Week 4 Lab: Sentiment Analysis II

Joe DeCesaro

IPCC Report Twitter

```
library(quanteda)
#devtools::install_github("quanteda/quanteda.sentiment") #not available currently through CRAN
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(patchwork)
```

Load the data and sentiment lexicons

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

dat<- raw_tweets[,c(4,6, 10, 11)] # Extract Date and Title fields

#load sentiment lexicons
bing_sent <- get_sentiments('bing')
nrc_sent <- get_sentiments('nrc')
```

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 <- tibble(text = dat$Title,
                 id = seq(1:length(dat$Title)),
                 date = as.Date(dat$Date,
                                '%m/%d/%y'),
                 sentiment = dat$Sentiment,
                 emotion = dat$Emotion) %>%
  mutate(text = str_replace(string = text,
                            pattern = "http.*[:space:]",
                            replacement = ""),
         text = str_replace(string = text,
                            pattern = "http.*$",
                            replacement = ""),
         text = str_replace(string = text,
```

```
                            pattern = "@.*[:space:]",
                            replacement = ""),
         text = str_replace(string = text,
                            pattern = "@.*$",
                            replacement = ""),
         text = str_to_lower(text))

tweets$text <- iconv(tweets$text,
                     "latin1",
                     "ASCII",
                     sub="")
```

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

Make dataframe with each word in different row, remove all stop words, add in the `bing` sentiment, and a numerical sentiment score.

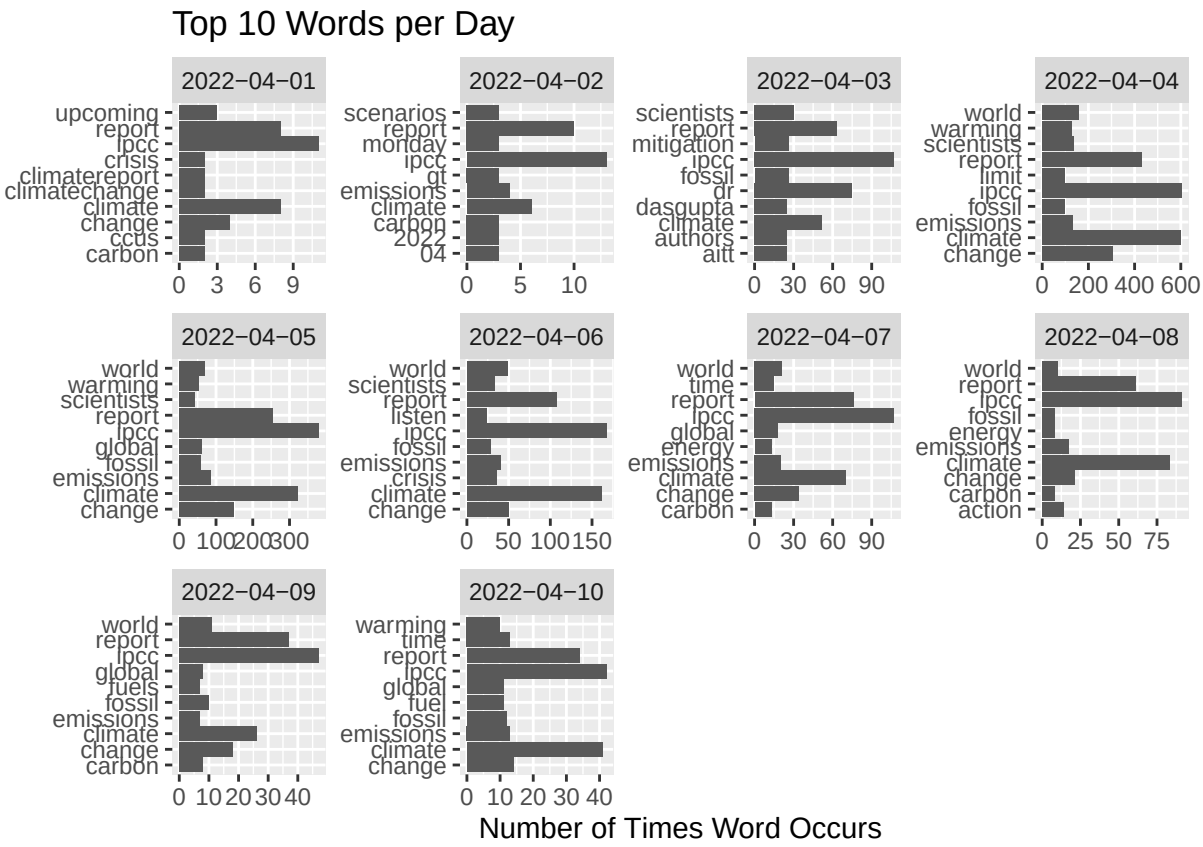
```
#tokenize tweets to individual words
words <- tweets %>%
  select(id, date, text) %>%
  unnest_tokens(output = word, input = text, token = "words") %>%
  anti_join(stop_words, by = "word") %>%
  left_join(bing_sent, by = "word") %>%
  left_join(
    tribble(
      ~sentiment, ~sent_score,
      "positive", 1,
      "negative", -1),
    by = "sentiment")
```

Make a slice of the top ten words of each day then make a plot.

```
ten_words <- words %>%
  group_by(date, word) %>%
  summarise(count = n()) %>%
  group_by(date) %>%
  slice_max(count,
            n = 10,
            with_ties = FALSE) %>%
  ggplot(aes(x = count,
             y = word)) +
  geom_col() +
  facet_wrap(~date, scales = "free") +
  guides(fill = "none") +
  labs(x = "Number of Times Word Occurs",
       y = "",
       title = "Top 10 Words per Day")
```

```
## `summarise()` has grouped output by 'date'. You can override using the `.groups` argument.
```

```
ten_words
```



The day the report is released and for a couple of days after there is a great increase in the number of mentions of IPCC because the counts are much higher for all words. IPCC, report, and climate is high throughout the dates but this is expected.

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

```
words %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("darkgreen", "red4"),
                   max.words = 100)
```

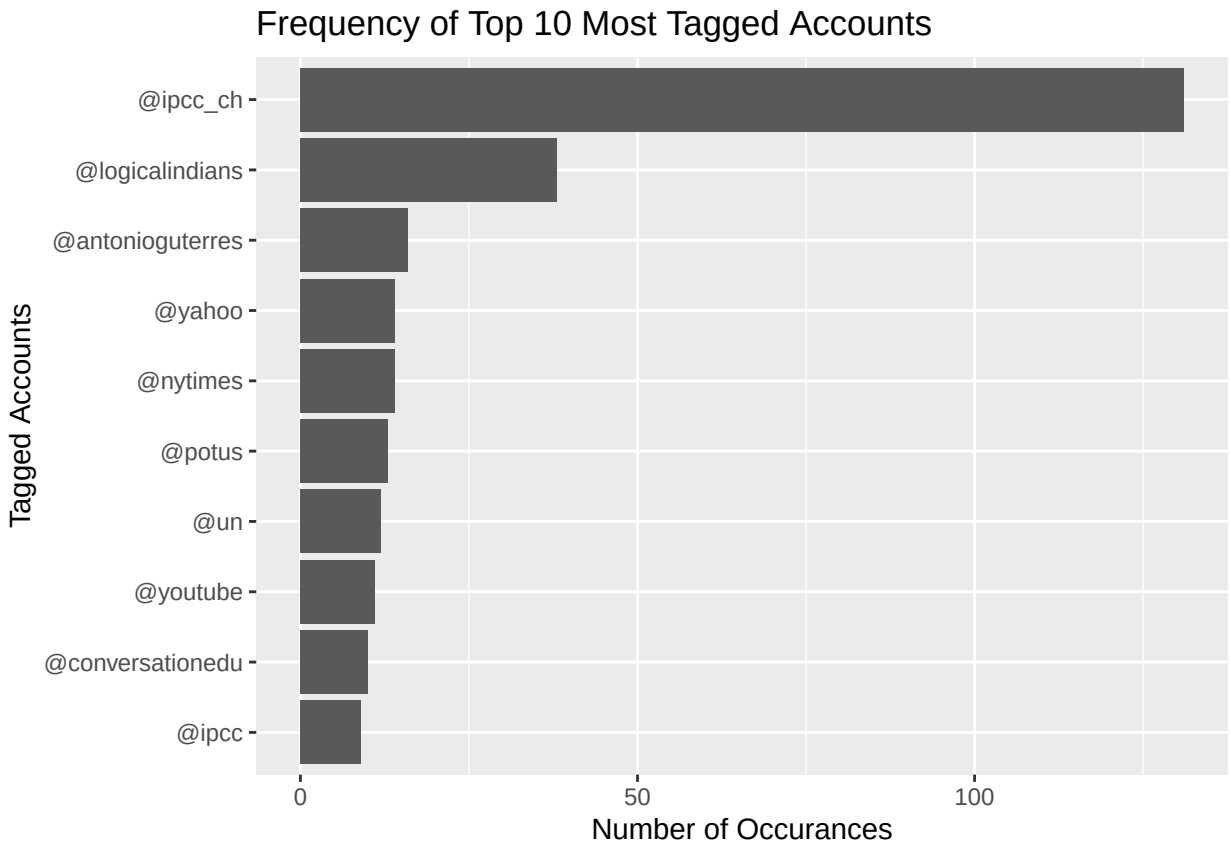
```
## Joining, by = c("word", "sentiment")
```

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.

```
corpus <- corpus(dat$Title) #enter quanteda

tag_tweets <- tokens(corpus, remove_punct = TRUE) %>%
  tokens_keep(pattern = "@*") %>%
  dfm() %>%
  textstat_frequency(n = 10) %>%
  ggplot(aes(x = frequency,
             y = reorder(feature, frequency))) +
  geom_col() +
  labs(x = "Number of Occurances",
       y = "Tagged Accounts",
       title = "Frequency of Top 10 Most Tagged Accounts")

tag_tweets
```



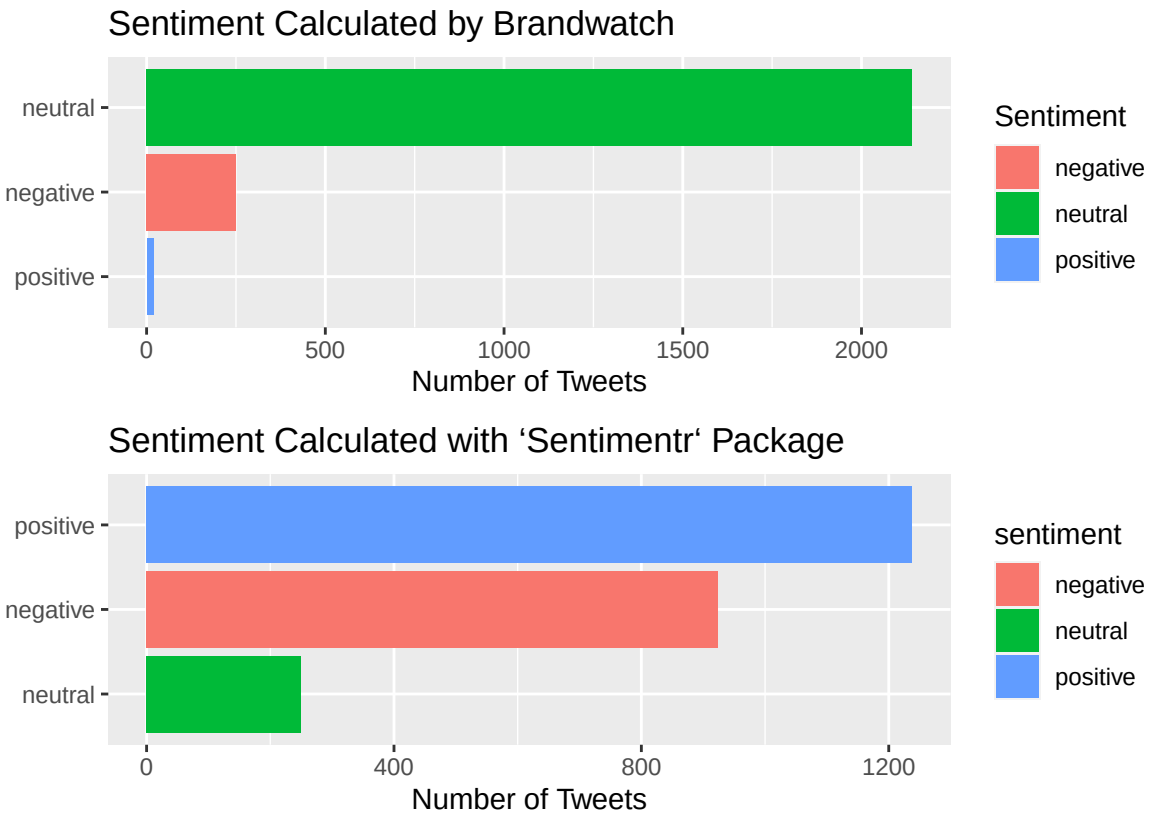
5. The Twitter data download comes with a variable called "Sentiment" that must be calculated by Brandwatch. Use your own method to assign each tweet a polarity score (Positive, Negative, Neutral) and compare your classification to Brandwatch's (hint: you'll need to revisit the "raw_tweets" data frame).

```
sentimentr_calc <- get_sentences(dat$Title) %>%
  sentiment() %>%
  group_by(element_id) %>%
  summarize(sentiment_score = mean(sentiment)) %>%
  mutate(sentiment = case_when(sentiment_score < 0 ~ "negative",
                               sentiment_score == 0 ~ "neutral",
                               sentiment_score > 0 ~ "positive")) %>%
  group_by(sentiment) %>%
  summarize(count = n())%>%
  ggplot(aes(x = count,
             y = reorder(sentiment, count),
             fill = sentiment)) +
  geom_col() +
  labs(x = "Number of Tweets",
       y = "",
       title = "Sentiment Calculated with `Sentimentr` Package")

brandwatch_calc <- dat %>%
  filter(Sentiment %in% c("positive", "negative", "neutral")) %>%
  group_by(Sentiment) %>%
```

```
  summarize(count = n()) %>%
  ggplot(aes(x = count,
             y = reorder(Sentiment, count),
             fill = Sentiment)) +
  geom_col() +
  labs(x = "Number of Tweets",
       y = "",
       title = "Sentiment Calculated by Brandwatch")


brandwatch_calc / sentimentr_calc
```



It appears that the brandwatch sentiment lexicon is much more likely for a word to be deemed neutral compared to the sentimentr package. I wonder if this has something to do with brands only wanting to be sure about sentiment analysis so they take a more conservative approach to deciding what words are positive, negative and neutral. Maybe the brandwatch sentiment lexicon is more conservative with labeling a word as positive or negative directly from Twitter because Twitter has so many languages participating.