Tweet analysis

Daniel Shang

Load the necessary libraries

```
library(gutenbergr)
```

```
## Warning: package 'gutenbergr' was built under R version 4.0.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(stringr)
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.0.3
```

```
library(ggplot2)
library(SnowballC)
```

```
## Warning: package 'SnowballC' was built under R version 4.0.3
```

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.0.3
```

```
## Loading required package: RColorBrewer
```

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.0.3
```

```
## Loading required package: NLP
```

```
## Warning: package 'NLP' was built under R version 4.0.3
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 4.0.3
```

```
library(latexpdf)
```

```
## Warning: package 'latexpdf' was built under R version 4.0.3
```

Data cleaning and preparation

```
# Load the data
tweet = read.csv('C:/Users/34527/Desktop/vaccine_tweet.csv')

# Take a glance at the data
summary(tweet)
```

```
##    UserName            Handle           Timestamp            Text
##  Length:50006       Length:50006       Length:50006       Length:50006
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##    Comments            Likes             Retweets
##  Length:50006       Length:50006       Length:50006
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

```
# Drop the column that is not necessary
to_drop = c('UserName')
tweet = tweet[,colnames(tweet)!= to_drop]

# Check if there is any missing data
sum(tweet['Text'] == '')
```

```
## [1] 0
```

```
sum(is.null(tweet['Text']))
```

```
## [1] 0
```

```r
sum(is.na(tweet[,'Text']))
```

```
## [1] 0
```

Data visualization

```r
# Unnest the tokens in each Tweet and group them by Timestamp column
tidy_tweet = tweet %>% select(Timestamp, Text) %>% unnest_tokens('word', Text)

# Remove stop words
tidy_tweet = tidy_tweet %>% anti_join(stop_words)
```

```
## Joining, by = "word"
```

```r
# Remove numbers
tidy_tweet = tidy_tweet[-grep("\\b\\d+\\b", tidy_tweet$word),]

# Remove spaces
tidy_tweet$word = gsub('\\s+', '', tidy_tweet$word)

# Stemming, the process in which all the words are reduced to its most basic form. For
## example, 'producing' is reduced to 'produce'
tidy_tweet = tidy_tweet %>% mutate_at('word', funs(wordStem((.), language = 'en')))
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```r
# Manually remove the words that would not help with the analysis
tidy_tweet = tidy_tweet %>% filter(!(word == 'â' | word == 'https' | word == 'à' |
    word == 'http' | word == 'ð' | word == 'vaccine' | word == 'covid' |
    word == 'coronavirus' | word == 'vaccin' | word == 'covid19'))
```

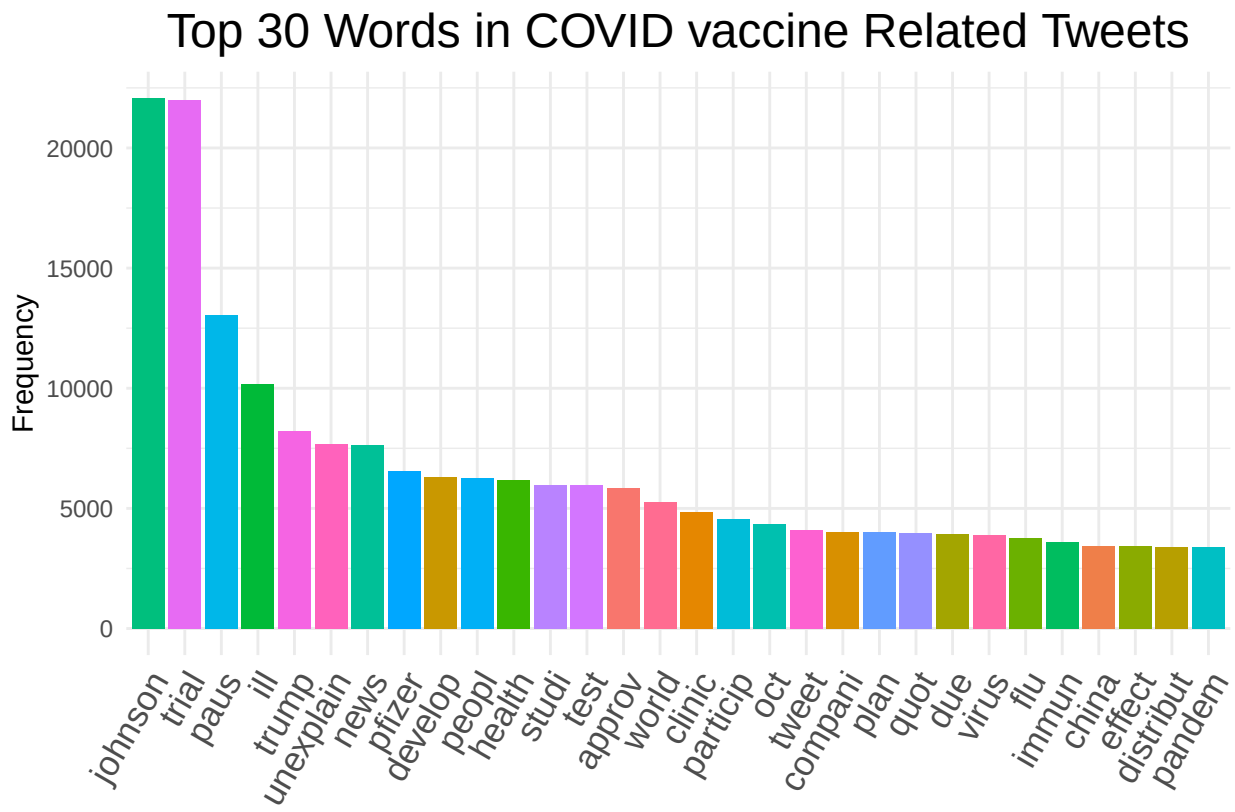
```r
# Create a document-term matrix (DTM) and inspect the previous five elements
tidy_tweetDTM = tidy_tweet %>% count(Timestamp, word) %>% cast_dtm(Timestamp, word, n)
inspect(tidy_tweetDTM[1:5, 1:5])
```

```
## <<DocumentTermMatrix (documents: 5, terms: 5)>>
## Non-/sparse entries: 7/18
## Sparsity           : 72%
```

```
## Maximal term length: 7
## Weighting          : term frequency (tf)
## Sample             :
##                             Terms
## Docs                         bid biggest china control countri
##   2020-10-09T20:47:16.000Z   1       1     5       1       1
##   2020-10-09T20:47:22.000Z   0       0     3       0       1
##   2020-10-09T20:47:25.000Z   0       0     0       0       0
##   2020-10-09T20:47:51.000Z   0       0     0       0       0
##   2020-10-09T20:48:00.000Z   0       0     0       0       0
```

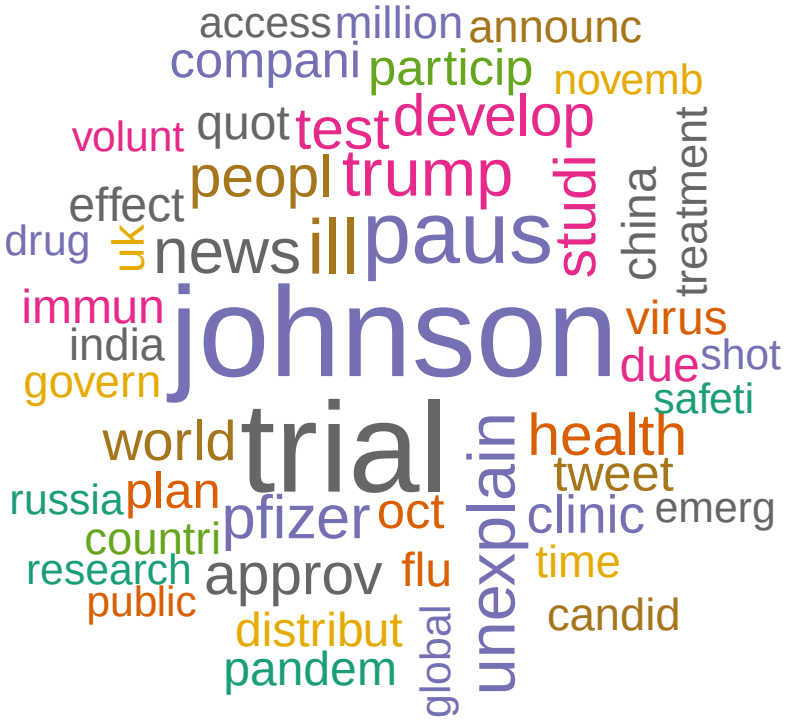
```
# Manually remove the words that do not add insigts to the analysis
top_word = tidy_tweet %>% count(word) %>% arrange(desc(n))
```

```
# Extract and visualize the top 30 words based on the number of times they are mentioned
top_word %>% slice(1:30) %>% ggplot(aes(x = reorder(word, -n), y = n, fill = word)) +
  geom_bar(stat = 'identity') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 13)) +
  theme(plot.title = element_text(hjust = 0.5, size = 18)) +
  ylab('Frequency') +
  xlab('') +
  ggtitle('Top 30 Words in COVID vaccine Related Tweets') + guides(fill = FALSE)
```



```
# Visualize the words using word cloud
wordcloud(top_word$word, top_word$n, random.order = FALSE, rot.per = 0.2, scale = c(4, 1),
          max.words = 50, random.color = TRUE, colors = brewer.pal(11, 'Dark2'))
```

```
## Warning in brewer.pal(11, "Dark2"): n too large, allowed maximum for palette Dark2 is 8
## Returning the palette you asked for with that many colors
```



Sentiment Analysis

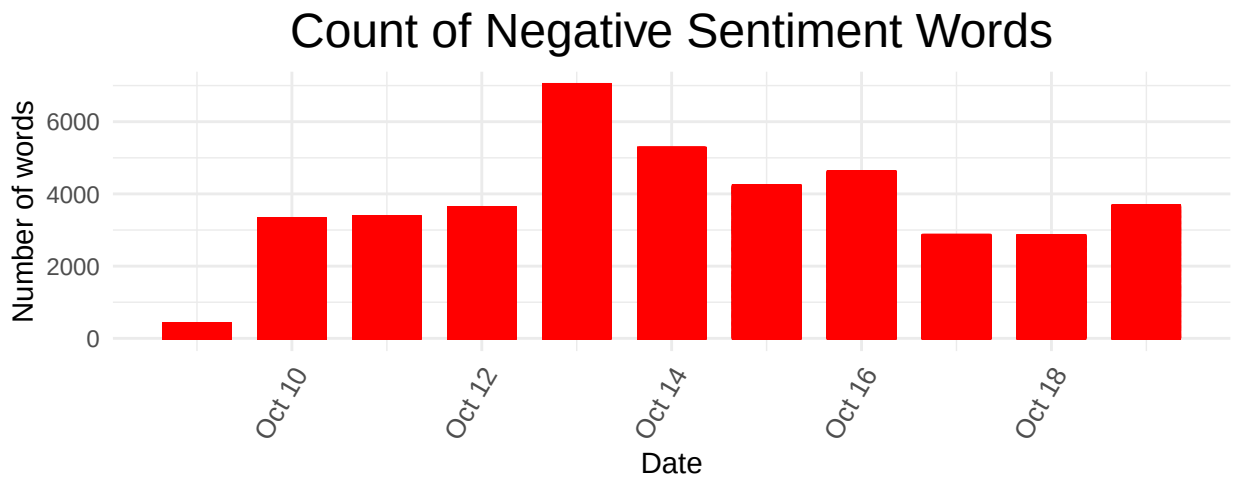
```
# Use the function in tidytext package to determine the sentiment of words in each tweet.
## Then, count the number of positive and negative words in each tweet.
tidy_tweet_sentiment = tidy_tweet %>% inner_join(get_sentiments('bing')) %>%
  count(Timestamp, sentiment)
```

```
## Joining, by = "word"
```

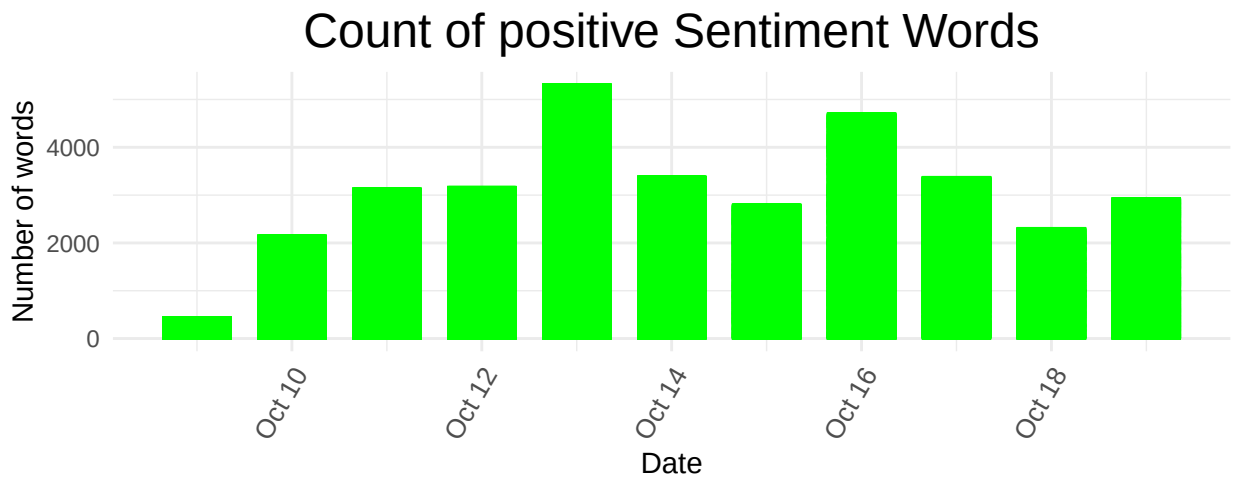
```
tidy_tweet_sentiment$Timestamp = substr(tidy_tweet_sentiment$Timestamp, 1, 10)

tidy_tweet_sentiment$Timestamp =
  as.Date(tidy_tweet_sentiment$Timestamp, format = '%Y-%m-%d')
```

```
# Count the number of negative words for each day over the period of which the data
## covered
tidy_tweet_sentiment %>% filter(sentiment == 'negative') %>%
```

```
  ggplot(aes(x = Timestamp, y = n)) +
  geom_bar(stat = 'identity', color = 'red', width = 0.7) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 10)) +
  theme(plot.title = element_text(hjust = 0.5, size = 18)) +
  ylab('Number of words') +
  xlab('Date') +
  ggtitle('Count of Negative Sentiment Words') +
  theme(aspect.ratio = 1/4)
```

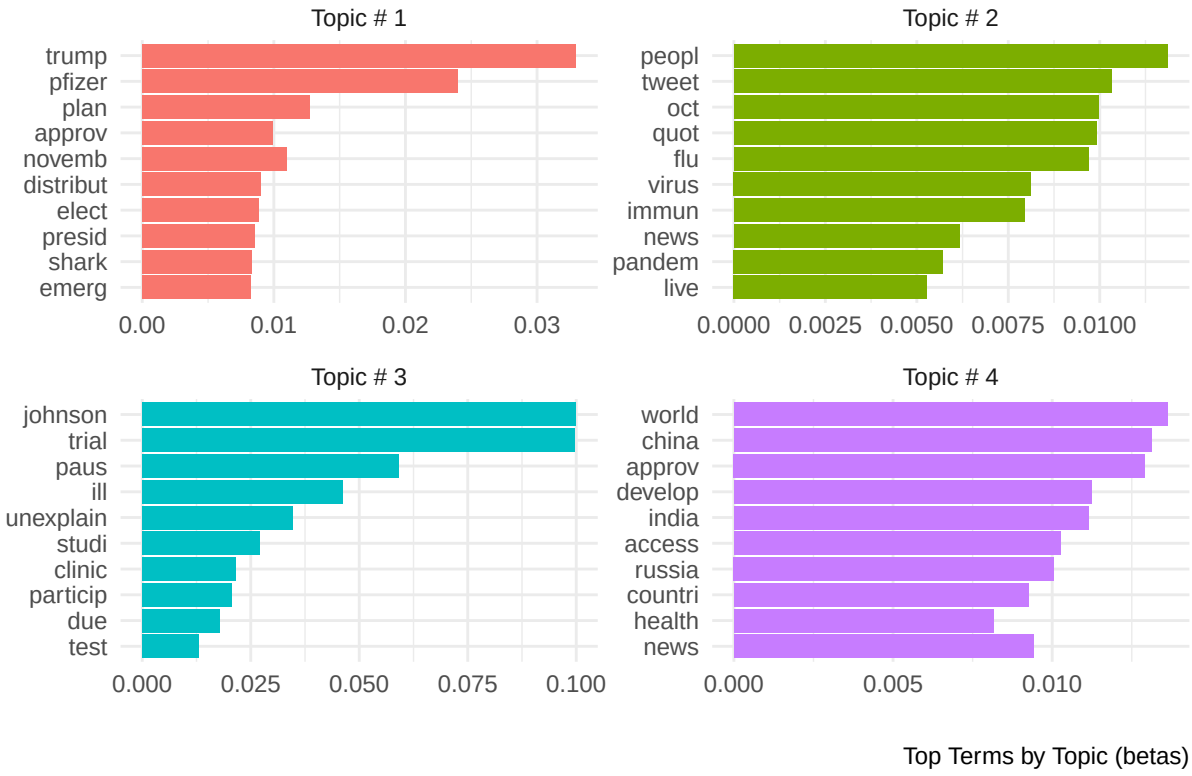



```
# Count the number of positive words for each day over the period of which the data
## covered
tidy_tweet_sentiment %>% filter(sentiment == 'positive') %>%
  ggplot(aes(x = Timestamp, y = n)) +
  geom_bar(stat = 'identity', color = 'green', width = 0.7) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 10)) +
  theme(plot.title = element_text(hjust = 0.5, size = 18)) +
  ylab('Number of words') +
  xlab('Date') +
  ggtitle('Count of positive Sentiment Words') +
  theme(aspect.ratio = 1/4)
```



Topic Model

```
# After several tries, I found that, when I set the number of topics to four, the
## words that falls under each made the most sense. Therefore, I am showing the top
## ten words ranked by the matrics 'beta' that falls under each topic.
tweet_topic_model = LDA(tidy_tweetDTM, k = 4, control = list(seed = 123))
tweet_topics = tidy(tweet_topic_model, matrix = 'beta')
top_terms = tweet_topics %>% group_by(topic) %>% top_n(10, beta) %>% ungroup() %>%
  arrange (topic, -beta)

top_terms %>% mutate(term = reorder(term, beta)) %>%
  mutate(topic = paste('Topic #', topic)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~topic, scales = 'free') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 18)) +
  labs(title = 'Topic Model of Tweet Words', caption = 'Top Terms by Topic (betas)') +
  ylab('') +
  xlab('') +
  coord_flip()
```


Topic Model of Tweet Words
Topic # 1
trump
pfizer
plan
approv
novemb
distribut
elect
presid
shark
emerg
0.00
0.01
0.02
0.03
Topic # 2
peopl
tweet
oct
quot
flu
virus
immun
news
pandem
live
0.0000
0.0025
0.0050
0.0075
0.0100
Topic # 3
johnson
trial
paus
ill
unexplain
studi
clinic
particip
due
test
0.000
0.025
0.050
0.075
0.100
Topic # 4
world
china
approv
develop
india
access
russia
countri
health
news
0.000
0.005
0.010
Top Terms by Topic (betas)