Trump Social Descriptive Data

Contents

Load the packages, data and functions

```
rm(list=ls())
require(here)
require(stringr)
require(dplyr)
require(ggplot2)
require(lubridate)
require(RColorBrewer)
require(wordcloud)
require(tidyr)
require(randomForest)
require(tidytext)
require(textstem)
require(tidyverse)
require(tm)
require(SnowballC)
require(quanteda.textplots)
require(quantmod)
require(alphavantager)
require(quanteda)
require(rvest)
require(httr)
require(xml2)
require(textdata)
require(sentimentr)
require(syuzhet)
require(text)

truths_raw <- read.csv(here("data/political_data", "truths_new.csv"))
snp_raw <- read.csv(here("data/market_data", "SPY_24_25.csv"))

source(here("helperfunctions/truths_cleaning_function.R"))
```

Process the data with the helperfunction

First we process the truth social posts ("truths") using our custom function.

```
truths <-truths_processer(truths_raw) # can take one or two minutes
#head(truths)
```

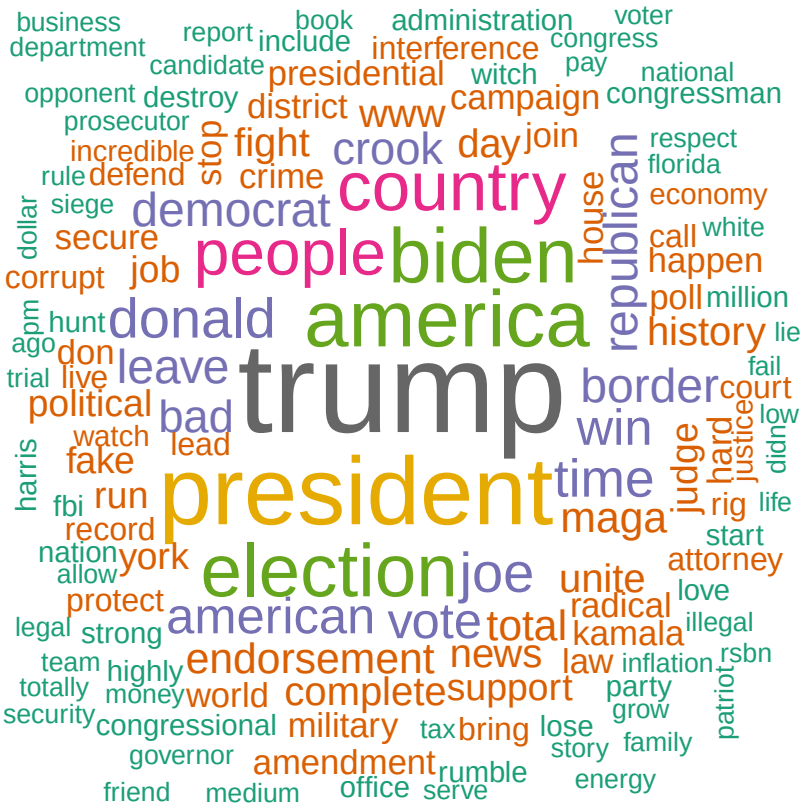
This does multiple things. Mainly it turns the scrapped data that is a string per post, cleans it and tokenizes the post. It separates the date and prepares the date to make the plots below by creating many columns.

The second part tokenizes the tweets and lemmatizes them. This means each word becomes an "obersvation and all observations are collected on the post level. Then the words are also lemmatized, meaning reduced to their basic form. For example is becomes be, runnin or ran become run ect.

#Wordcloud

We remove the Tweets that only contain links or nothing at all from the dataset

```
truths_main <- truths %>%
  filter(link != 1, media != 1)

all_tokens <- unlist(truths_main$tokens)

word_freqs <- table(all_tokens)

count_token_occurrences <- function(data, word) {
  sum(sapply(data$tokens, function(token_list) {
    sum(token_list == word)}))}

count_token_occurrences(truths_main, "https")

wordcloud(
  words = names(word_freqs),
  freq = as.numeric(word_freqs),
  min.freq = 300,  # adjust depending on how important it is
  max.words = 400,
  random.order = FALSE,
  colors = brewer.pal(8, "Dark2"))
```

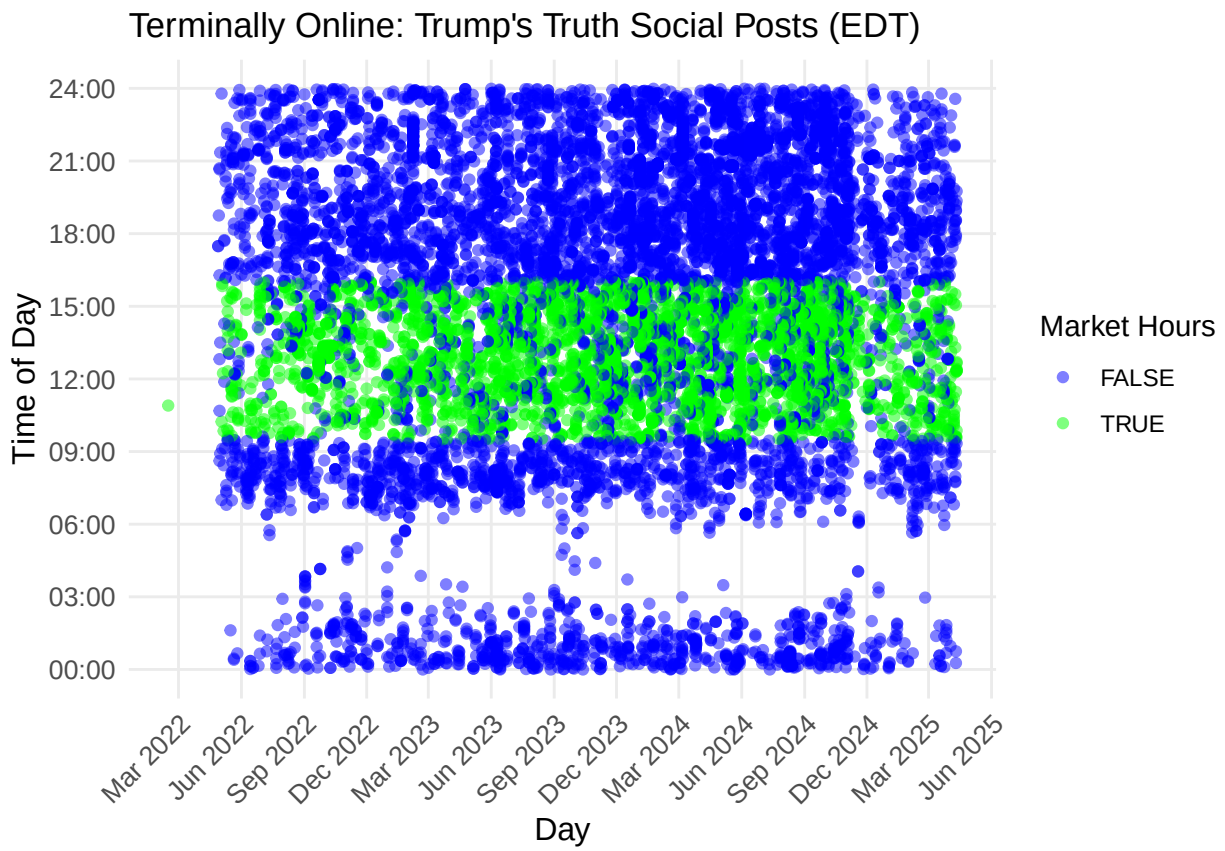


From this wordcloud we can gather some of the most used words by trump. Now the question is: how much does he post during open markets?

For that we can plot each post on a time plot:

```
weekday_names <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")

truths_main <- truths_main %>%
  mutate(
    weekday = weekday_names[wday(day, week_start = 1)],
    is_market_hour = case_when(
      weekday %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday") &
        time_shifted >= (-2.5) & time_shifted <= 4 ~ TRUE,
      TRUE ~ FALSE))
```

```
ggplot(truths_main, aes(x = day, y = time_shifted)) +
  geom_point(aes(color = is_market_hour), alpha = 0.5) +

  # Custom colors for market vs non-market hours
  scale_color_manual(values = c("FALSE" = "blue", "TRUE" = "green")) +

  scale_y_continuous(
    breaks = seq(-12, 12, by = 3),
    labels = c("00:00", "03:00", "06:00", "09:00", "12:00", "15:00", "18:00", "21:00", "24:00")
  ) +
  labs(title = "Terminally Online: Trump's Truth Social Posts (EDT)",
       x = "Day",
       y = "Time of Day",
       color = "Market Hours") +

  theme_minimal() +

  scale_x_date(
    date_labels = "%b %Y",
    date_breaks = "3 months"
  ) +

  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +

  theme(
    panel.grid.minor = element_blank(),
    panel.grid.major = element_line(linewidth = 0.5),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10))
```



Now lets get only those tweets

```
truths_market <- truths_main %>%
  filter(is_market_hour != FALSE)
```

Get the open and close

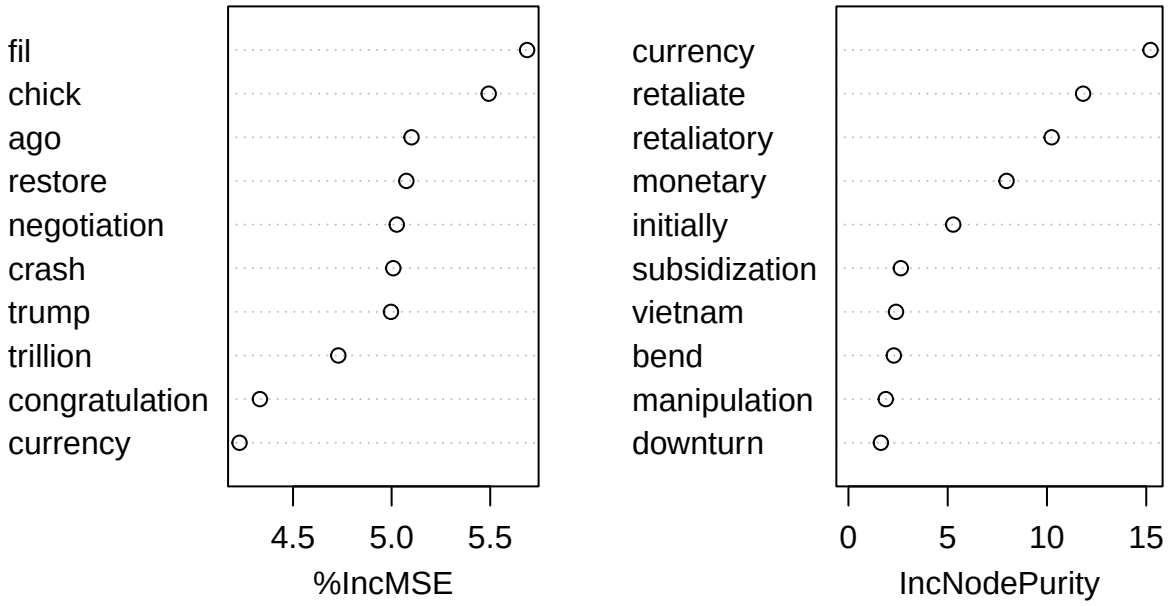
```
snp_data <- snp_raw %>%
  mutate(timestamp = dmy_hm(timestamp))


truths_market <- truths_market %>%
  mutate(post_time = ymd_hms(date_time_parsed))  # Rename or reuse as needed

# Join on exact datetime
truths_market <- truths_market %>%
  left_join(snp_data, by = c("post_time" = "timestamp"))

truths_market <- truths_market %>%
  mutate(minute_return = Open - Close)

truths_market <- truths_market %>%
  filter(!is.na(Open))
```

```r
# First, ensure tokens are characters and unnest them
truths_unnested <- truths_market %>%
  select(minute_return, tokens) %>%#only select dependent value and tokens
  mutate(tokens = lapply(tokens, as.character)) %>% # turn them into characters
  unnest(tokens) %>% #unnest the tokens
  mutate(token_present = 1)# just shows if any token is present or if any row is empty
# at this point we have a dataframe that has one column where each token is on a separate row (so if a
#each with one token)


# Now, we widen the data, making sure only tokens are pivoted
token_wide_df <- truths_unnested %>%
  pivot_wider(
    names_from = tokens,
    values_fn = max,
    values_from = token_present,
  )
# This basically makes it so each unique token has a column and there is only 1 row for each unique val
# wide dataframe where each column represents a token that is NA if that token doesnt exist in a post (


# Check the result
print(token_wide_df)
```

```r
#Replace NA with 0
token_wide_df[ , -1] <- token_wide_df[ , -1] %>% replace(is.na(.), 0) #due to problems to do it during
colnames(token_wide_df) <- make.names(colnames(token_wide_df))
rfdf <- na.omit(token_wide_df)
```

```r
# Random Forest Model
train_index <- sample(1:nrow(rfdf), size = 0.8 * nrow(rfdf))  # 80% training data
train_data <- rfdf[train_index, ]
test_data <- rfdf[-train_index, ]

rf_model <- randomForest(minute_return ~ ., data = train_data, importance = TRUE, ntree = 200)
#importance(rf_model)
varImpPlot(rf_model, n.var = 10)
```

rf_model


fil
chick
ago
restore
negotiation
crash
trump
trillion
congratulation
currency
4.5
5.0
5.5
%IncMSE
currency
retaliate
retaliatory
monetary
initially
subsidization
vietnam
bend
manipulation
downturn
0
5
10
15
IncNodePurity

Assign Sentiment Scores to tweets

```
nrc_scores <- get_nrc_sentiment(truths_market$post)

truths_sentiment <- bind_cols(truths_market, nrc_scores)
```

This is the Tweet wordcloud

He was banned in 2021 after Jan 6 so this is when the tweets stop

```
tweets <- read.csv(here("data/political_data", "tweets.csv"))

#Only keep original Tweets
tweets <- tweets %>% filter(isRetweet != "t")
tokens <- tokens(tweets$text)
dfm <- dfm(tokens)

corpus_tweets <- corpus(tweets$text)

additional_words <- c("@realdonaldtrump")
all_stopwords <- c(stopwords("en"), additional_words)

tokens_tweets <- tokens(corpus_tweets,
                        remove_punct = TRUE,
                        remove_numbers = TRUE)

tokens_tweets <- tokens_remove(tokens_tweets, all_stopwords)  # Remove stopwords

dfm_tweets <- dfm(tokens_tweets)

textplot_wordcloud(dfm_tweets, min_count = 400, color = RColorBrewer::brewer.pal(8, "Dark2"))
```