DS5110 Homework 6

Ameya Santosh Gidh

2024-04-02

Part A

Problem 1

```
# Suppress startup messages from packages
suppressPackageStartupMessages(library(tidyverse))

# Load necessary libraries
library(readr)      # For reading data
library(dplyr)      # For data manipulation
library(ggplot2)    # For data visualization
library(tidyverse)  # Comprehensive data manipulation and visualization package
library(glmnet)     # For fitting Lasso and Elastic-Net regularized generalized linear models
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-8
```

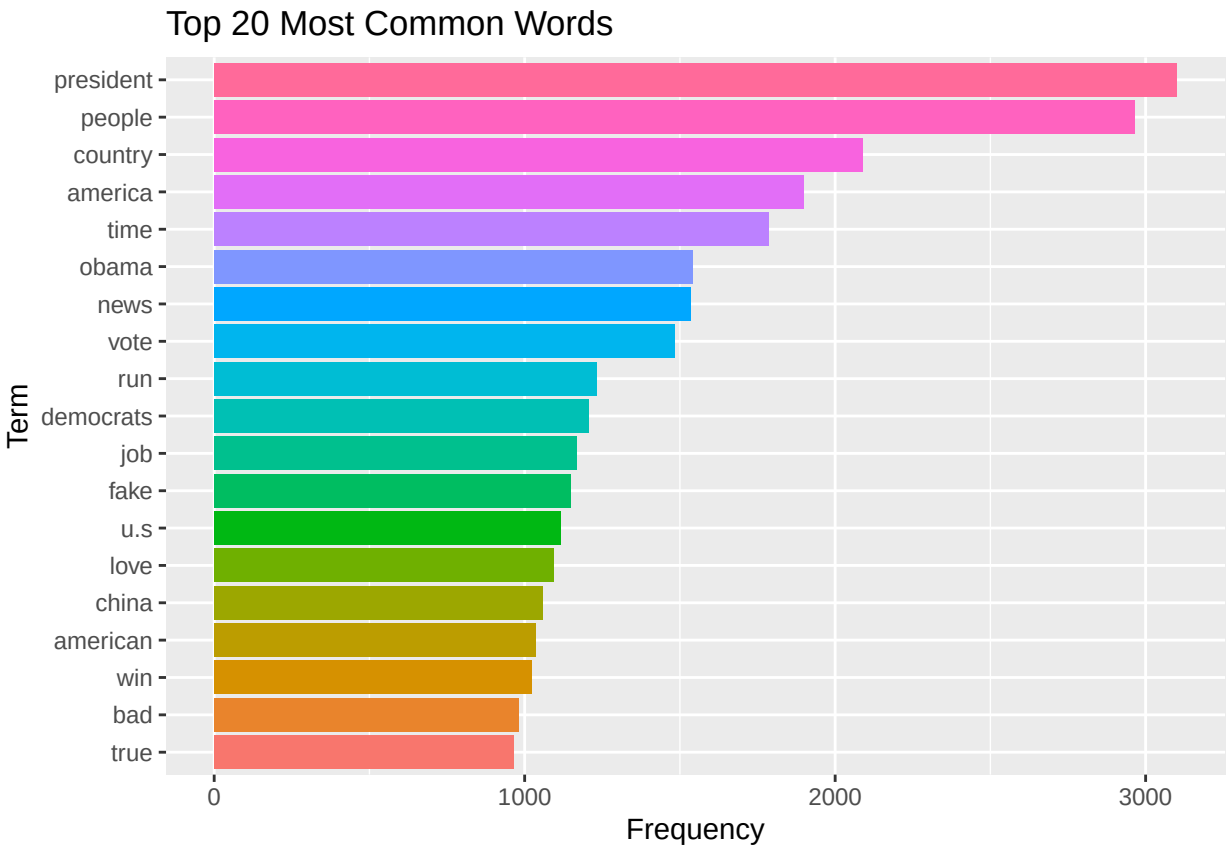
```
library(stringr)    # For string manipulation
library(tokenizers) # For tokenization
library(lubridate)  # For working with dates
library(tidytext)   # For text mining with tidy data principles

# Check if tidyverse package is installed, if not, install it
if (!requireNamespace("tidyverse", quietly = TRUE)) {
  install.packages("tidyverse")
}
```

```
# Read the data from the CSV file into a data frame
col_types_vals <- cols(
  id = col_double(),
  text = col_character(),
  isRetweet = col_logical(),
  isDeleted = col_logical(),
  device = col_character(),
  favorites = col_double(),
  retweets = col_double(),
  date = col_datetime(format = "")
```

```
)

data <- read_csv("realDonaldTrump-20201106.csv", col_types = col_types_vals)

# View the structure and contents of the data frame
# View(data)

# Tidy the data by filtering out retweets and tweets with no text content
tidy_data <- data %>%
  filter(!isRetweet, str_detect(text, "[:space:]"))

# Define a vector of words to remove from the text data
trump_names <- c("donaldtrump", "donald", "trump", "realdonaldtrump", "amp", "$")

# Tokenize the text data and remove stop words
tidy_data <- tidy_data %>%
  unnest_tokens("word", text, to_lower = TRUE) %>%
  anti_join(stop_words, by = "word")

# Filter out specific words, URLs, and mentions
tidy_data <- tidy_data %>%
  filter(!word %in% trump_names,
         !str_detect(word, "http"),
         !str_detect(word, "@"))

# Convert any non-ASCII characters to ASCII
library(stringi)
tidy_data$word <- stri_trans_general(tidy_data$word, "latin-ascii")

# Remove any rows with NA values
tidy_data <- na.omit(tidy_data)
# View(tidy_data)

# Count the frequency of each word and select the top 20
word_counts <- tidy_data %>%
  count(word) %>%
  top_n(20)
```

```
## Selecting by n
```

```
# Remove any rows where the word count is 12312
word_counts <- word_counts %>% filter(n != 12312)

# Plot the top 20 most common words
word_counts %>%
  ggplot(aes(x=reorder(word, n), y=n, fill=reorder(word, n))) +
  geom_col(show.legend=FALSE) +
  coord_flip() +
  labs(x="Term", y="Frequency", title="Top 20 Most Common Words")
```



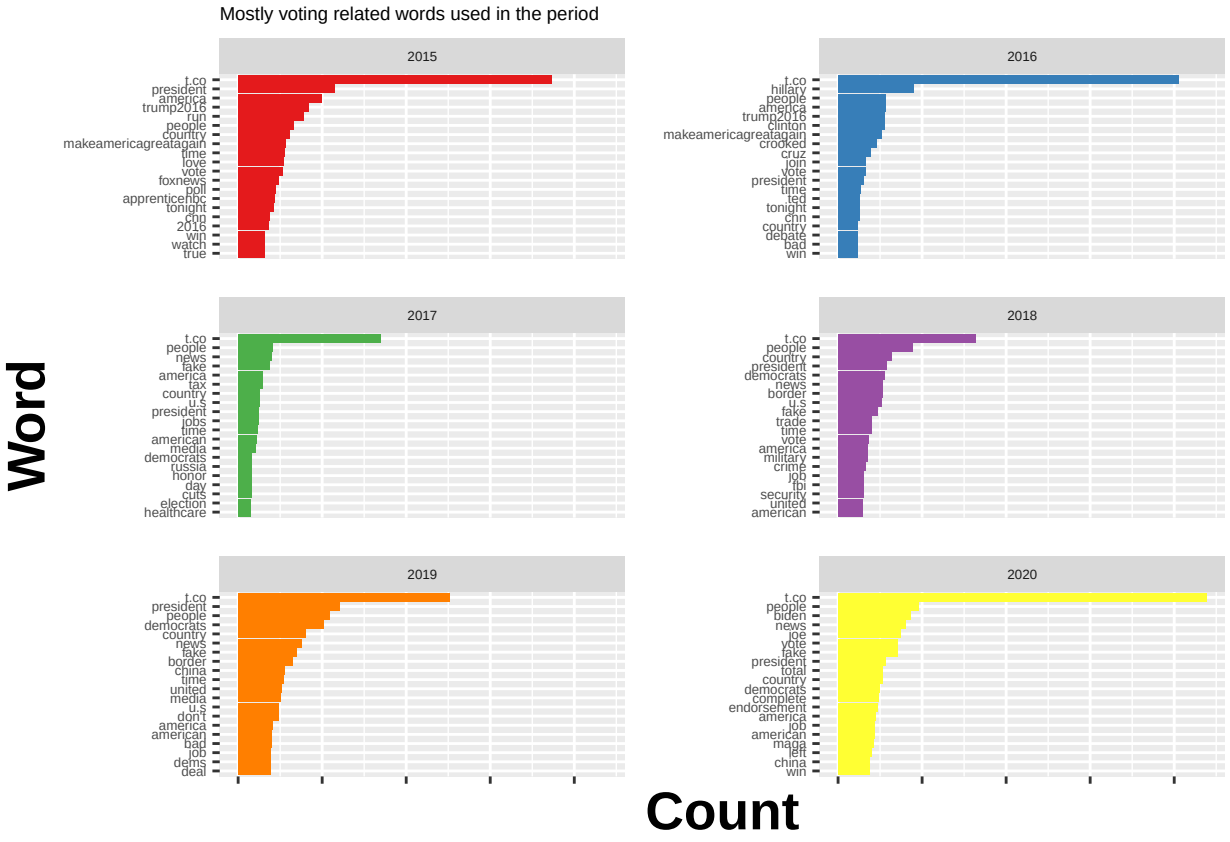
The visualization above indicates that among the top 20 tweets by Donald Trump, the most frequent theme is related to the term "President," with other tweets also revolving around political topics.

Problem 2

```
tidy_data %>% mutate(year=year(date)) %>%
  filter(year %in% 2015:2020)%>%
  count(word, year) %>%
  group_by(year) %>%
  top_n(20) %>%
  ggplot(aes(x=reorder_within(word, n, year), y=n, fill=as.factor(year))) + geom_col(show.legend=FALSE)
scale_x_reordered()+
facet_wrap(~year, scales="free_y", ncol=2)+
scale_fill_brewer(palette="Set1")+
scale_y_continuous(labels = NULL)+
labs(x="Word", y="Count", title="Mostly voting related words used in the period")+
  theme(text = element_text(size = 6), axis.title = element_text(size = 20, face = "bold"),  panel.spac
```

Donald Trump's tweets for each year from 2015-2020

```
## Selecting by n
```



The plot reveals that before the election, Donald Trump's tweets mainly focused on topics such as Make America Great Again and Hillary Clinton. After winning the election, his tweets shifted towards discussing China, jobs, and Democrats. Interestingly, prior to the election, we observe the resurgence of Make America Great Again along with mentions of Joe Biden, who eventually became President.

Part B

Problem 3

```
# Calculate TF-IDF values for each word in each year
years <- tidy_data %>%
  # Extract year from date column
  mutate(year = year(date)) %>%
  # Filter data for the years 2015 to 2020
  filter(year %in% 2015:2020) %>%
  # Count occurrences of each word within each year and calculate TF-IDF
  count(word, year) %>%
  bind_tf_idf(word, year, n)

# Select top 20 words with highest TF-IDF for each year
years %>%
  group_by(year) %>%
  top_n(20, tf_idf) %>%
  # Create a ggplot object
  ggplot(aes(x = reorder_within(word, tf_idf, year), y = tf_idf, fill = as.factor(year))) +
    # Create a bar plot with flipped coordinates
    geom_col(show.legend = FALSE) +
```

```
    coord_flip() +
    # Facet the plot by year, allowing y-axis scales to vary
    facet_wrap(~year, scale = "free_y", ncol = 3) +
    # Reorder x-axis
    scale_x_reordered() +
    # Remove y-axis labels
    scale_y_continuous(labels = NULL) +
    # Set axis and plot titles
    labs(title = "", x = "Word", y = "tf_idf") +
    # Adjust text size
    theme(axis.text = element_text(size = 5), axis.title = element_text(size = 10, face = "bold"))
```

It came as no surprise that "Make America Great Again" and "Trump2016" were the most commonly tweeted expressions, considering that 2015-2016 was characterized by election rallies. In the following years, Trump often tweeted about his policies, alongside regularly denouncing "Fake News." As expected, COVID and Coronavirus were the primary topics of Trump's discussions in 2020, which was to be anticipated.

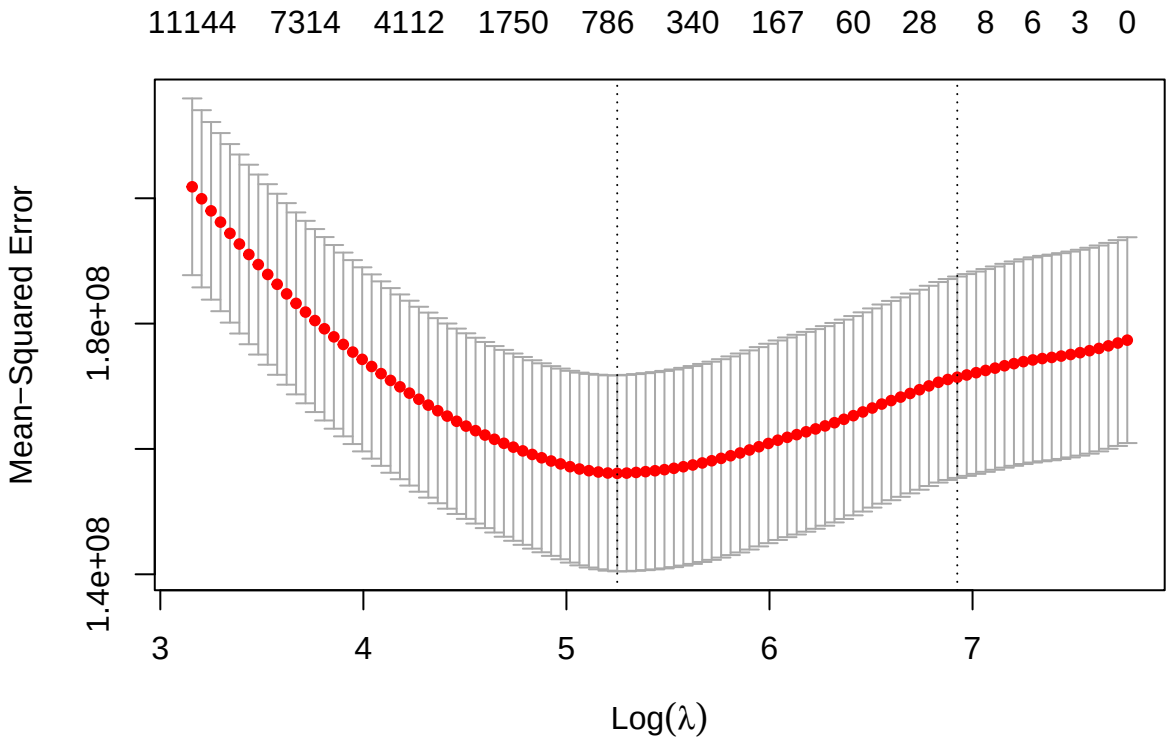
Problem 4

```
set.seed(30)
#view(tidy_data)
#excluding 2015
tidy_data <- tidy_data %>%
  filter(year(date)>=2016) %>%
  count(id, word) %>%
  cast_sparse(id, word, n)
tweetIds <- tibble(id=rownames(tidy_data))
data$id <- as.character(data$id)
```

```
#join by id
tweetIds <- tweetIds %>% left_join(data)
```

```
## Joining with `by = join_by(id)`
```

```
rt <- tweetIds$retweets
fit1 <- cv.glmnet(tidy_data, rt)
plot(fit1)
```




```
fit1
```

```
##
## Call:  cv.glmnet(x = tidy_data, y = rt)
##
## Measure: Mean-Squared Error
##
##     Lambda Index   Measure       SE Nonzero
## min  190.6    55 156105621 15633813     726
## 1se 1017.1    19 171442922 16066334      14
```

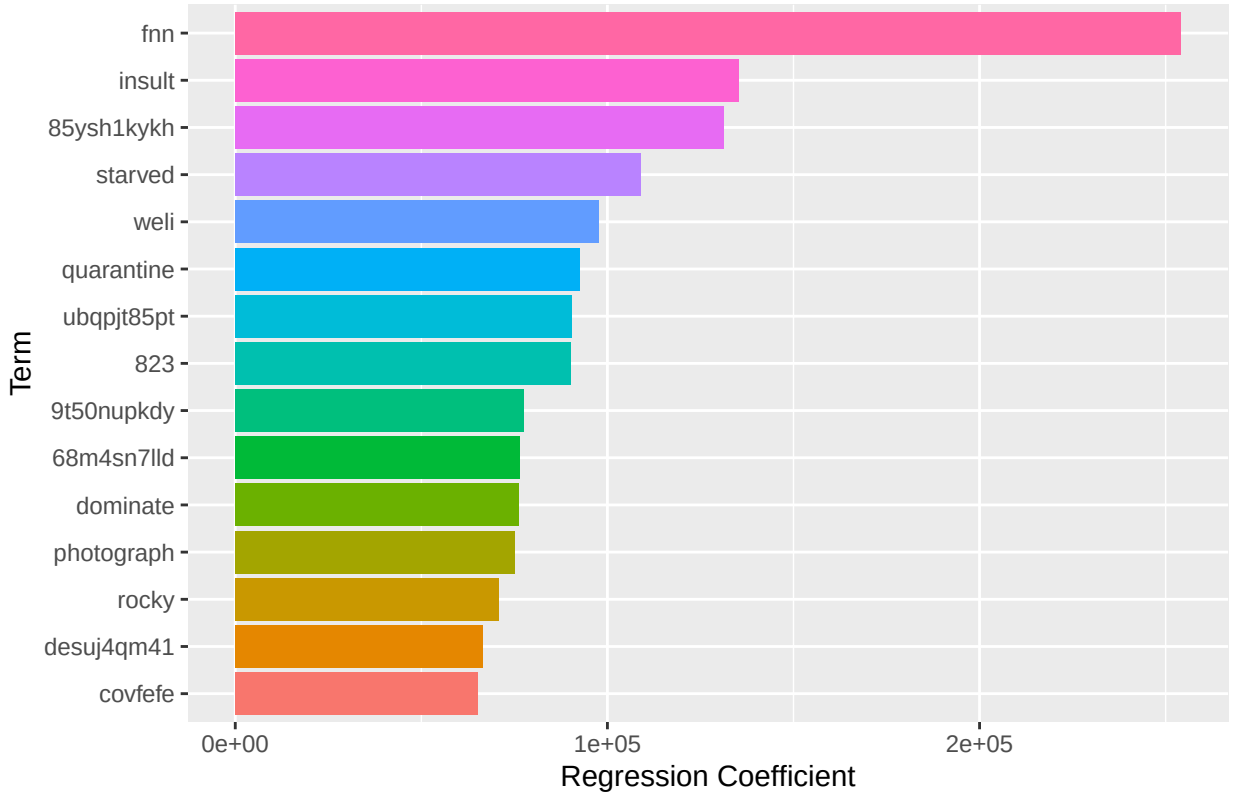
As per Mean Squared Error, the best model has lambda 199.6 and uses 557 terms.

Problem 5

```
# Extract coefficients from the model fit using lambda.min
coef_ <- coef(fit1, s = "lambda.min")

# Convert coefficients to tibble format
coef_ <- tibble(word = rownames(coef_), coef = as.numeric(coef_))

# Select the top 15 coefficients
```

```r
coef_ %>%
  top_n(15) %>%
  ggplot(aes(x = reorder(word, coef), y = coef, fill = reorder(word, coef))) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    labs(x = "Term", y = "Regression Coefficient", title = "")  # Add labels and title
```

```
## Selecting by coef
```



The retweets with the highest regression coefficient contain the hashtag "#fnn," which stands for Fox News Networks.