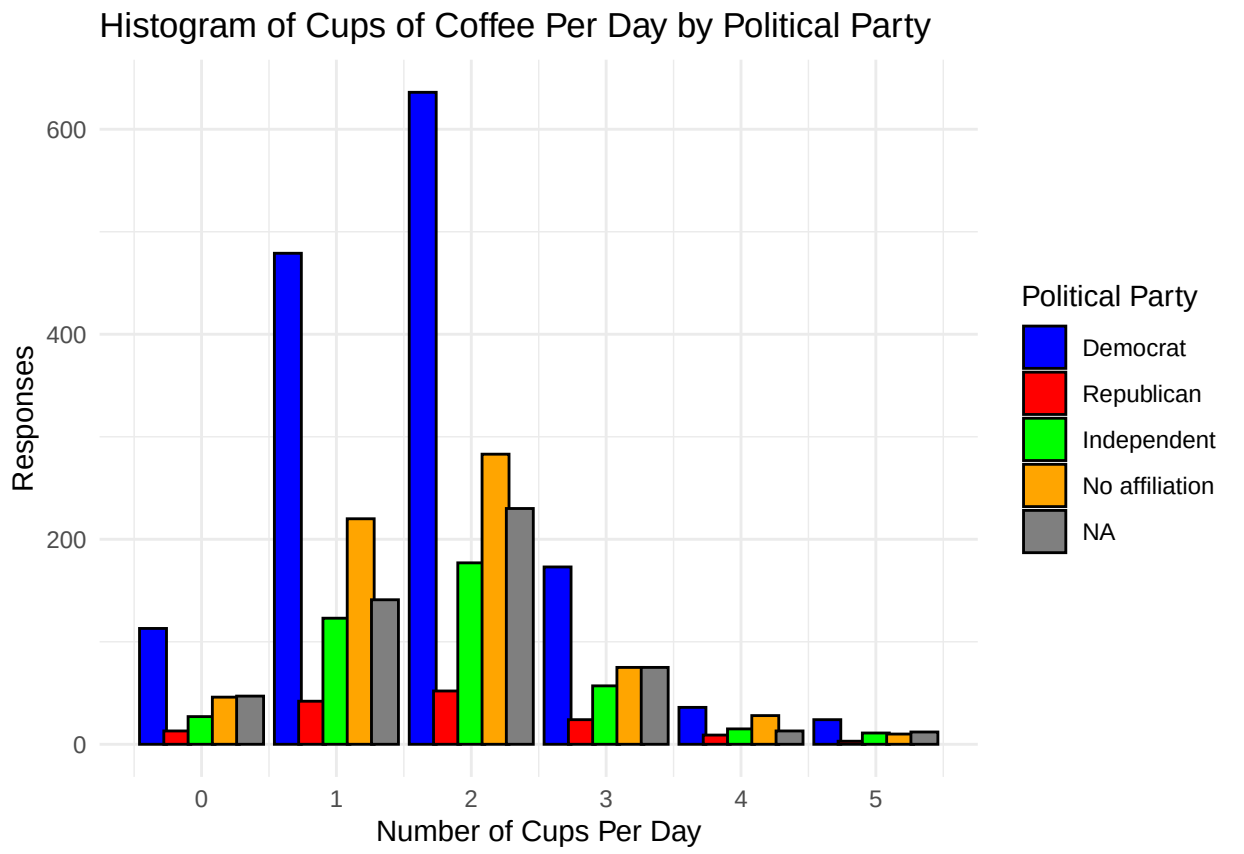
```
print(tail(survey_df,4))
```

```
##      submission_id   zip             age gender        cups cups_num
## 3277        42EpEY 91505 25-34 years old   <NA> More than 4        5
## 3278        g5ggRM 60131 18-24 years old   Male           1        1
## 3279        rlgbDN  2351 25-34 years old   Male           2        2
## 3280        0EGYe9 32765 25-34 years old Female           1        1
##                                                                    home_brew
## 3277           Espresso, Bean-to-cup machine, Cold brew, French press, Pour over
## 3278 Espresso, Pod/capsule machine (e.g. Keurig/Nespresso), Instant coffee, Other
## 3279                                                                  Pour over
## 3280                                  Pour over, French press, Espresso, Other
##         party
## 3277     <NA>
## 3278 Democrat
## 3279 Democrat
## 3280 Democrat
```

```
print(sapply(survey_df,class))
```

```
## submission_id           zip           age        gender          cups
##   "character"   "character"   "character"   "character"   "character"
##      cups_num     home_brew         party
##     "integer"   "character"   "character"
```

```r
survey_df$party <- factor(survey_df$party, levels = c("Democrat", "Republican", "Independent", "No affi

#Plotting a histogram of cups/day by political party
ggplot(survey_df, aes(x = cups_num, fill = party)) +
  geom_histogram(position = position_dodge(width = 0.9, preserve = "single"),
                 binwidth = 1,
                 color = "black") +
  scale_x_continuous(
    breaks = seq(0, 5, by = 1),
    labels = seq(0, 5, by = 1)
  ) +
  scale_fill_manual(
    values = c("Democrat" = "blue",
               "Republican" = "red",
               "Independent" = "green",
               "No affiliation" = "orange")
  ) +
  labs(title = "Histogram of Cups of Coffee Per Day by Political Party",
       x = "Number of Cups Per Day",
       y = "Responses",
       fill = "Political Party") +
  theme_minimal()
```



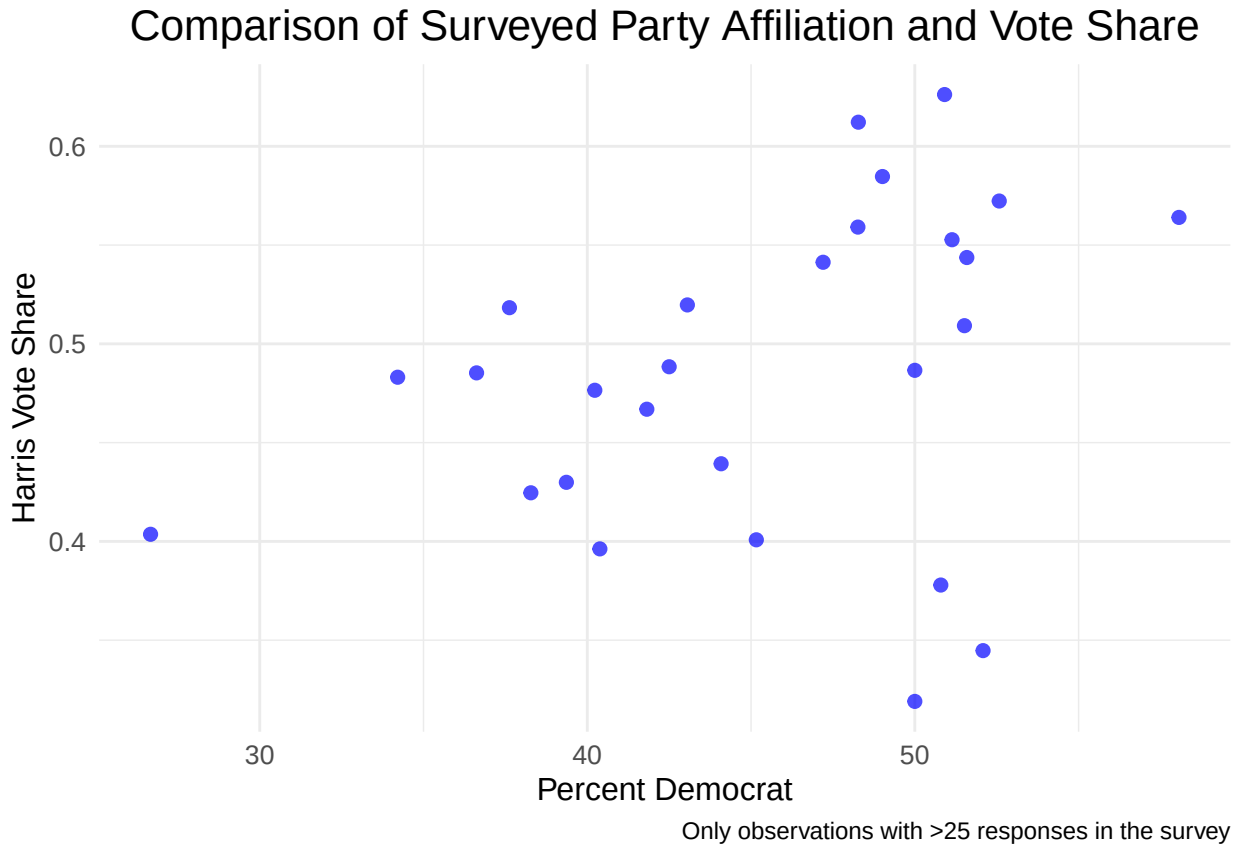

```
#Reading in, cleaning, and merging geographic data
zip_df <- read.csv("zip_code_database.csv")
survey_df$zip <- as.integer(survey_df$zip)
joined_df <- left_join(survey_df,zip_df, by = "zip")

print(sum(!(joined_df$zip %in% zip_df$zip)))
```

```
## [1] 117
```

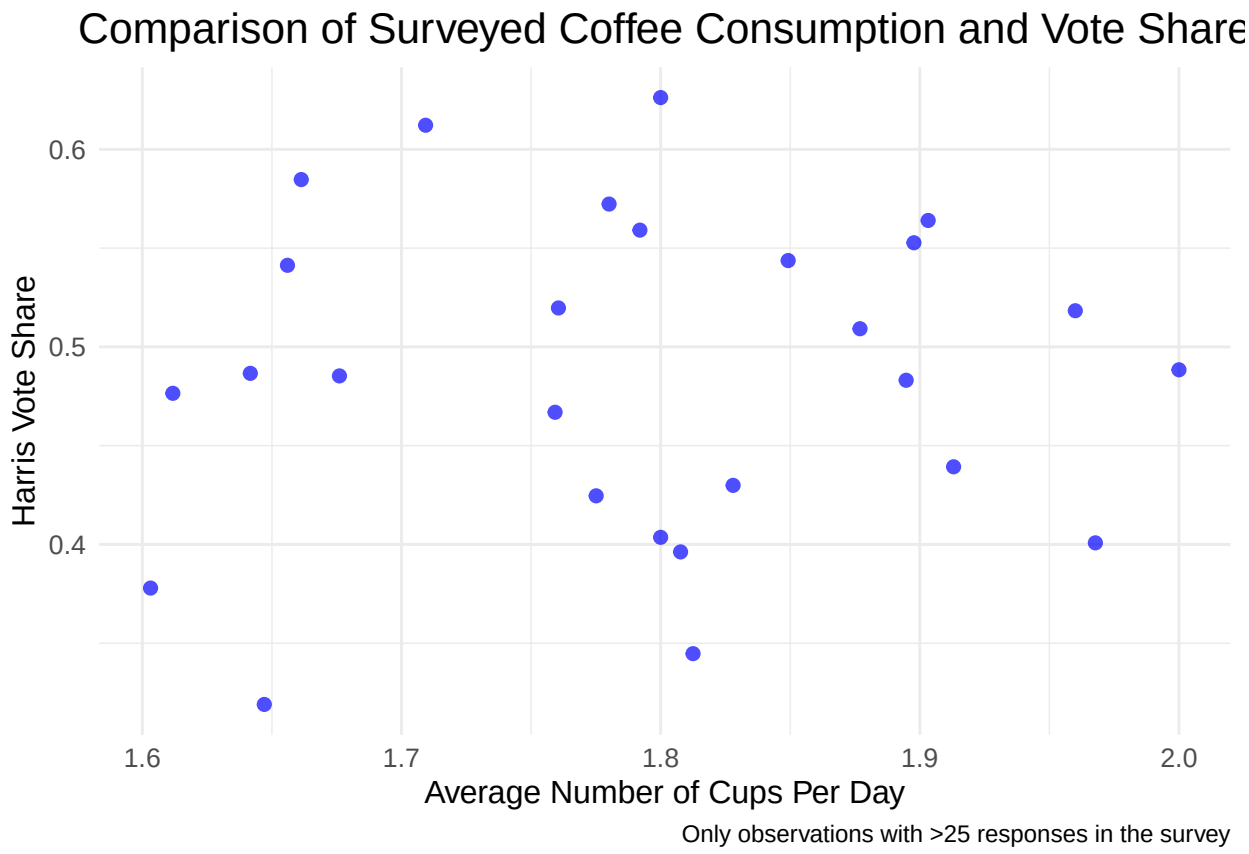
```
#Function to calculate the mode of a column
calculate_mode <- function(x) {
  unique_x <- unique(x)
  counts <- tabulate(match(x, unique_x))
  modes <- unique_x[counts == max(counts)]
  paste(modes, collapse = ", ")
}

#Creating a new data frame with results by state
survey_state <- group_by(joined_df, joined_df$state) %>% summarize(avg_cups = mean(cups_num, na.rm=TRUE)
                                                                   preferred_method = calculate_mode(home_
                                                                   pct_dem = 100*sum(party == "Democrat",
                                                                   pct_rep = 100*sum(party == "Republican
                                                                   num_responses = n())
colnames(survey_state)[colnames(survey_state) == "joined_df$state"] <- "state_abr"
```

```r
#Reading in and cleaning election data
election_df <- read.csv("election_2024.csv")

int_columns <- c(2,3,5,6,8,9,11)
pct_columns <- c(4,7,10)

election_df[,int_columns] <- sapply(election_df[,int_columns], function(x) as.integer(gsub(",","",x)))
election_df[,pct_columns] <- sapply(election_df[,pct_columns], function(x) (1/100)*as.numeric(gsub("%",
election_df[is.na(election_df)] <- 0

#Cleaning up the states and giving them the same labels
election_df <- election_df[!election_df$state %in% c("CD-1","CD-2","CD-3"),]

state_names <- c(
  "Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
  "Connecticut", "Delaware", "Florida", "Georgia", "Hawaii", "Idaho",
  "Illinois", "Indiana", "Iowa", "Kansas", "Kentucky", "Louisiana",
  "Maine", "Maryland", "Massachusetts", "Michigan", "Minnesota",
  "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada",
  "New Hampshire", "New Jersey", "New Mexico", "New York", "North Carolina",
  "North Dakota", "Ohio", "Oklahoma", "Oregon", "Pennsylvania",
  "Rhode Island", "South Carolina", "South Dakota", "Tennessee",
  "Texas", "Utah", "Vermont", "Virginia", "Washington",
  "West Virginia", "Wisconsin", "Wyoming", "District of Columbia"
)

state_abbreviations <- c(
  "AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA", "HI", "ID",
  "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD", "MA", "MI", "MN",
  "MS", "MO", "MT", "NE", "NV", "NH", "NJ", "NM", "NY", "NC", "ND",
  "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN", "TX", "UT", "VT",
  "VA", "WA", "WV", "WI", "WY", "DC"
)

state_abr_df <- data.frame(state = state_names, abr = state_abbreviations)

election_df$state_abr <- sapply(election_df$state, function(x) state_abr_df$abr[state_abr_df$state == x]

survey_election_df <- left_join(election_df, survey_state, by = "state_abr")

#Plotting vote share against survey results for party affiliation and for coffee consumption
ggplot(data = filter(survey_election_df, num_responses > 25), aes(x = pct_dem, y = harris_votes_share))
  geom_point(color = "blue", size = 2, alpha = 0.7) +
  labs(
    title = "Comparison of Surveyed Party Affiliation and Vote Share",
    x = "Percent Democrat",
    y = "Harris Vote Share",
    caption = "Only observations with >25 responses in the survey"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12),
```

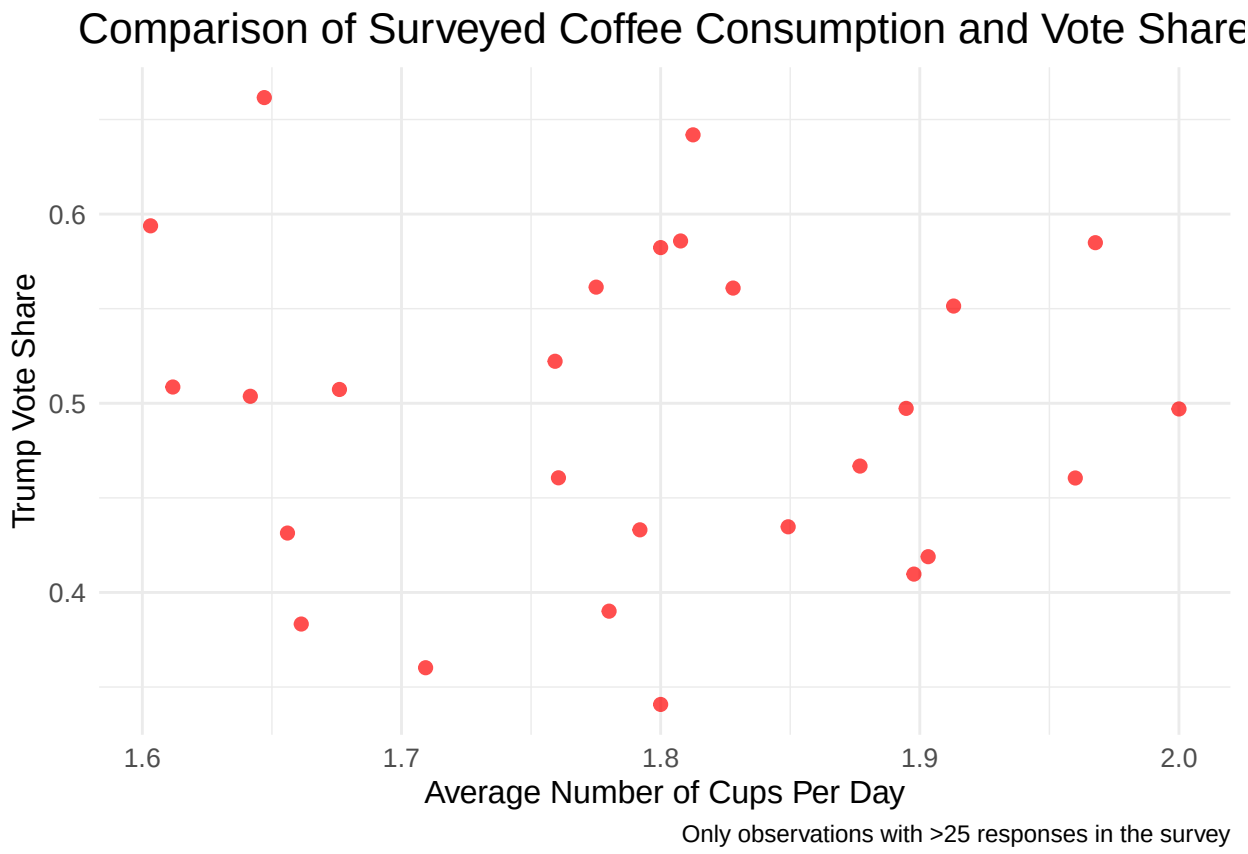
```
    axis.text = element_text(size = 10)
  )
```



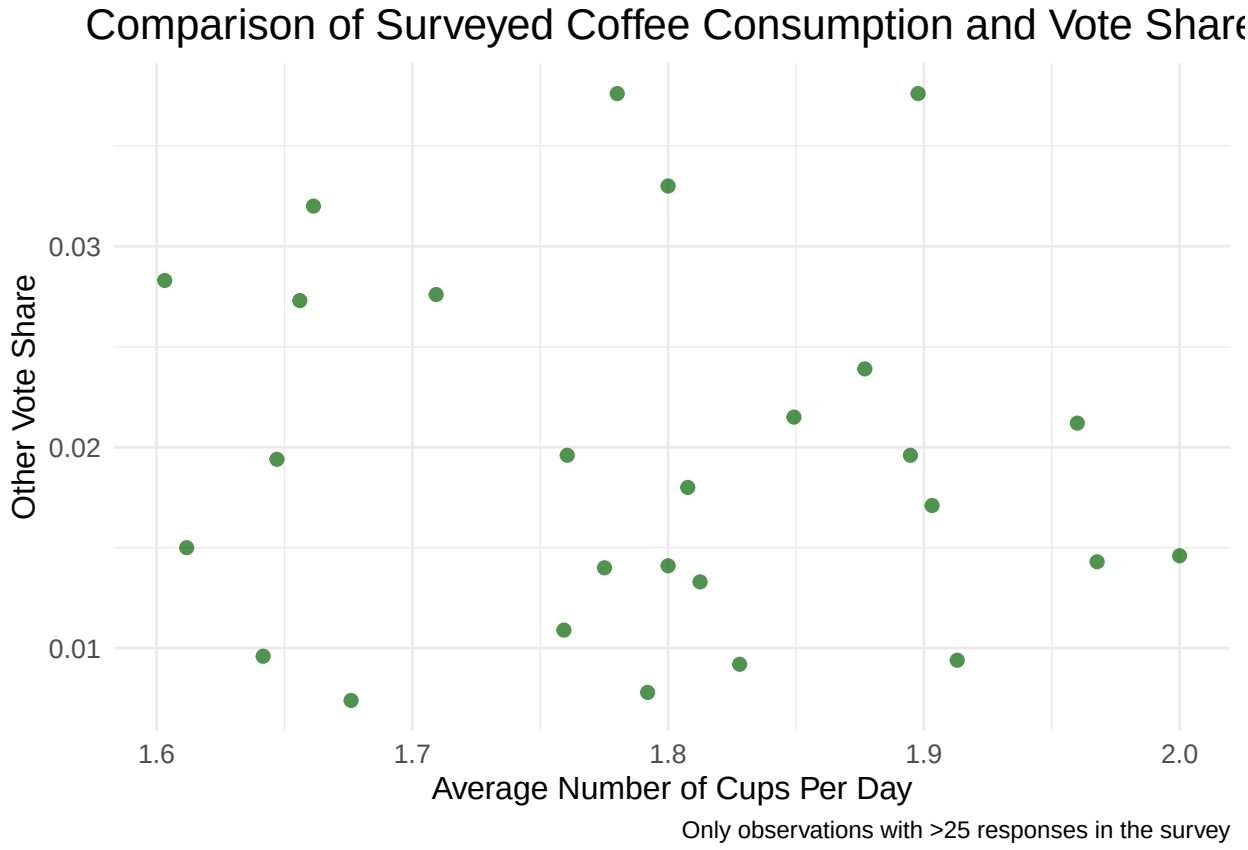

```
ggplot(data = filter(survey_election_df, num_responses > 25), aes(x = avg_cups, y = harris_votes_share))
  geom_point(color = "blue", size = 2, alpha = 0.7) +
  labs(
    title = "Comparison of Surveyed Coffee Consumption and Vote Share",
    x = "Average Number of Cups Per Day",
    y = "Harris Vote Share",
    caption = "Only observations with >25 responses in the survey"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)
  )
```




```
ggplot(data = filter(survey_election_df, num_responses > 25), aes(x = avg_cups, y = trump_votes_share))
  geom_point(color = "red", size = 2, alpha = 0.7) +
  labs(
    title = "Comparison of Surveyed Coffee Consumption and Vote Share",
    x = "Average Number of Cups Per Day",
    y = "Trump Vote Share",
    caption = "Only observations with >25 responses in the survey"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)
  )
```




```
ggplot(data = filter(survey_election_df, num_responses > 25), aes(x = avg_cups, y = other_votes_share))
  geom_point(color = "darkgreen", size = 2, alpha = 0.7) +
  labs(
    title = "Comparison of Surveyed Coffee Consumption and Vote Share",
    x = "Average Number of Cups Per Day",
    y = "Other Vote Share",
    caption = "Only observations with >25 responses in the survey"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)
  )
```




```
write.xlsx(survey_election_df, file = "overview_hw1.xlsx")
```

Visual examination of the scatterplots showing the relationship between surveyed coffee consumption and vote share indicates that there is no meaningful relationship between these two variables. For every category of vote share it is the case that there is no observable trend related to the average number of cups per day.

```
#Read in from excel and set up the data from the Google Trends API pull (done in Python)
google_trend_data <- read_excel("coffee_output_2.xlsx", sheet = 1)
google_trend_data$favorite <- apply(google_trend_data, 1, function(row){
    colnames(google_trend_data)[which.max(row)]
})
google_trend_data$state_abr <- sapply(google_trend_data$geoName, function(x) state_abr_df$abr[state_abr

#Determine how many of the google trends top searches match the survey favorites
survey_google_matches <- mapply(grepl, google_trend_data$favorite, survey_election_df$preferred_method)
num_matches <- sum(survey_google_matches)
print(head(google_trend_data))
```

```
## # A tibble: 6 x 12
##   geoName    `bean-to-cup machine` `coffee brewer` `coffee extract` `cold brew`
##   <chr>                      <dbl>           <dbl>            <dbl>       <dbl>
## 1 Alabama                        0               0                1          18
## 2 Alaska                         0               0                0          13
## 3 Arizona                        0               1                0          18
## 4 Arkansas                       0               1                0          18
```

```
## 5 California                     0              0               0          19
## 6 Colorado                       0              0               0          16
## # i 7 more variables: espresso <dbl>, 'french press' <dbl>,
## #   'instant coffee' <dbl>, keurig <dbl>, 'pour over' <dbl>, favorite <chr>,
## #   state_abr <chr>
```

```
print("The number of matches by state between the top survey result and most-searched term:")
```

```
## [1] "The number of matches by state between the top survey result and most-searched term:"
```

```
print(num_matches)
```

```
## [1] 4
```

The Google trend results were extremely consistent from state to state, with "espresso" being the most searched term of the surveyed options in every state. This is likely a function of the particular dynamics of web searching favoring shorter and single-word searches, as well as searches of more general concepts like espresso than more specific multi-word methods. To further explore this we might consider grouping together multiple terms that refer to the same or similar methods, or we might look at more than just the highest rated term. I experienced semi-frequent issues with access to the google API that slowed down my ability to pull the relevant data, so I have left this with the results of the exact question asked in the assignment

M-P-Sullivan / Google Trend.py Secret

Created now

Code Revisions 1

Quant Methods Lab 1 Bonus Problem

Google Trend.py

```python
import time
import pandas as pd
from pytrends.request import TrendReq
from pytrends.exceptions import TooManyRequestsError

pytrend = TrendReq()
print(1)

# Define Keywords (must be in chunks of 5 or fewer per Google API, had issues with larger)
keywords_chunk1 = ['bean-to-cup machine', 'coffee brewer', 'coffee extract', 'cold brew', 'espress
keywords_chunk2 = ['french press', 'instant coffee', 'keurig', 'pour over']
print(2)

# Function to fetch interest by region with retry and delay
def get_interest_by_region(keywords):
    while True:
        try:
            pytrend.build_payload(
                kw_list=keywords,
                timeframe='today 12-m',  # Last 12 months
                geo='US',
                gprop=''
            )
            time.sleep(10)  # Add a delay of 10 seconds between requests to deal with 429 errors f
            return pytrend.interest_by_region(resolution='REGION')
        except TooManyRequestsError:
            print("Too many requests. Retrying in 30 seconds...")
            time.sleep(30)  # Wait 30 seconds before retrying to further deal with 429s
        except Exception as e:
            print(f"An unexpected error occurred: {e}")
            break

print(3)

# Fetch data for each chunk
interest_by_region1 = get_interest_by_region(keywords_chunk1)
print(4)
```

```
interest_by_region2 = get_interest_by_region(keywords_chunk2)

print(5)
# Combine results
interest_by_region_combined = pd.concat([interest_by_region1, interest_by_region2], axis=1)

print("\nInterest By Region (Combined):")
print(interest_by_region_combined.head(10))
```