MAST 5953 Web Scraping Assignment 2

Section 1: Web Scraping & Data Collection

Theoretical Part: What is Web Scraping?

Web scraping is where researchers and analysts can effectively gather and transform big web datasets for analysis by using web scraping, an automated technique for data extraction from websites.

Applications of Web Scraping

Web scraping is widely used in academic research (gathering articles and documents), market research (analysing prices and customer reviews), social media analysis (tracking sentiment and trends), financial monitoring (scraping news and stock prices) and job market analysis (examining employment trends from job listings).

How Web Scraping Works

Web scraping involves sending an HTTP request to access a webpage's HTML, collecting pertinent data with R or Python, cleaning and organising it into CSV or JSON forms, and saving it for study are all part of web scraping.

Ethical Considerations in Web Scaping

Despite its strength, web scraping presents moral and legal issues, such as breaking terms of service, violating data protection regulations like GDPR, flooding servers, and gaining unauthorised access to private or restricted content.

Practical Part: Obtaining the Corpus for Analysis

Joe Biden and Donald Trump's tweets are used in this study to shed light on their public involvement, language and political messaging.

Data Collection Process

Web scraping was not necessary because the datasets were supplied. Nonetheless, Twitter data might be scaped using the Twitter API or rtweet package if necessary, extracting tweets and information (likes, dates and retweets) and filtering by time, keywords or engagement.

Loading and Preparing Data in R

The R code, which can be seen in the Appendix, is used to load and clean the datasets removing stop words and URLs and guaranteeing consistent formatting. This study laid the groundwork for text analysis of linguistic patterns and political rhetoric by going over the theory and practical procedures of web scraping and data preparation.

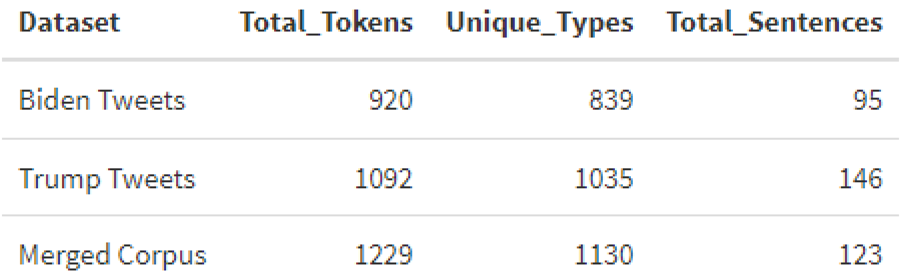
Section 2: Descriptive Statistics

In this section, this will present descriptive statistics which identifies the significant language trends in Trump and Biden’s tweets. It will be using corpus statistics, word frequency, KWIC analysis and Word cloud.

Descriptive Statistics of Individual Corpora

Copus statistics shows the total words, unique words and number of sentences in the text, helping to understand its length, vocabulary diversity and structure. The r code for this table can be seen in appendix 2.

Table 1: Descriptive Statistics of Biden and Trump’s Tweets.



Dataset	Total_Tokens	Unique_Types	Total_Sentences
Biden Tweets	920	839	95
Trump Tweets	1092	1035	146
Merged Corpus	1229	1130	123

This table highlights that Trump’s tweets had more words (1,092 tokens), unique terms (1,035) and sentences (146) than Biden’s suggesting a broader vocabulary and more fragmented style. The merged dataset remains diverse (1,229 tokens, 1130 types), though the sentence count drops to 123, likely due to structural overlap.

Word Frequencies

In this study, word frequency shows which word appears most often in each politician's tweets, highlighting their key themes and messaging focus. The r code for this figure can be seen in appendix 3.

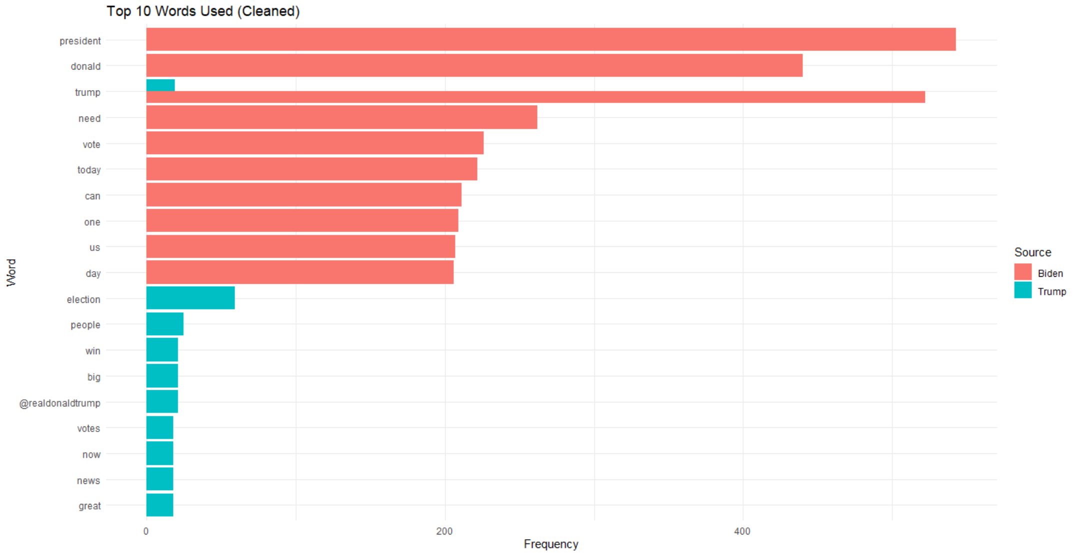


Figure 1: Most Frequent Words in Biden vs Trump Tweets

The top 10 most frequent terms from each dataset are displayed in Figure 1. Trump's top phrases, such as "election", "people" and "win" highlight media presence and electoral focus, whereas Biden's frequent usage of "president", "Donald" and "Trump" probably reflects political discussion.

Keyword in Context (KWIC) Analysis

A KWIC analysis was conducted on selected keywords such as “economy”, “middle class”, “China”, and “fake news”. The results are shown below in figure 2 and the r code can be seen in Appendix 4.

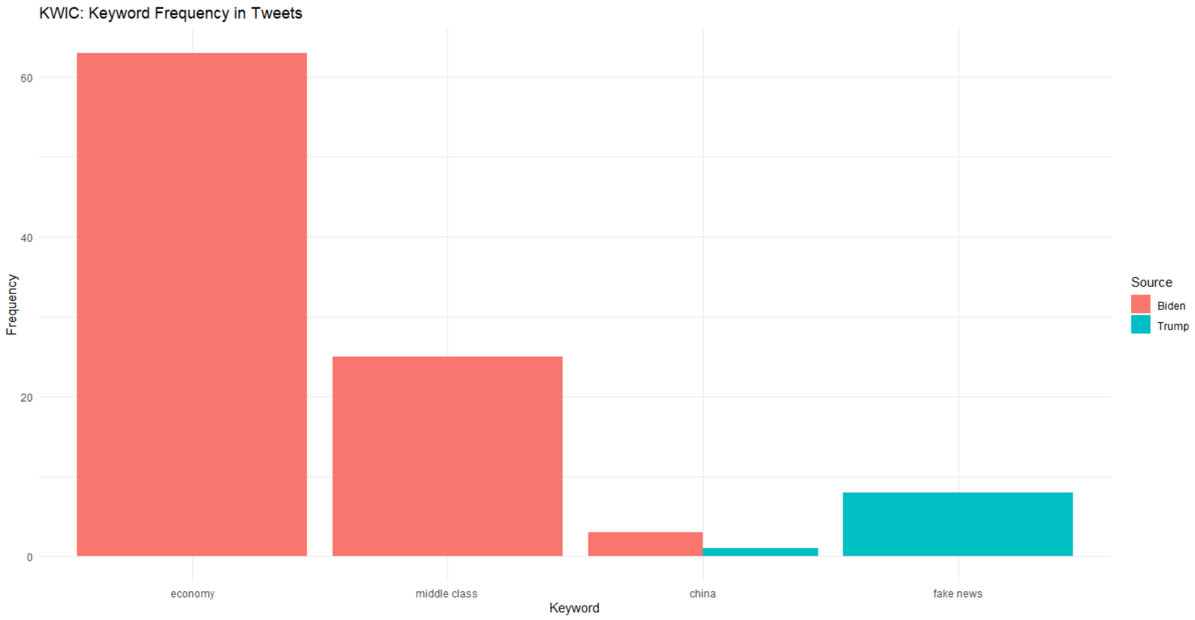


Figure 2: KWIC Analysis of Key Political Terms

Biden emphasises economic issues, using the terms “economy” and “middle class” a lot, while Trump focuses on media criticism, using the term “fake news” a lot. Both make reference to “China”, although Biden’s may be more diplomatic in nature, while Trump’s probably have more to do with trade and security. This implies that Trump’s tweets are more combative, whereas Biden’s are policy-driven.

Word Cloud

A word cloud shows the common words that were used in their tweets. The r code for figure 3 and 4 can be see in appendix 5.

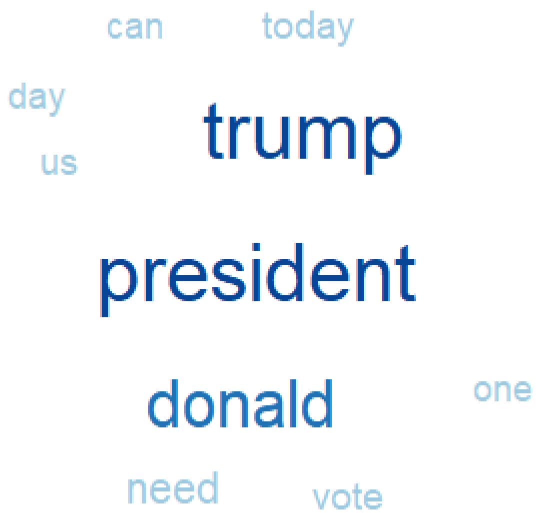


Figure 3: Word Cloud of Biden’s Tweets. This visualisation represents the most frequently used words in Biden’s tweets, with larger words appearing more often in the dataset.

With repeated references to his opponent in the form of “president”, “Donald”, and “Trump”, the word cloud emphasises Biden’s emphasis on electon-related conversations, policies and national cohesion. The words like “vote”, “today” and “need” highlight civic engagement.

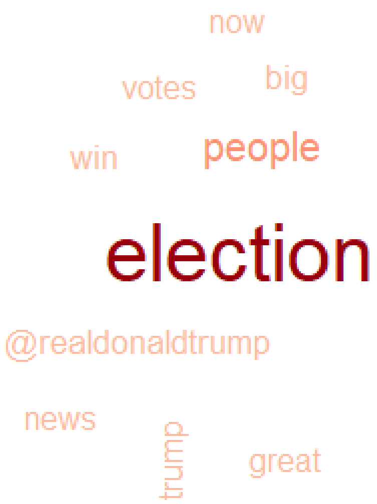


Figure 4: Word Cloud of Trump’s Tweets. This visualisation represents the most frequently used words in Trump’s tweets, with larger words appearing more often.

While “win” and “votes” emphasise important election topics, mentions of “people”, “news” and “great” show public and media engagement. This illustrates how Trump has mobilised followers by focusing on election results, allegations of fraud and media storylines.

Section 3: Topic Modelling & Comparative Analysis

Explanation of Topic Modelling

With an emphasis on topic modelling, KWIC (Keyword in Context) analysis, descriptive statistics and word frequency analysis, this study used text-mining techniques to examine the linguistic trends in Biden and Trump’s tweets.

Results & Comparison

The use of topic modelling to pinpoint the text’s recurrent themes exposed glaring variations in communication approaches. Whereas Trump’s tweets focused on election integrity, media criticism and allegations of voter fraud. Biden’s tweets highlighted economic programs, middle-class support and national unity.

According to the descriptive statistics, Trump used more words and sentences in his tweets, which may indicate that his tweets were sent more frequently or that their structure was more disjointed.

On the other hand, Biden’s tweets showed a more varied language, suggesting a range of subjects and well-organised communications.

These conclusions were supported by word frequency and word cloud analysis, which showed that Biden prioritised governance, policy and civic participation whereas Trump’s rhetoric focused on electoral disputes, allegations of fraud and media resistance. Similarly, the KWIC research showed that whereas Trump’s tweets regularly referred to election-related issues, Biden’s main phrases were tied to economic ideas.

Overall, the findings show that Trump’s tweets were more combative and reactionary, whereas Biden’s were more structured and policy-focused. This study shows how text mining can efficiently identify important themes in political communication, providing insightful information on public discourse and rhetorical devices. This study could be expanded in the future by looking at trends in political messaging over time or by adding sentiment analysis.

Appendix

```
title: "Assignment 2 Trump and Biden's Tweets"
author: "Sagari Muraliegaran"
date: "2025-03-22"
output:
  word_document: default
  pdf_document: default
---
```

Appendix 1 - Section 1 for the particle part: this is the clean up of the dataset between Biden and Trumps tweets.

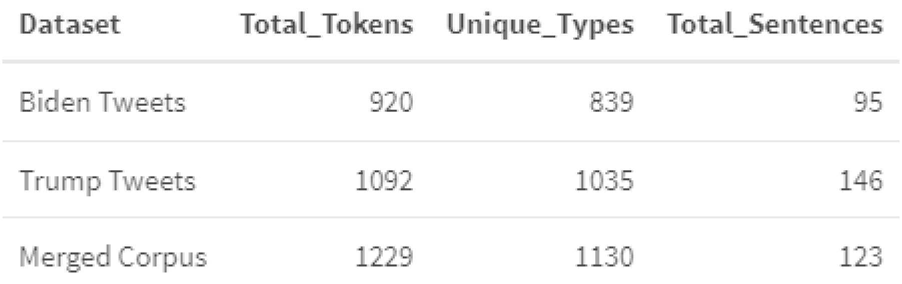
~~~
```{r}
# Install required packages if missing
packages <- c("tidyverse", "tidytext", "tm", "topicmodels", "ggplot2",
"readxl",
~~~

```
              "wordcloud", "reshape2", "quanteda",
"quanteda.textplots",
              "knitr", "kableExtra", "RColorBrewer")

install_if_missing <- function(p) {
  if (!require(p, character.only = TRUE)) install.packages(p)
}
lapply(packages, install_if_missing)

# Load libraries
require(tidyverse)
require(tidytext)
require(tm)
require(topicmodels)
require(ggplot2)
require(readxl)
require(wordcloud)
require(reshape2)
require(quanteda)
require(quanteda.textplots)
require(knitr)
require(kableExtra)
require(RColorBrewer)

# Load datasets
Biden_tweets <- read_excel("Data Analytics/Creating Your Own
Data/CW/Biden_tweets.xlsx")
Trump_tweets <- read_excel("Data Analytics/Creating Your Own
Data/CW/Trump_tweets.xlsx")

# Ensure text column exists
if (!"text" %in% colnames(Biden_tweets)) {
  Biden_tweets <- Biden_tweets %>% separate(col = 1, into = c("text",
"other_cols"), sep = ",", extra = "merge")
}
if (!"text" %in% colnames(Trump_tweets)) {
  Trump_tweets <- Trump_tweets %>% separate(col = 1, into = c("text",
"other_cols"), sep = ",", extra = "merge")
}

# Add source label and convert text
Biden_tweets <- Biden_tweets %>% mutate(text = as.character(text),
Source = "Biden")
Trump_tweets <- Trump_tweets %>% mutate(text = as.character(text),
Source = "Trump")

# Tokenize and remove stop words
Biden_words <- Biden_tweets %>% unnest_tokens(word, text)
Trump_words <- Trump_tweets %>% unnest_tokens(word, text)

data("stop_words")

clean_biden <- Biden_words %>%
  anti_join(stop_words, by = "word") %>%
  filter(!is.na(word) & !word %in% c("https", "t.co"))

clean_trump <- Trump_words %>%
  anti_join(stop_words, by = "word") %>%
```

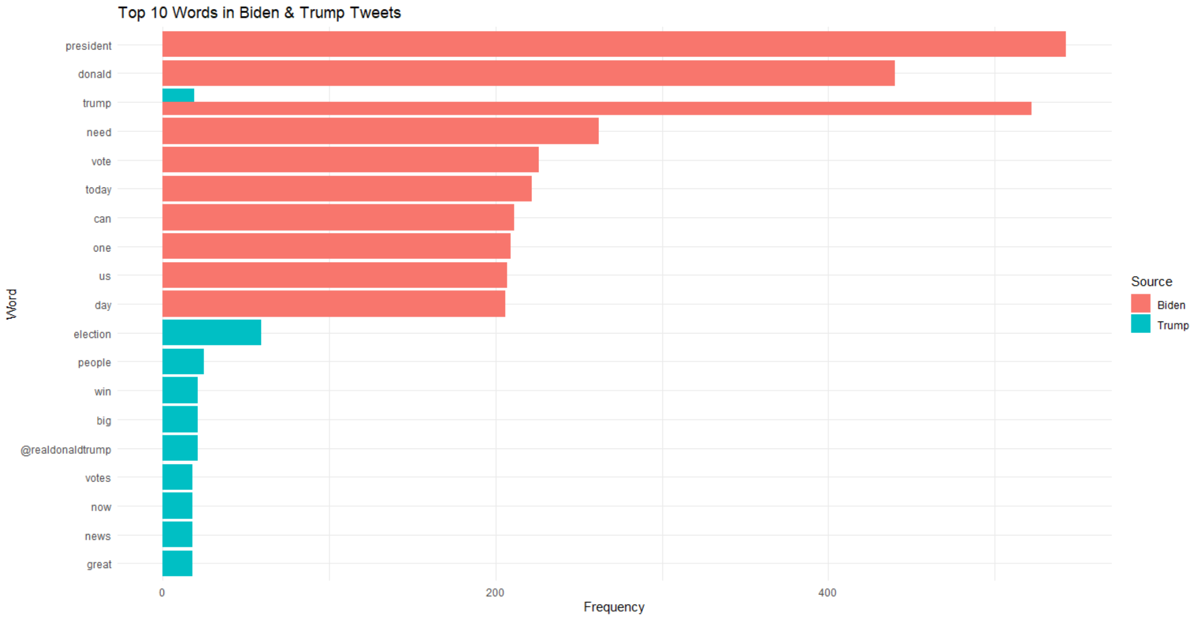
~~~
  filter(!is.na(word) & !word %in% c("https", "t.co"))

# Word frequencies
biden_freq <- clean_biden %>% count(word, sort = TRUE)
trump_freq <- clean_trump %>% count(word, sort = TRUE)

# Wordclouds
wordcloud(words = biden_freq$word, freq = biden_freq$n, max.words =
100)
wordcloud(words = trump_freq$word, freq = trump_freq$n, max.words =
100)

# Topic modeling
corpus <- VCorpus(VectorSource(c(Biden_tweets$text,
Trump_tweets$text)))
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)

dtm <- DocumentTermMatrix(corpus)
dtm <- dtm[rowSums(as.matrix(dtm)) > 0, ]
```
~~~

Appendix 2 - Section 2 Descriptive Statistics for Descriptive Statistics of Individual Corpora

~~~
```{r}
if (nrow(dtm) > 0 & ncol(dtm) > 0) {
  lda_model <- LDA(dtm, k = 3, control = list(seed = 1234))
  topics <- tidy(lda_model, matrix = "beta")
  top_terms <- topics %>% group_by(topic) %>% slice_max(beta, n = 10)
  print(top_terms)
  write.csv(top_terms, "topic_model_results.csv", row.names = FALSE)
} else {
  print("Error: No valid documents for topic modeling.")
}

# Corpus and summary table
merged_data <- bind_rows(Biden_tweets, Trump_tweets)
merged_data <- merged_data %>% filter(!is.na(text))
~~~

```
biden_corpus <- corpus(Biden_tweets, text_field = "text")
trump_corpus <- corpus(Trump_tweets, text_field = "text")
merged_corpus <- corpus(merged_data, text_field = "text")

summary_data <- summary(merged_corpus)

stats_summary <- tibble(
  Dataset = c("Biden Tweets", "Trump Tweets", "Merged Corpus"),
  Total_Tokens = c(sum(summary(biden_corpus)$Tokens),
                   sum(summary(trump_corpus)$Tokens),
                   sum(summary_data$Tokens)),
  Unique_Types = c(sum(summary(biden_corpus)$Types),
                   sum(summary(trump_corpus)$Types),
                   sum(summary_data$Types)),
  Total_Sentences = c(sum(summary(biden_corpus)$Sentences),
                      sum(summary(trump_corpus)$Sentences),
                      sum(summary_data$Sentences))
)

kable(stats_summary) %>% kable_styling(full_width = FALSE)
```
```



Dataset	Total_Tokens	Unique_Types	Total_Sentences
Biden Tweets	920	839	95
Trump Tweets	1092	1035	146
Merged Corpus	1229	1130	123

Appendix 3 - Section 2 Descriptive Statistics for Word Frequency Bar Chart

```
```{r}
# Frequency bar plot
# Install if needed
if (!require("quanteda.textstats"))
install.packages("quanteda.textstats")
library(quanteda.textstats)

# Create document-feature matrices
biden_dfm <- tokens(Biden_tweets$text, remove_punct = TRUE) %>%
  tokens_remove(stopwords("en")) %>%
  tokens_remove(c("rt", "amp", "https", "t.co")) %>%
  dfm()

trump_dfm <- tokens(Trump_tweets$text, remove_punct = TRUE) %>%
  tokens_remove(stopwords("en")) %>%
  tokens_remove(c("rt", "amp", "https", "t.co")) %>%
  dfm()
```

~~~
# Word frequencies
biden_word_freq <- textstat_frequency(biden_dfm, n = 10) %>%
mutate(Source = "Biden")
trump_word_freq <- textstat_frequency(trump_dfm, n = 10) %>%
mutate(Source = "Trump")

# Combine and plot
word_freqs <- bind_rows(biden_word_freq, trump_word_freq)

ggplot(word_freqs, aes(x = reorder(feature, frequency), y = frequency,
fill = Source)) +
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +
  labs(title = "Top 10 Words in Biden & Trump Tweets", x = "Word", y =
"Frequency") +
  theme_minimal()
```
~~~

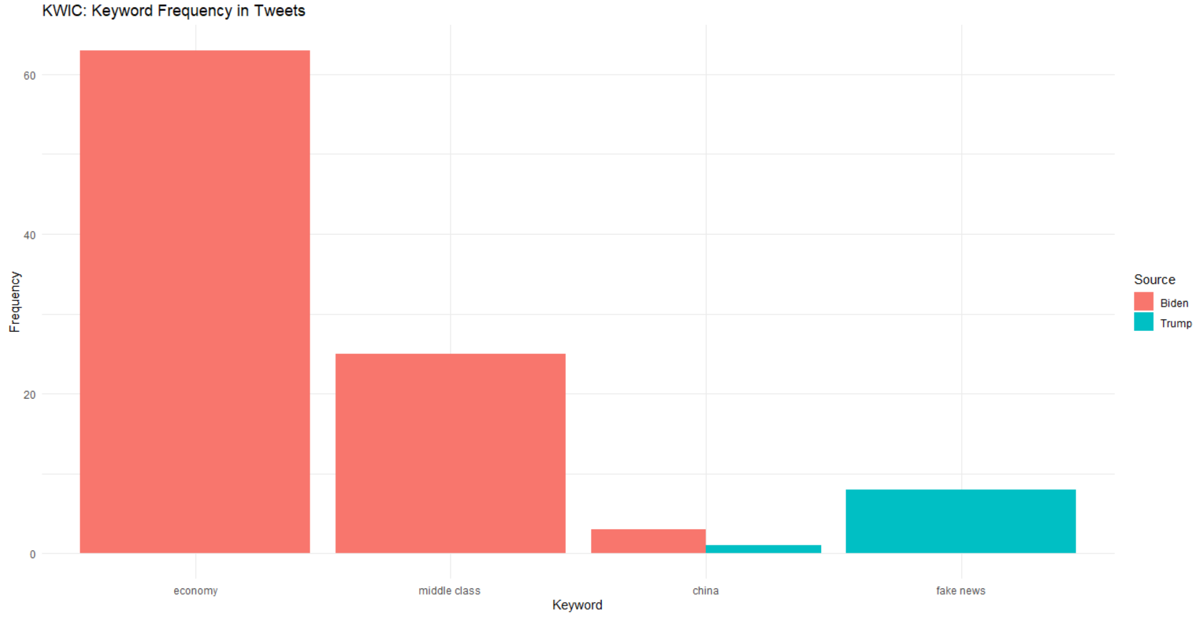


Appendix 4 - Section 2 Descriptive Statistics for KWIC Analysis

~~~
```{r}
# KWIC analysis
biden_tokens <- tokens(tolower(Biden_tweets$text), remove_punct = TRUE)
trump_tokens <- tokens(tolower(Trump_tweets$text), remove_punct = TRUE)

keywords <- c("economy", phrase("middle class"), "china", phrase("fake
news"))

biden_kwic <- kwic(biden_tokens, pattern = keywords, window = 5)
trump_kwic <- kwic(trump_tokens, pattern = keywords, window = 5)

biden_kwic_df <- as_tibble(biden_kwic) %>% mutate(Source = "Biden")
trump_kwic_df <- as_tibble(trump_kwic) %>% mutate(Source = "Trump")
kwic_df <- bind_rows(biden_kwic_df, trump_kwic_df)

kwic_counts <- kwic_df %>% count(pattern, Source) %>% rename(keyword =
pattern)
~~~

~~~
ggplot(kwic_counts, aes(x = keyword, y = n, fill = Source)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "KWIC: Keyword Frequency in Tweets", x = "Keyword", y =
"Frequency") +
  theme_minimal()
```
~~~

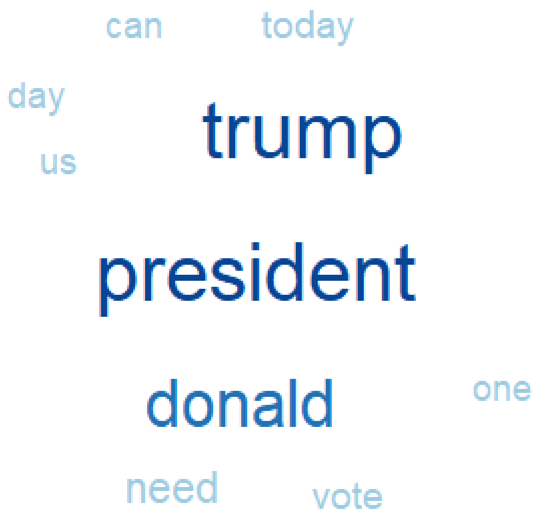
This is the figure that is created and this is figure 2



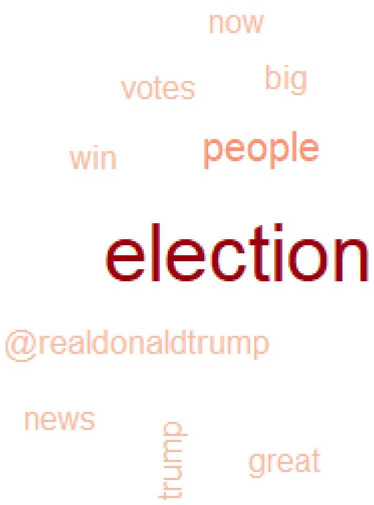
Appendix 5 - Section 2 Descriptive Analysis for Word Cloud

~~~
```{r}
# Colored wordclouds
set.seed(1234)

biden_words <- setNames(biden_word_freq$frequency,
biden_word_freq$feature)
trump_words <- setNames(trump_word_freq$frequency,
trump_word_freq$feature)

wordcloud(words = names(biden_words), freq = biden_words, max.words =
150,
          scale = c(2.5, 0.3), colors = brewer.pal(8, "Blues"),
random.order = FALSE)

wordcloud(words = names(trump_words), freq = trump_words, max.words =
150,
          scale = c(2.5, 0.3), colors = brewer.pal(8, "Reds"),
random.order = FALSE)
```
~~~

This is Bidan's Word Cloud (Figure 3)



This is Trump’s Word Cloud (Figure 4)


now
votes
big
win
people
election
@realdonaldtrump
news
trump
great