Assignment 3 - Sentiment Analysis II

Marie Rivers

4/20/2022

```
library(quanteda)
library(quanteda.sentiment)
library(quanteda.textstats)
library(tidyverse)
library(tidytext)
library(lubridate)
library(wordcloud) #visualization of common words in the data set
library(reshape2)
library(sentimentr)
library(kableExtra)
```

This assignment uses tweet data for the term 'IPCC'

```
raw_tweets <- read.csv("https://raw.githubusercontent.com/MaRo406/EDS_231-text-sentiment/main/dat/IPCC_

dat<- raw_tweets[,c(5,7)] # Extract Date and Title fields

tweets <- tibble(text = dat$Title,
                 id = seq(1:length(dat$Title)),
                 date = as.Date(dat$Date,'%m/%d/%y'))

#clean up the URLs from the tweets (people linking to news articles and such)
tweets$text <- gsub("http[^[:space:]]*", "",tweets$text) # substitute http links with nothing
tweets$text <- str_to_lower(tweets$text)
```

1. Think about how to further clean a twitter data set. Let's assume that the mentions of twitter accounts is not useful to us. Remove them from the text field of the tweets tibble.

```
tweets_clean <- tweets %>%
  mutate(text_clean = text)  # keeping a column of the original text as a check

tweets_clean$text_clean <- gsub("@[^[:space:]]*", "", tweets_clean$text_clean)
```

2. Compare the ten most common terms in the tweets per day. Do you notice anything interesting?

```
#tokenize tweets to individual words
words <- tweets_clean %>%
  select(id, date, text_clean) %>%
  unnest_tokens(output = word, input = text_clean, token = "words") %>%
  anti_join(stop_words, by = "word")
```

```
words_count <- words %>%
  count(date, word)

top_ten_per_day <- words_count %>%
  group_by(date) %>%
  top_n(10, n)
```

```
top_ten_table = aggregate(top_ten_per_day$word, list(top_ten_per_day$date), paste, collapse=", ") %>%
  rename(Date = Group.1) %>%
  rename(top_words = x) %>%
  kable(col.names = c("Date", "Top 10 Words")) %>%
  kable_paper(full_width = TRUE) %>%
  row_spec(c(0), background = "lightgray")
```

```
## Warning in latex_new_row_builder(target_row, table_info, bold, italic,
## monospace, : Setting full_width = TRUE will turn the table into a tabu
## environment where colors are not really easily configable with this package.
## Please consider turn off full_width.
```

```
top_ten_table
```

Date	Top 10 Words
2022-04-01	carbon, change, climate, climatereport, fossil, ipcc, monday, rapid, read, report, upcoming
2022-04-02	04, 2022, carbon, change, climate, emissions, gt, ipcc, monday, report, scenarios
2022-04-03	aitt, authors, climate, dasgupta, dipak, dr, fossil, hosted, ipcc, joyashree, lead, lifespaces, mahindra, mitigation, purushottam, reminder, report, roy, scientists, set, space, sunita, teri, twitter, unpack
2022-04-04	change, climate, emissions, fossil, ipcc, limit, report, scientists, warming, world
2022-04-05	action, change, climate, emissions, fossil, global, ipcc, report, warming, world
2022-04-06	change, climate, crisis, emissions, fossil, ipcc, listen, oil, report, scientists, world
2022-04-07	change, climate, climatechange, emissions, energy, global, ipcc, report, time, world
2022-04-08	action, carbon, change, climate, climatechange, emissions, global, ipcc, released, report, warming, world
2022-04-09	carbon, change, climate, emissions, fossil, fuels, global, ipcc, it's, oil, report, warming, world
2022-04-10	change, climate, emissions, fossil, fuel, global, ipcc, report, time, warming

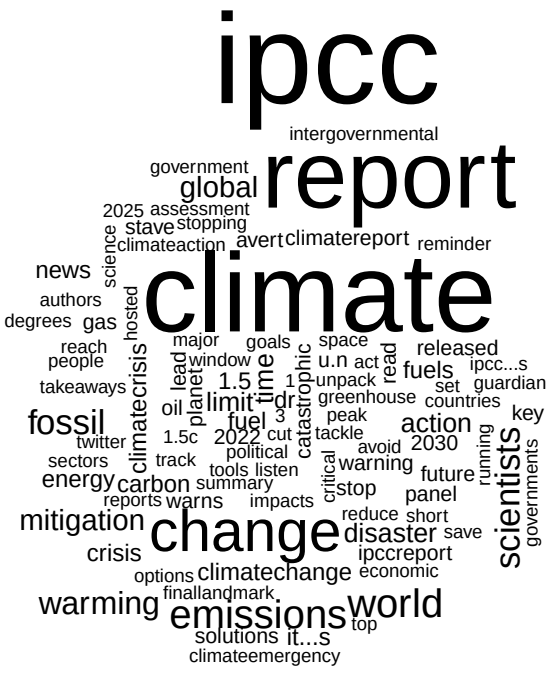
```
# xxx...based on the top 10 words there might be some more cleaning to do...should we remove duplicate
```

3. Adjust the wordcloud in the "wordcloud" chunk by coloring the positive and negative words so they are identifiable.

```
#load sentiment lexicon
bing_sent <- get_sentiments('bing')

words_sent <- words %>%
  left_join(bing_sent, by = "word") %>%
  left_join(
    tribble(
      ~sentiment, ~sent_score,
      "positive", 1,
      "negative", -1),
    by = "sentiment")

#xxx...color by sentiment
wordcloud_sent <- words_sent %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```



4. Let's say we are interested in the most prominent entities in the Twitter discussion. Which are the top 10 most tagged accounts in the data set. Hint: the "explore_hashtags" chunk is a good starting point.

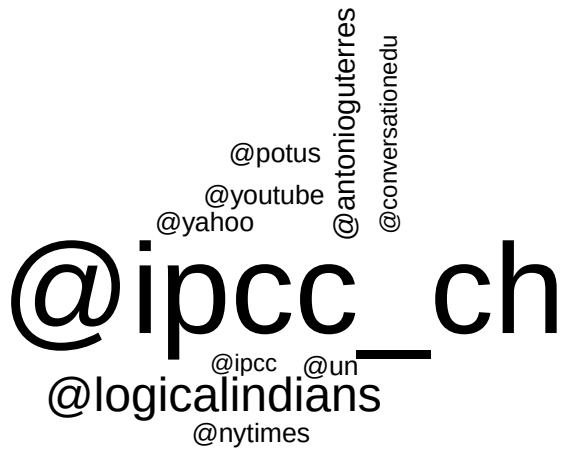
```
corpus <- corpus(dat$Title) #enter quanteda
# corpus is a collection of documents (ie tweets) with metadata
```

```
tagged_tweets <- tokens(corpus, remove_punct = TRUE) %>%
                 tokens_keep(pattern = "@*")
dfm_tagged<- dfm(tagged_tweets)

tstat_freq <- textstat_frequency(dfm_tagged, n = 10)

#tidytext gives us tools to convert to tidy from non-tidy formats
tagged_tib<- tidy(dfm_tagged)

tagged_tib %>%
   count(term) %>%
   with(wordcloud(term, n, max.words = 10))
```



5. The Twitter data download comes with a variable called "Sentiment" that must be calculated by Brandwatch. Use your own method to assign each tweet a polarity score (Positive, Negative, Neutral) and compare your classification to Brandwatch's (hint: you'll need to revisit the "raw_tweets" data frame).

```
dat2<- raw_tweets[,c(5, 7, 11)] # Extract Date, Title, and Sentiment fields

tweets2 <- tibble(text = dat2$Title,
                  element_id = seq(1:length(dat2$Title)),
```

```
                  date = as.Date(dat2$Date,'%m/%d/%y'),
                  sent_brandwatch = dat2$Sentiment)
```

```
sent_method2 <- sentiment_by(tweets2$text)

tweets2 <- inner_join(tweets2, sent_method2, by = "element_id") %>%
  mutate(sent_method2 = case_when(
    ave_sentiment < 0 ~ "negative",
    ave_sentiment > 0 ~ "positive",
    ave_sentiment == 0 ~ "neutral"))
```

xxx. . . think of way to visualize this