

# OQMSBS - Lab Assignment 1

Michael Sullivan

2025-01-25

## Part I

### Question 1.1 a) and 1.1 b)

Below are the prints with the head and tail of the dataset as well as the class of each variable

```
#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      VOLPeM <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      OEGYe9 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))
```

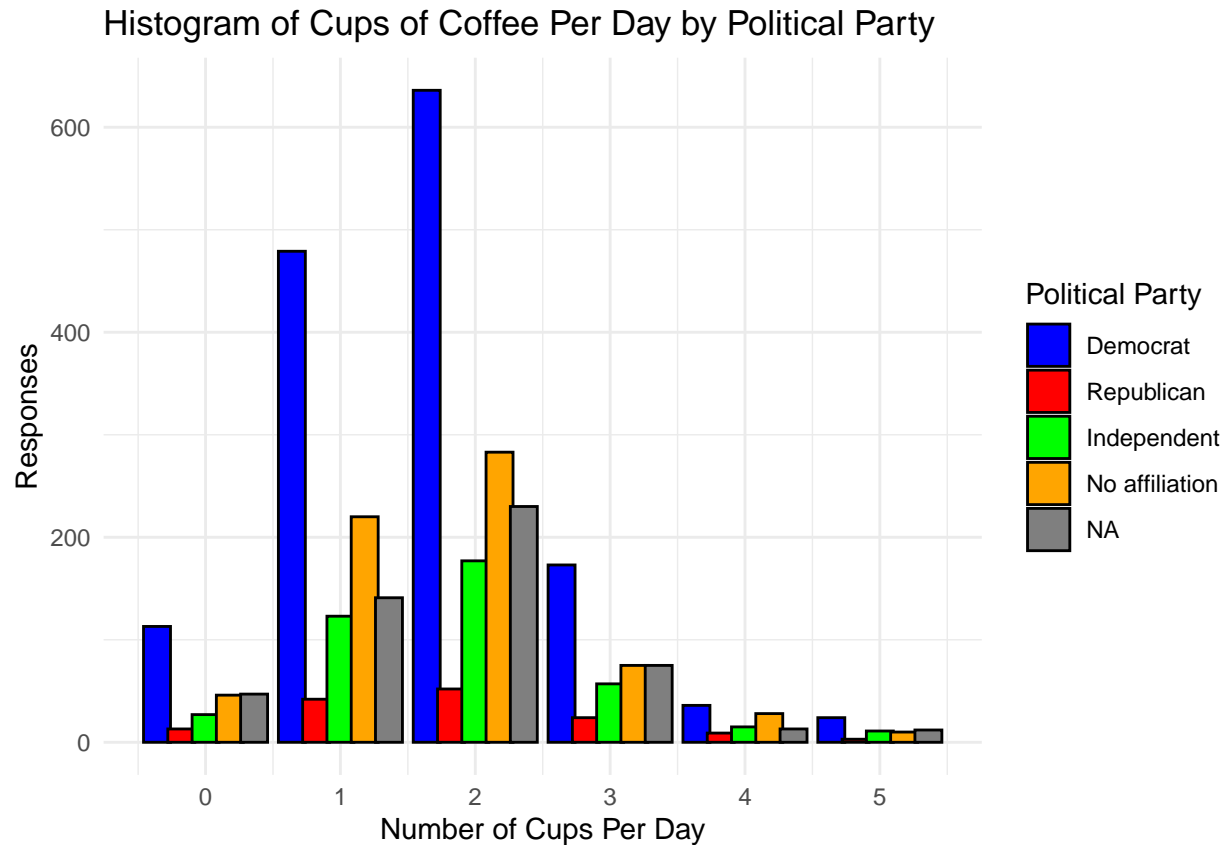
```
## submission_id      zip      age      gender      cups
## "character" "character" "character" "character" "character"
## cups_num      home_brew      party
## "integer" "character" "character"
```

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

### Question 1.1 c)

Below is a histogram showing the relationship between coffee consumption and party affiliation

```
#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()
```



#### Question 2.1 and 2.2 a)

Below is the code to load the zip code metadata and merge the survey results

```
#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")
```

#### Question 2.2 b) and 2.2 c)

Below is the print giving the number of unmatched zip codes (which is 117)

```
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 = ", ")
}
```

## Question 3.1

The below section creates the `survey_state` data frame with the average coffee consumption (`avg_cups`), preferred homebrew method (`preferred_method`) and political affiliations (`pct_dem`, `pct_rep`) for each state.

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

## Part II

### Question 1.1 and 1.2

The below section reads in the election results data and ensures that the numeric columns are properly formatted.

```
#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("%", "", x)))
election_df[is.na(election_df)] <- 0
```

### Question 2.1

The below section merges the survey data with the election data by state (it primarily is dealing with the fact that in one data set we have state names, in the other we have abbreviations)

```
#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")

```

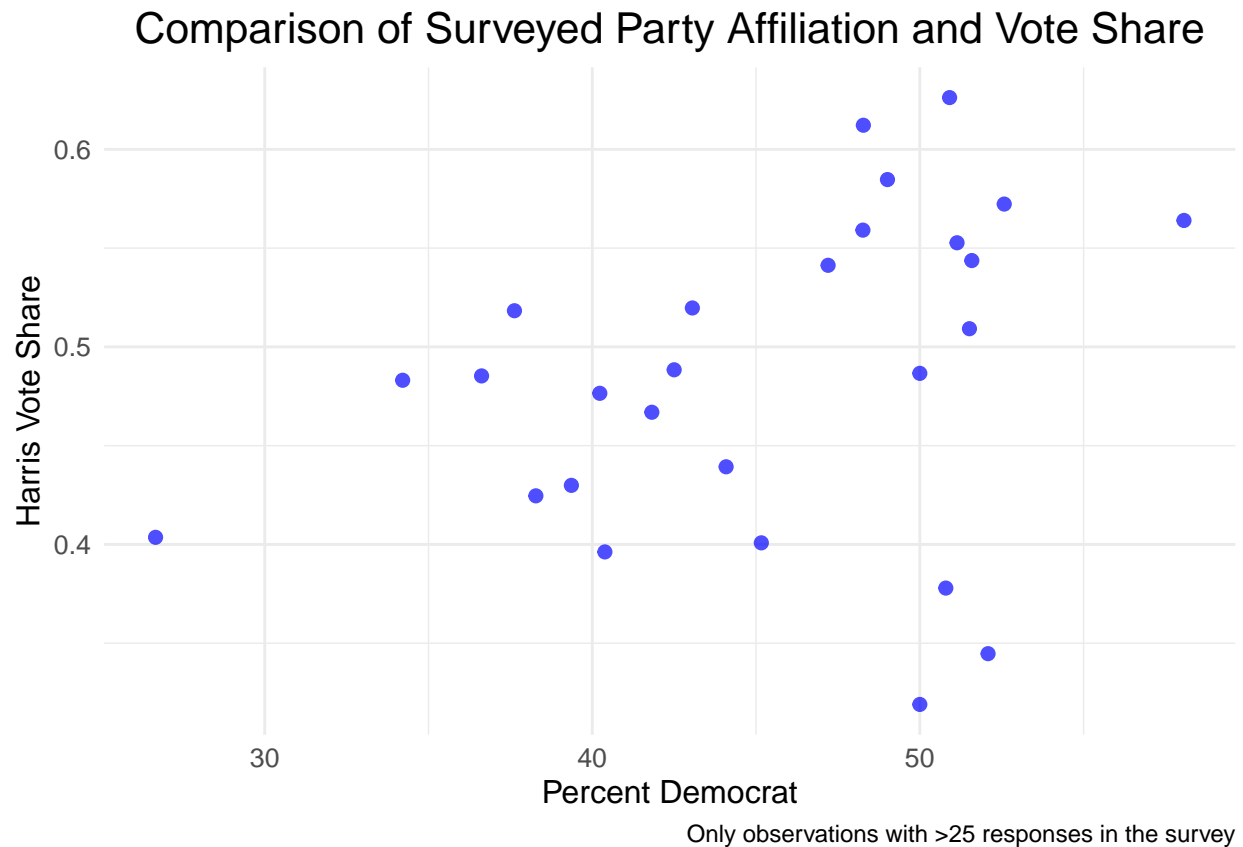
### Question 3.1

Below we plot the relationship between share of respondents supporting democrats and Harris vote share. We exclude states with 25 or fewer responses in the survey to handle the fact that their sample sizes are small.

```

#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)
  )

```

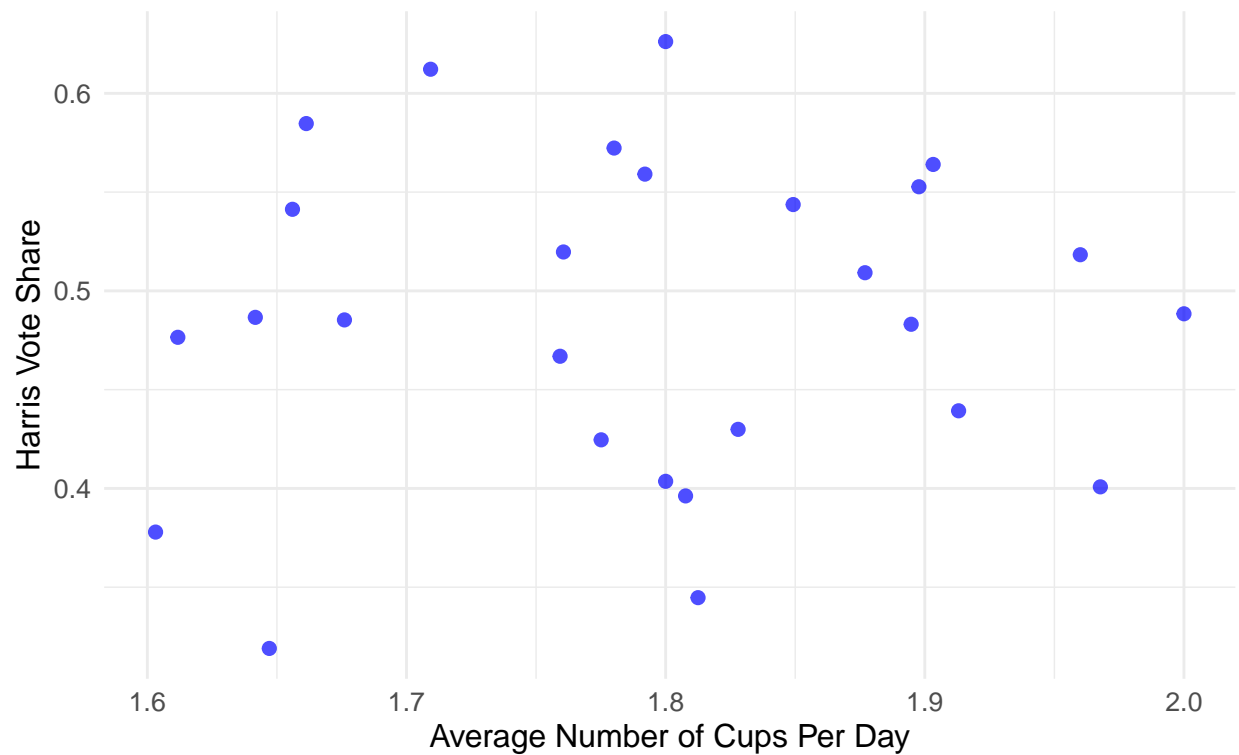


## Question 3.2

Below are a series of scatterplots showing the relationship between daily coffee consumption and voting outcomes by presidential candidate. 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.

```
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)
  )
```

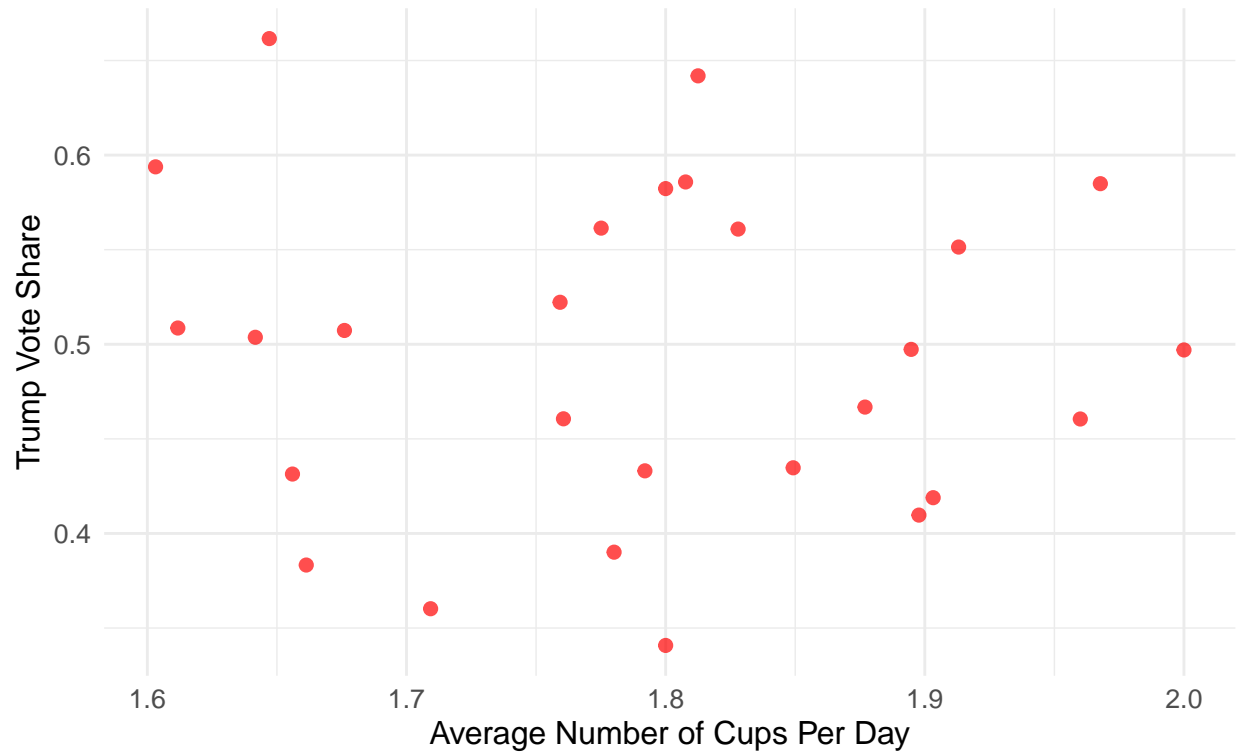
## Comparison of Surveyed Coffee Consumption and Vote Share



Only observations with >25 responses in the survey

```
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)  
  )
```

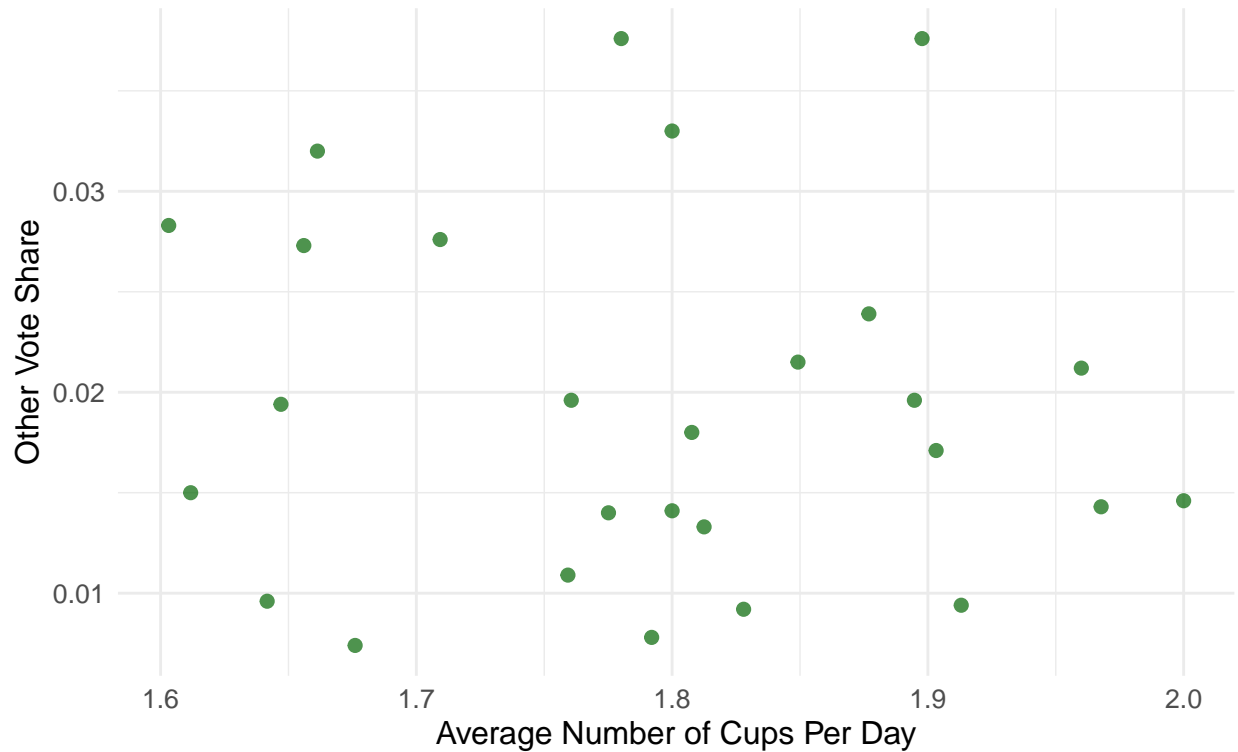
## Comparison of Surveyed Coffee Consumption and Vote Share



```
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)
  )
```



## Comparison of Surveyed Coffee Consumption and Vote Share



Only observations with >25 responses in the survey

### 4.1

Below is the code to save the merged dataset as an excel file

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

## Bonus Questions

Below is the code reading the Google trend data from an excel sheet and identifying in how many states the most popular homebrew method was also the most searched on Google (it was 4). I've also attached below the R markdown file my Python code for retrieving the data.

On examining the data, 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 there being more 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 frequent issues with access to the Google API (you can see in my Python code I had to incorporate some code to handle Google throwing errors and making continuing attempts in order to pull the data) 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

```

#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 == x])

#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

```

## M-P-Sullivan / Google Trend.py

Secret



Created now

&lt;&gt; Code - Revisions 1

## Quant Methods Lab 1 Bonus Problem

## &lt;&gt; Google Trend.py

```
1  import time
2  import pandas as pd
3  from pytrends.request import TrendReq
4  from pytrends.exceptions import TooManyRequestsError
5
6  pytrend = TrendReq()
7  print(1)
8
9  # Define Keywords (must be in chunks of 5 or fewer per Google API, had issues with larger)
10 keywords_chunk1 = ['bean-to-cup machine', 'coffee brewer', 'coffee extract', 'cold brew', 'espress
11 keywords_chunk2 = ['french press', 'instant coffee', 'keurig', 'pour over']
12 print(2)
13
14 # Function to fetch interest by region with retry and delay
15 def get_interest_by_region(keywords):
16     while True:
17         try:
18             pytrend.build_payload(
19                 kw_list=keywords,
20                 timeframe='today 12-m', # Last 12 months
21                 geo='US',
22                 gprop=''
23             )
24             time.sleep(10) # Add a delay of 10 seconds between requests to deal with 429 errors f
25             return pytrend.interest_by_region(resolution='REGION')
26         except TooManyRequestsError:
27             print("Too many requests. Retrying in 30 seconds...")
28             time.sleep(30) # Wait 30 seconds before retrying to further deal with 429s
29         except Exception as e:
30             print(f"An unexpected error occurred: {e}")
31             break
32
33 print(3)
34
35 # Fetch data for each chunk
36 interest_by_region1 = get_interest_by_region(keywords_chunk1)
37 print(4)
```

```
38 interest_by_region2 = get_interest_by_region(keywords_chunk2)
39
40 print(5)
41 # Combine results
42 interest_by_region_combined = pd.concat([interest_by_region1, interest_by_region2], axis=1)
43
44 print("\nInterest By Region (Combined):")
45 print(interest_by_region_combined.head(10))
```