EDS 231 Topic 4 Lab

Wylie Hampson

4/24/2022

In this assignment we are continuing to work with sentiment classification on text. Here we are working with tweets that had to do with IPCC.

Read in the twitter dataset from class.

```
twitter_data <- read.csv(here("data", "IPCC_tweets_April1-10_sample.csv"))

# Extract dates and titles.
dat <- twitter_data[,c(4,6)]
```

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.

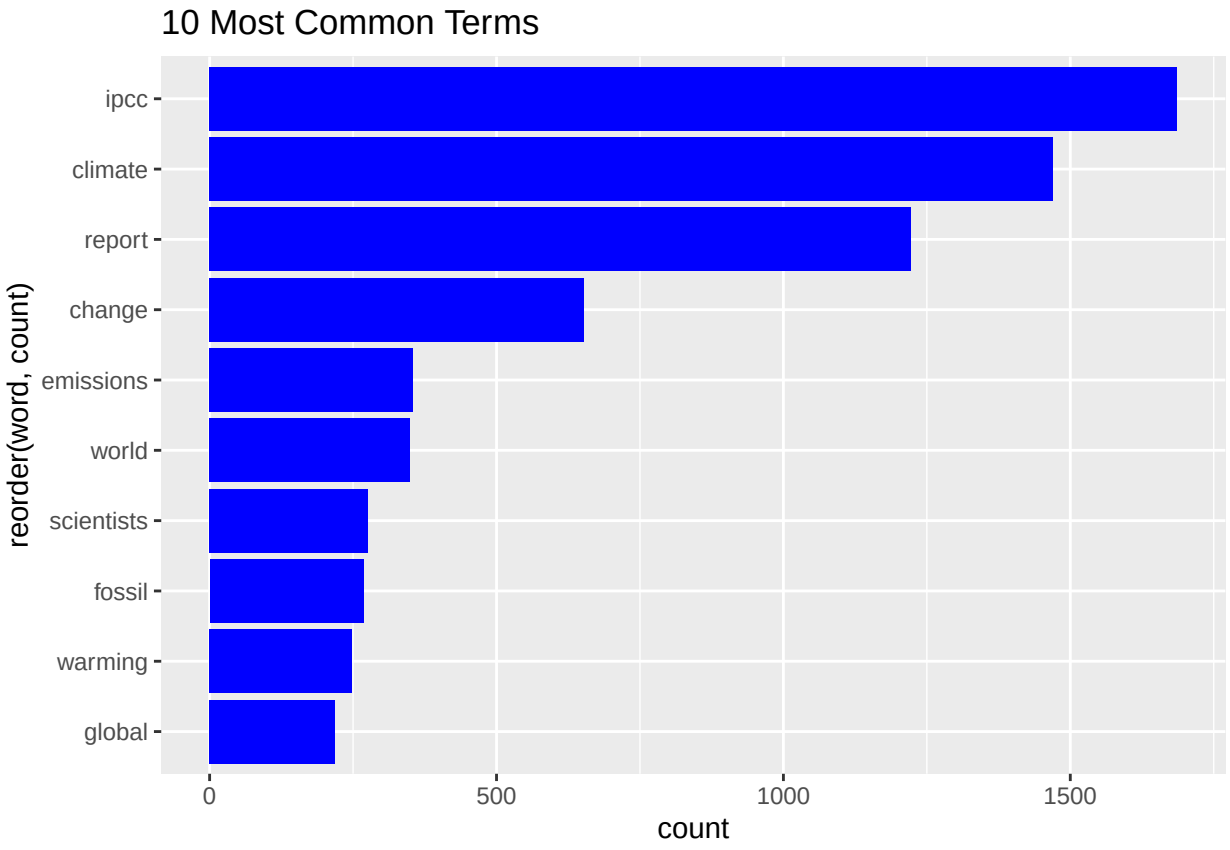
```
# Only keep columns of interest.
tweet_data <- tibble(id = seq(1:length(dat$Title)),
                     text = dat$Title,
                     date = as.Date(dat$Date,'%m/%d/%y'))

# Remove mentions from the text.
tweet_data$text <- gsub("@[^[:space:]]*", "", tweet_data$text)

# Remove emojis from the text.
tweet_data$text <- iconv(tweet_data$text, "latin1", "ASCII", sub="")

# Let's also clean up the HTMLs and convert text to lower case.
tweet_data$text <- gsub("http[^[:space:]]*", "", tweet_data$text)
tweet_data$text <- str_to_lower(tweet_data$text)
```

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

```
# Remove the stop words from the tweets.
tweet_words <- tweet_data %>%
  unnest_tokens(output = word, input = text, token = "words") %>%
  anti_join(stop_words, by = "word")

# Count the 10 most common words for each date.
word_count <- tweet_words %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  slice_max(count, n = 10, with_ties = FALSE)

word_count
```

```
## # A tibble: 10 x 2
##    word       count
##    <chr>      <int>
##  1 ipcc        1686
##  2 climate     1470
##  3 report      1223
##  4 change       653
##  5 emissions    355
##  6 world        349
##  7 scientists   276
##  8 fossil       268
##  9 warming      248
## 10 global       219
```

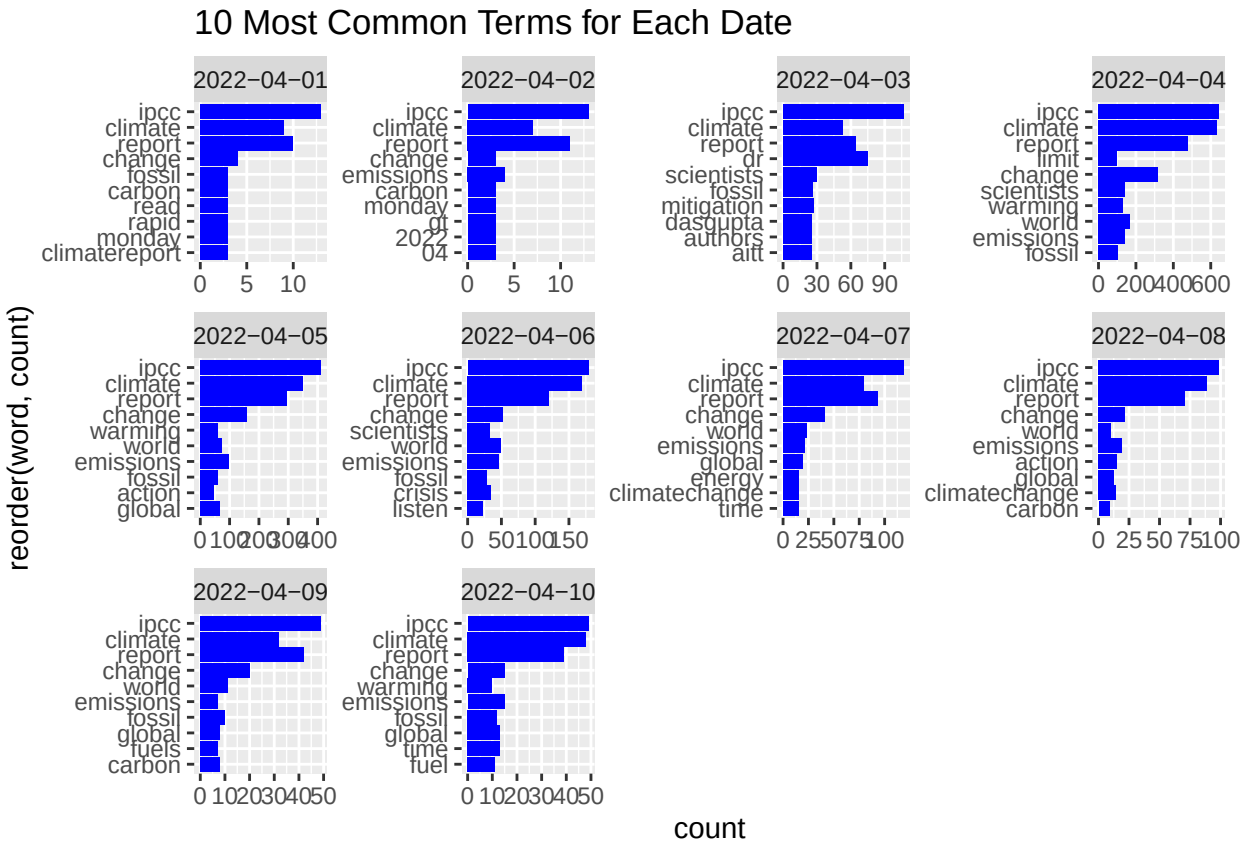
```
ggplot(data = word_count, aes(x=count, y=reorder(word, count))) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "10 Most Common Terms")
```




```
word_count_daily <- tweet_words %>%
  group_by(date, word) %>%
  summarise(count = n()) %>%
  slice_max(count, n = 10, with_ties = FALSE)
```

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

```
ggplot(data = word_count_daily, aes(x = count, y = reorder(word, count))) +
  geom_bar(stat = "identity", fill = "blue") +
  facet_wrap(~ date, scale = "free") +
  labs(title = "10 Most Common Terms for Each Date")
```



Some interesting things to note about the above charts. The top three most common terms both overall and for each day are always "ipcc", "climate", and "report". This makes sense because the tweets are about the ipcc climate report. You will also notice the count for these words is much higher on April 4th and 5th, which also makes sense becasue the report was approved on the 4th so obviously there will be more mentions then. Other than that, the top ten terms are all fairly similar for each of the ten days that we are looking at and they are all related to climate change.

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

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

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

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.

```
# Let's start with the raw data again and clean it up, but this time leave the mentions in it.
# We will use the quanteda package to do this.

# Convert the data into class corpus so that it can be tokenized using the tokens() function.
corpus <- corpus(dat$Title)
#summary(corpus)

# Tokenize the text so each tweet is a list of tokens
tokens <- tokens(corpus)

#clean it up
tokens <- tokens(tokens,
                 remove_punct = TRUE,
                 remove_numbers = TRUE)

# Remove stop words
tokens <- tokens_select(tokens, stopwords('english'), selection = 'remove')

# Make tokens lowercase.
tokens <- tokens_tolower(tokens)

# Now we actually remove all of the tokens that aren't mentions (don't start with "@")
```

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

dfm_mentions <- dfm(mention_tweets)

tstat_freq <- textstat_frequency(dfm_mentions, n = 100)
top_ten_mentions <- head(tstat_freq, 10)

top_ten_mentions
```

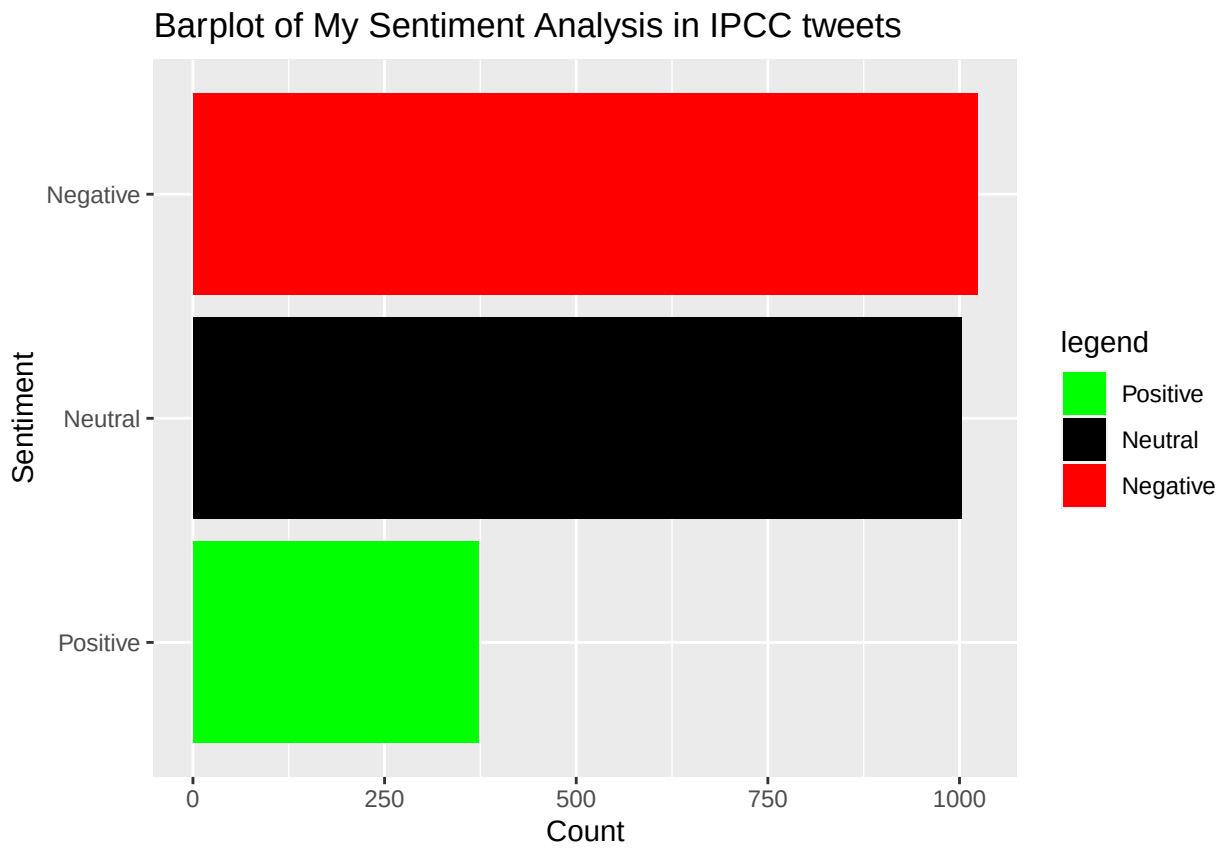
```
##             feature frequency rank docfreq group
## 1          @ipcc_ch       131    1     131   all
## 2   @logicalindians        38    2      38   all
## 3  @antonioguterres        16    3      16   all
## 4          @nytimes        14    4      14   all
## 5            @yahoo        14    4      14   all
## 6            @potus        13    6      13   all
## 7               @un        12    7      12   all
## 8          @youtube        11    8      11   all
## 9  @conversationedu        10    9      10   all
## 10           @ipcc         9   10       9   all
```

And here are the top ten mentions.

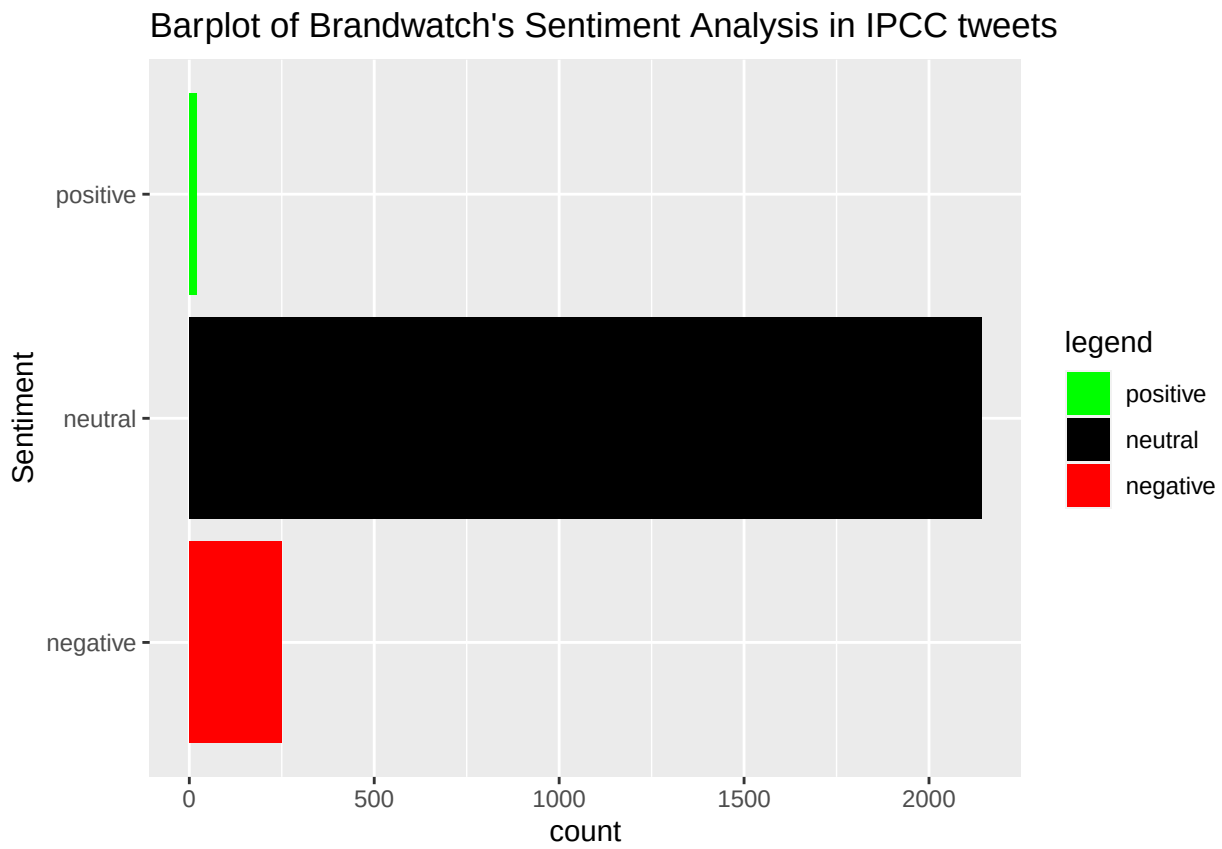
The Twitter data download comes with a variable called "Sentiment" that must be calculated by Brandwatch. Use your own method to assign each tweet a polarity score (Positive, Negative, Neutral) and compare your classification to Brandwatch's (hint: you'll need to revisit the "raw_tweets" data frame).

```
# Remember, `twitter_data` is the raw dataset and `tweet_data` is the cleaned up dataframe from above.

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

# `tweet_words` is the individual words from `tweet_data` with stop words removed from up above.
# Now I'm adding sentiment scores to the words.
tweet_words <- tweet_words %>%
  left_join(bing_sent, by = "word") %>%
  left_join(tribble(~ sentiment, ~ sent_score,
                    "positive", 1,
                    "negative",-1),
            by = "sentiment")

# Replace NA values with a score of 0 and a sentiment of neutral.
tweet_words$sent_score <- replace_na(tweet_words$sent_score, 0)
tweet_words$sentiment <- replace_na(tweet_words$sentiment, "neutral")

# Now give each tweet a sentiment score.
#take average sentiment score by tweet
tweet_sent <- tweet_data %>%
  left_join(tweet_words %>%
              group_by(id) %>%
```

```
              summarize(sent_score = mean(sent_score, na.rm = T)),
            by = "id")

# Count how many tweets of each sentiment value there is.
neutral <- length(which(tweet_sent$sent_score == 0))
positive <- length(which(tweet_sent$sent_score > 0))
negative <- length(which(tweet_sent$sent_score < 0))

Sentiment <- c("Positive","Neutral","Negative")

Count <- c(positive,neutral,negative)

# Create a dataframe showing sentiment values and their counts
output <- data.frame(Sentiment,Count)

output$Sentiment<-factor(output$Sentiment,levels=Sentiment)

# Here is a histogram of the tweets using my sentiment classification.
my_sentiment <- ggplot(output, aes(x=Sentiment,y=Count)) +
  geom_bar(stat = "identity", aes(fill = Sentiment)) +
  coord_flip() +
  scale_fill_manual("legend", values = c("Positive" = "green",
                                         "Neutral" = "black",
                                         "Negative" = "red")) +
  ggtitle("Barplot of My Sentiment Analysis in IPCC tweets")

# This pulls in the tweets with the Brandwatch sentiment classification.
brandwatch_sentiment <- twitter_data[,c(4, 6, 10)] %>%
  group_by(Sentiment) %>%
  summarise(count = n())

# Here is a histogram of the tweets with the Brandwatch sentiment classificaiton.
brandwatch_plot <- ggplot(brandwatch_sentiment, aes(x = Sentiment, y = count)) +
  geom_bar(stat = "identity", aes(fill = Sentiment))+
  coord_flip() +
  scale_fill_manual("legend", values = c("positive" = "green", "neutral" = "black", "negative" = "red"))
  ggtitle("Barplot of Brandwatch's Sentiment Analysis in IPCC tweets")

# Both charts side by side.
my_sentiment
```




```
brandwatch_plot
```



From the two plots above, the first one is from my sentiment classification, and the second one is from Brandwatch's sentiment classification. As you can see they do differ a bit. It seems as though most of the tweets that my classification classified as negative were instead classified as neutral by Brandwatch. Both classifications showed that positive tweets were the least common, however my classification had over 200 more tweets ranked as positive over the 19 tweets that Brandwatch classified as positive. I don't think this necessarily means that one classification is right or wrong, I think it just shows that sentiment can be classified in different ways.