Coronavirus tweets sentiment analysis in R

Marina Romanova, OJNTSV

Task

Prepare a text mining model in R for one of the following tasks. Perform the preprocessing steps including at least tokenization, stop word elimination, word and bi-grams investigation (frequencies), word/bi-gram cloud. Use the results for topic mapping and/or sentiment analysis. Explain the results.

Data Preprocessing

Displaying data

I chose to analyse data scrapped from Twitter. The tweets further analysed have one thematic: Covid-19. There are the following columns in the dataset:

```
#Loading dataset
tweets_data <- read_delim("Corona_virus_tweets.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE)
```

```
## Rows: 41157 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ";"
## chr (3): Location, TweetAt, OriginalTweet
## dbl (2): UserName, ScreenName
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#displaying dataset
head(tweets_data)
```

```
## # A tibble: 6 × 5
##   UserName ScreenName Location                TweetAt    OriginalTweet
##      <dbl>      <dbl> <chr>                   <chr>      <chr>
## 1     3799      48751 London                  16-03-2020 "@MeNyrbie @Phil_Gah…
```

```
## 2     3800      48752 UK                          16-03-2020 "advice Talk to your…
## 3     3801      48753 Vagabonds                   16-03-2020 "Coronavirus Austral…
## 4     3802      48754 <NA>                        16-03-2020 "My food stock is no…
## 5     3803      48755 <NA>                        16-03-2020 "Me, ready to go at …
## 6     3804      48756 ÅšT: 36.319708,-82.363649 16-03-2020 "As news of the regi…
```

There are 5 columns and 41 157 rows in the original data. For further analysis only those columns will be taken into account:

- TweetAt
- OriginalTweet

```
#subsetting data
tweets_data <- select(tweets_data, TweetAt, OriginalTweet)
```

Which time period do we have for analysis?

```
#formatting date column to be date
tweets_data$TweetAt <- as.Date(tweets_data$TweetAt,format='%d-%m-%Y')
a <- table(tweets_data$TweetAt)
a
```

```
##
## 2020-03-16 2020-03-17 2020-03-18 2020-03-19 2020-03-20 2020-03-21 2020-03-22
##        656       1977       2742       3215       3448       2653       2114
## 2020-03-23 2020-03-24 2020-03-25 2020-03-26 2020-03-27 2020-03-28 2020-03-29
##       2062       1480       2979       1277        345         23        125
## 2020-03-30 2020-03-31 2020-04-01 2020-04-02 2020-04-03 2020-04-04 2020-04-05
##         87        316        630        954        810        767       1131
## 2020-04-06 2020-04-07 2020-04-08 2020-04-09 2020-04-10 2020-04-11 2020-04-12
##       1742       1843       1881       1471       1005        909        803
## 2020-04-13 2020-04-14
##       1428        284
```

```
print(paste("Earliest tweet:", min(tweets_data$TweetAt)))
```

```
## [1] "Earliest tweet: 2020-03-16"
```

```
print(paste("Latest tweet:", max(tweets_data$TweetAt)))
```

```
## [1] "Latest tweet: 2020-04-14"
```

As it can be seen, there are tweets from the first month of pandemic in the dataset.

Tokenization and text cleaning

For further analysis we need to tokenize and clean data Firstly, let's analyse data by unigram - so the unit for analysis will be one word.

We also need to clean the data. I excluded stopwords, digits, punctuation and 4 specific text combination that did not matter, but were often found in the text.

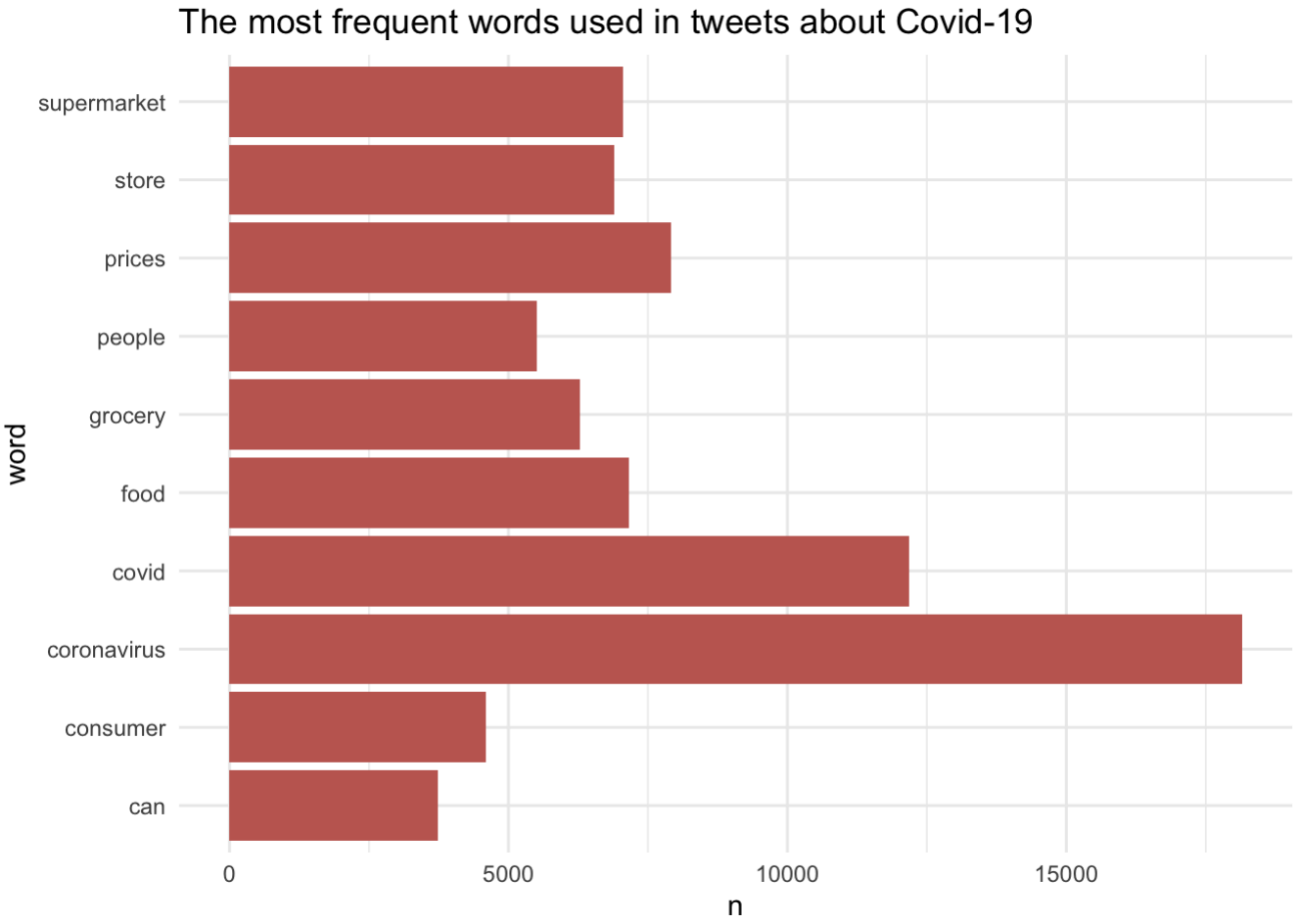
```
#making data to be in lower case
tweets_data$clean_lem = str_to_lower(tweets_data$OriginalTweet)
#tokenizing and cleaning text
tweets_token = tweets_data %>%
  unnest_tokens(word, clean_lem) %>%
  filter(!(word %in% stopwords("en"))) %>%
  filter(!(str_detect(word, "[[:digit:]]"))) %>%
  filter(!(str_detect(word, "[[:punct:]]"))) %>%
  filter(!(str_detect(word, "rt"))) %>%
  filter(!(str_detect(word, "@\\w+"))) %>%
  filter(!(str_detect(word, "https"))) %>%
  filter(!(str_detect(word, "amp")))
#displaying data
head(tweets_token)
```

```
## # A tibble: 6 × 3
##   TweetAt    OriginalTweet                                                word
##   <date>     <chr>                                                        <chr>
## 1 2020-03-16 @MeNyrbie @Phil_Gahan @Chrisitv https://t.co/iFz9FAn2Pa and … meny…
## 2 2020-03-16 @MeNyrbie @Phil_Gahan @Chrisitv https://t.co/iFz9FAn2Pa and … chri…
## 3 2020-03-16 advice Talk to your neighbours family to exchange phone numb… advi…
## 4 2020-03-16 advice Talk to your neighbours family to exchange phone numb… talk
## 5 2020-03-16 advice Talk to your neighbours family to exchange phone numb… neig…
## 6 2020-03-16 advice Talk to your neighbours family to exchange phone numb… fami…
```

Word frequency

What are the most common words (unigrams) used in Tweets?

```
tweets_token %>% count(word, sort = TRUE) %>%
  slice(1:10) %>%
  ggplot() + geom_bar(aes(word, n), stat = "identity", fill = "#c46960") +
  theme_minimal() +
  labs(title = "The most frequent words used in tweets about Covid-19") +
  coord_flip()
```



Before taking sentiment into account, let's take a look at the top-10 most frequent words in tweets. As it can be seen, the most frequently used word was, expectedly, coronavirus itself. It was used more then 15 000 times. Its synonym - covid - is at the second place. Another topic in the most frequent words is food. No wonder it is like that as it was a panic in the first month of the pandemic that the fod supply can be over.

Wordcloud

Here are 100 most frequent words from tweets displayed as wordcloud.

```
set.seed(1234) # for reproducibility
tweets_token %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

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

Bigram

Now let's take a look at bigrams. The data cleaning steps for tokenisation are the same as for unigrams.

```
#cleaning and tokening data
bigram_df <- tweets_data %>%
```

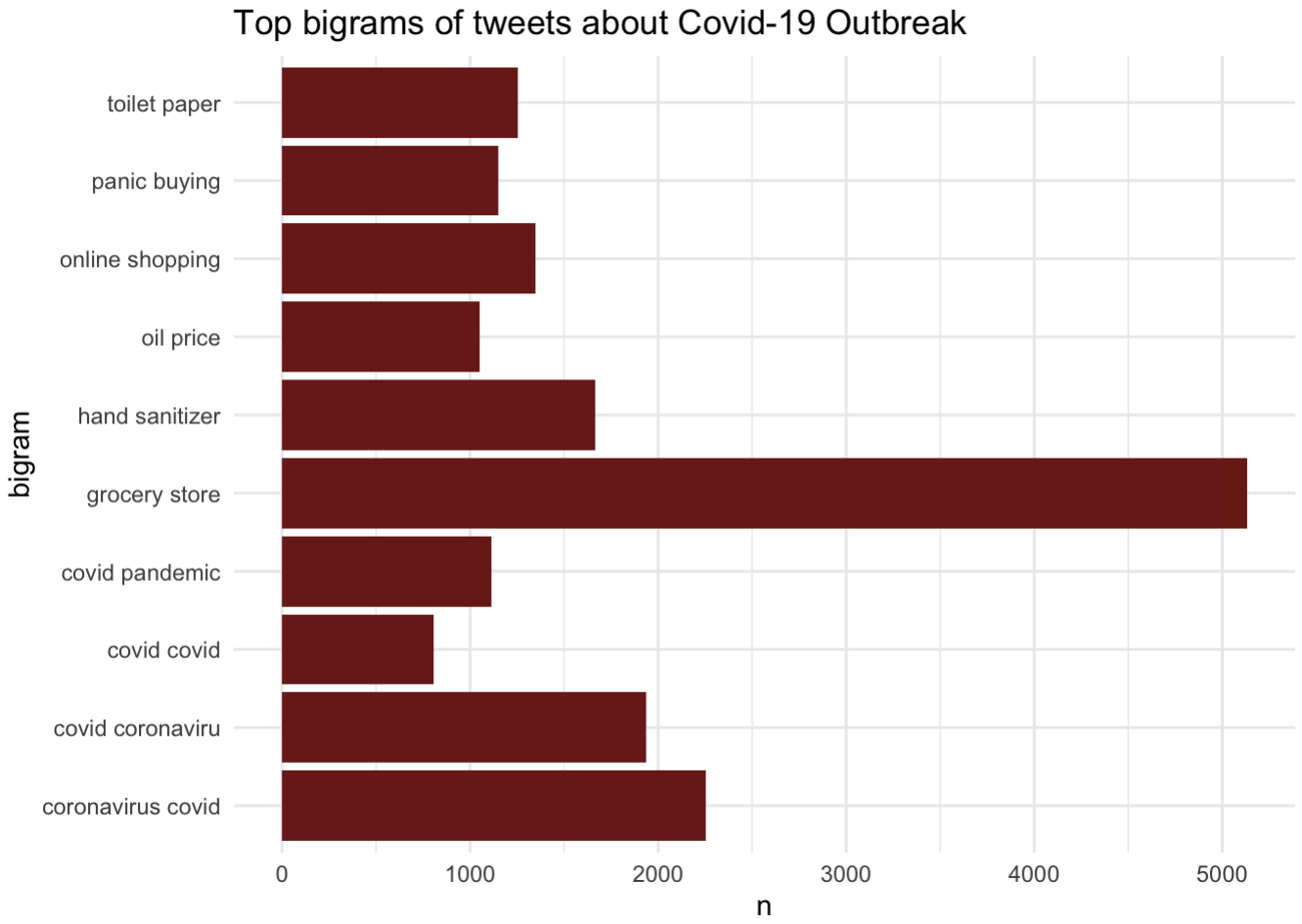
```
         mutate(OriginalTweet = str_replace_all(string = OriginalTweet ,
                                    pattern = "[[:punct:]]",
                                    " ")) %>%
         mutate(OriginalTweet = str_replace_all(string = OriginalTweet ,
                                    pattern = "[[:digit:]]",
                                    "")) %>%
        mutate(OriginalTweet = str_replace_all(string = OriginalTweet ,
                                    pattern = "https",
                                    "")) %>%
        mutate(OriginalTweet = str_replace_all(string = OriginalTweet ,
                                    pattern = "rt",
                                    "")) %>%
        mutate(OriginalTweet = str_replace_all(string = OriginalTweet ,
                                    pattern = "@\\w+",
                                    "")) %>%
        mutate(OriginalTweet = str_replace_all(string = OriginalTweet ,
                                    pattern = "amp",
                                    "")) %>%
        unnest_tokens(output = bigram,
                      input = OriginalTweet ,
                      token = "ngrams",
                      n = 2)
#creating bigrams
biwords_df <- bigram_df %>%
        separate(bigram, c("word1","word2"), sep= " ") %>%
        filter(!word1 %in% stop_words$word & !word2 %in% stop_words$word) %>%
        mutate(word2 = str_replace_all(string = word2 , pattern = "s$", "")) %>%
        unite(bigram, word1, word2 , sep = " ")

head(biwords_df)
```

```
## # A tibble: 6 × 3
##   TweetAt    clean_lem                                                bigram
##   <date>     <chr>                                                    <chr>
## 1 2020-03-16 @menyrbie @phil_gahan @chrisitv https://t.co/ifz9fan2pa and… menyr…
## 2 2020-03-16 @menyrbie @phil_gahan @chrisitv https://t.co/ifz9fan2pa and… phil …
## 3 2020-03-16 @menyrbie @phil_gahan @chrisitv https://t.co/ifz9fan2pa and… gahan…
## 4 2020-03-16 advice talk to your neighbours family to exchange phone num… advic…
## 5 2020-03-16 advice talk to your neighbours family to exchange phone num… neigh…
## 6 2020-03-16 advice talk to your neighbours family to exchange phone num… excha…
```

What are the most common bigrams used?

```
biwords_df %>% count(bigram, sort = TRUE) %>%
  slice(1:10) %>%
  ggplot() + geom_bar(aes(bigram, n), stat = "identity", fill = "#7B241C") +
  theme_minimal() +
  labs(title = "Top bigrams of tweets about Covid-19 Outbreak") +
  coord_flip()
```



Most frequent bigrams have the same topics as the unigrams: coronavirus and grocery panic.

Sentiment analysis

For sentiment analysis I will use NRC sentiment vocabulary. The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). For further analysis,

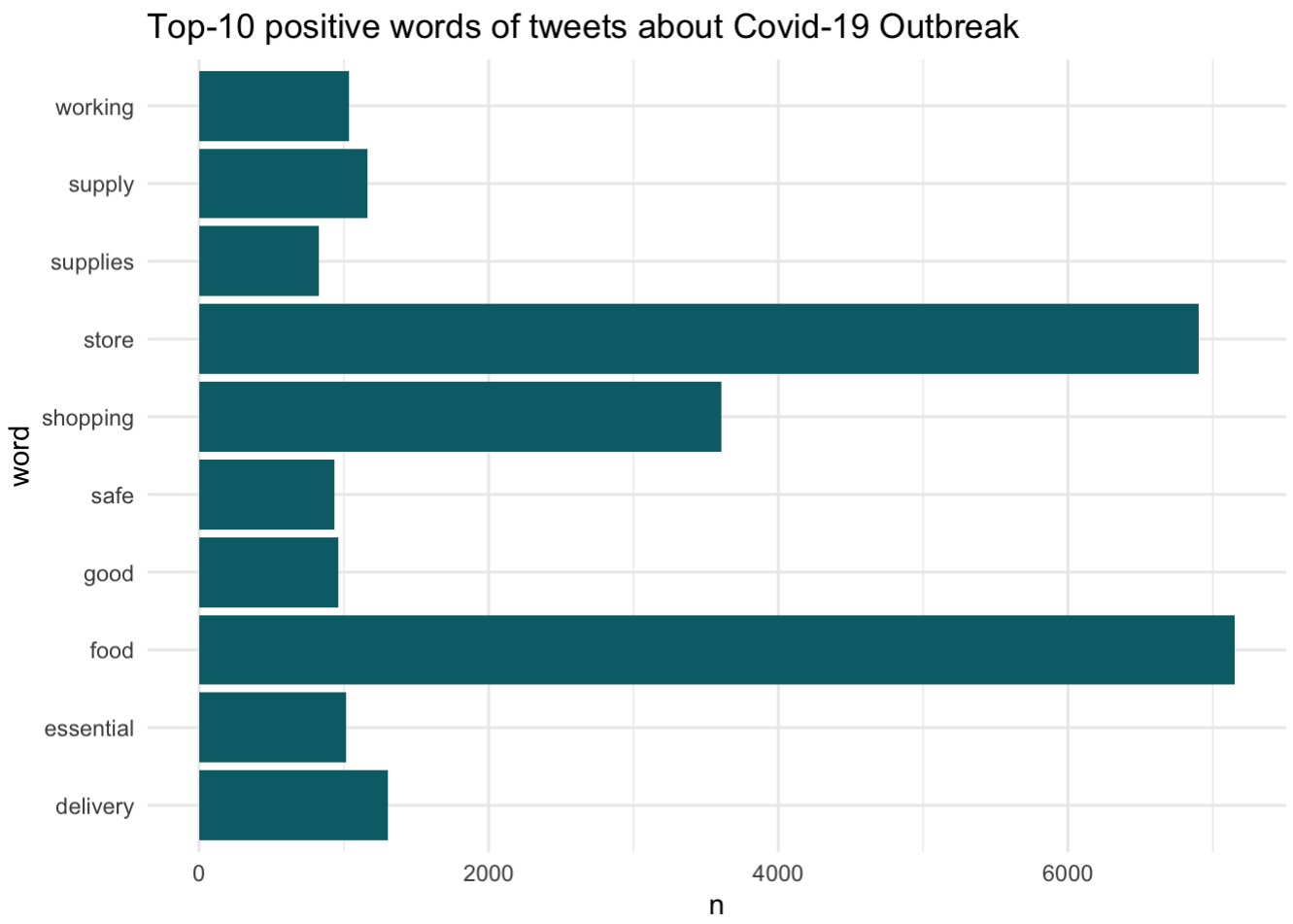
only sentiment will be taken into account.

But before analysing sentiment, let's take a look at most used positive and negative words

Most frequent used positive words

```
#counting positive sentiment
nrc_positive <- get_sentiments("nrc") %>%
  filter(sentiment == "positive")
#visualizing
tweets_token %>%
  inner_join(nrc_positive) %>%
  dplyr::count(word, sort = TRUE) %>%
  slice(1:10) %>%
  ggplot() + geom_bar(aes(word, n), stat = "identity", fill = "#006d77") +
  theme_minimal() +
  coord_flip() +
  labs(title = "Top-10 positive words of tweets about Covid-19 Outbreak")
```

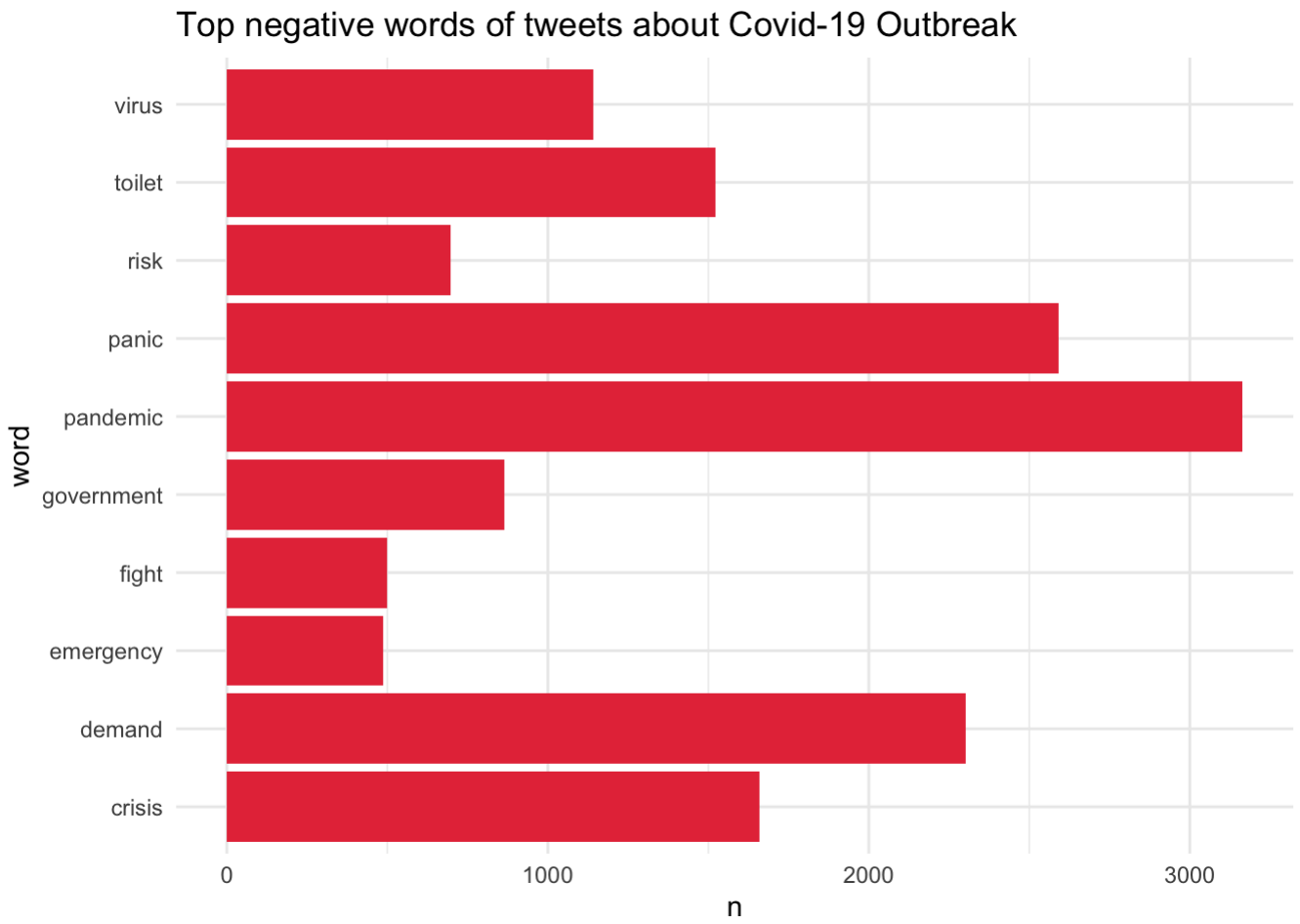
```
## Joining, by = "word"
```



Most frequent used negative words

```
#counting negative sentiment
nrc_negative <- get_sentiments("nrc") %>%
  filter(sentiment == "negative")
#visualizing sentiment
tweets_token %>%
  inner_join(nrc_negative) %>%
  dplyr::count(word, sort = TRUE) %>%
  slice(1:10) %>%
  ggplot() + geom_bar(aes(word, n), stat = "identity", fill = "#e63946") +
  theme_minimal() +
  coord_flip() +
  labs(title = "Top negative words of tweets about Covid-19 Outbreak")
```

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



There is a huge difference in topics stated between negative and positive words! The positive words are mostly about food (which is still not the most positive thing to talk about, especially for a whole month). However, the negative words are rather very upsetting: panic, crisis, emergency, panic - those are very negative words.

Calculating sentiment for each day

I am going to use very simple method of calculation the sentiment. I will count overall of positive and negative words for each date, and the sentiment here is the amount difference between the two.

```
# grouping words by date
tidy_tweets <- tweets_token %>%
    group_by(TweetAt)
# counting sentiment
```

```
sentiment_bydate <- tidy_tweets %>%
  inner_join(get_sentiments("bing")) %>%
  count(TweetAt, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

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

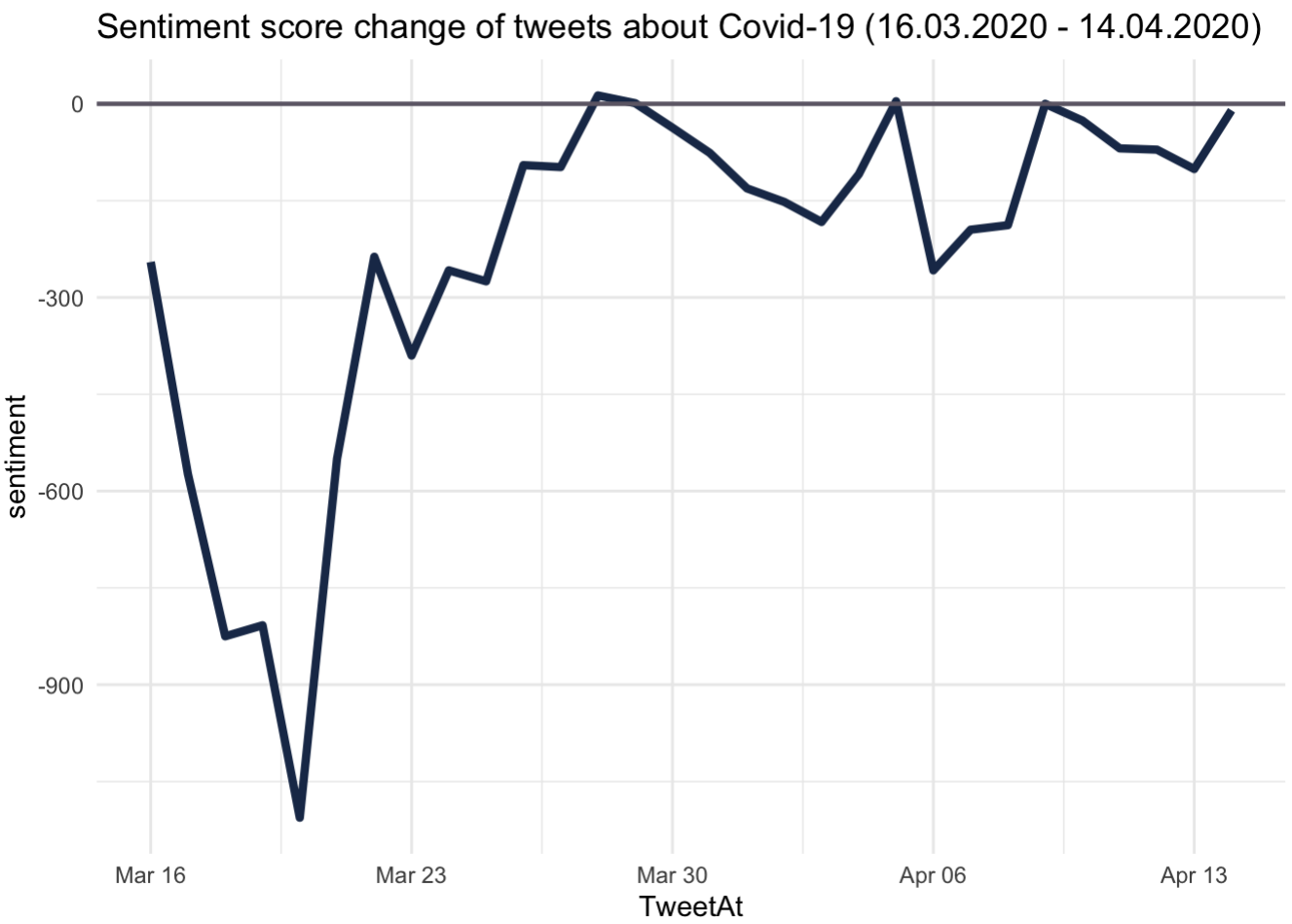
```
head(sentiment_bydate)
```

```
## # A tibble: 6 × 4
## # Groups:   TweetAt [6]
##   TweetAt    negative positive sentiment
##   <date>        <int>    <int>     <int>
## 1 2020-03-16      766      521      -245
## 2 2020-03-17     2092     1518      -574
## 3 2020-03-18     2931     2106      -825
## 4 2020-03-19     3301     2493      -808
## 5 2020-03-20     3836     2730     -1106
## 6 2020-03-21     2596     2046      -550
```

How the sentiment changed over time?

```
d <- ggplot(sentiment_bydate, aes(TweetAt, sentiment)) +
  geom_line(show.legend = TRUE, color = "#1d3557", linewidth = 1.5) +
  theme_minimal() +
  labs(title = "Sentiment score change of tweets about Covid-19 (16.03.2020 - 14.04.2020)")

d + geom_hline(yintercept=0, color = "#6d6875", size=0.8)
```



The result is upsetting but expected: almost in every single day of the first moth of Covid-19 outbreak the overall sentiment was negative, with the biggest fall in the first week of the pandemic. There were only 5 days where sentiment was overall positive, but the score was almost 0, so it was rather neutral.

Conclusion

By word and sentiment analysis of tweets for the first moth of covid-19 pandemic, it can be concluded that the topics were most;y discussed were coronovarius itself and assumptions about the food crisis. The overall sentiment score was negative in almost every day of the first month, having the biggest fall int the week 1 of the pandemic.

References

- https://www.tidytextmining.com/sentiment.html
- https://www.r-bloggers.com/2021/05/sentiment-analysis-in-r-3/

- https://rpubs.com/Shaahin/anxiety-bigram