Twitter Sentiment

Shale

4/20/2022

Loading & Further Cleaning

```
raw_tweets <- read_csv(here("data/IPCC_tweets_April1-10_sample.csv"))
```

```
## New names:
## Rows: 2411 Columns: 84
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (33): Query Name, Date, Title, Snippet, Url, Domain, Sentiment, Emotion... dbl
## (23): ...1, Query Id, Facebook Comments, Facebook Likes, Facebook Share... lgl
## (27): Assignment, Category Details, Checked, Display URLs, Facebook Aut... time
## (1): Time
## 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.
## * `` -> `...1`
```

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

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

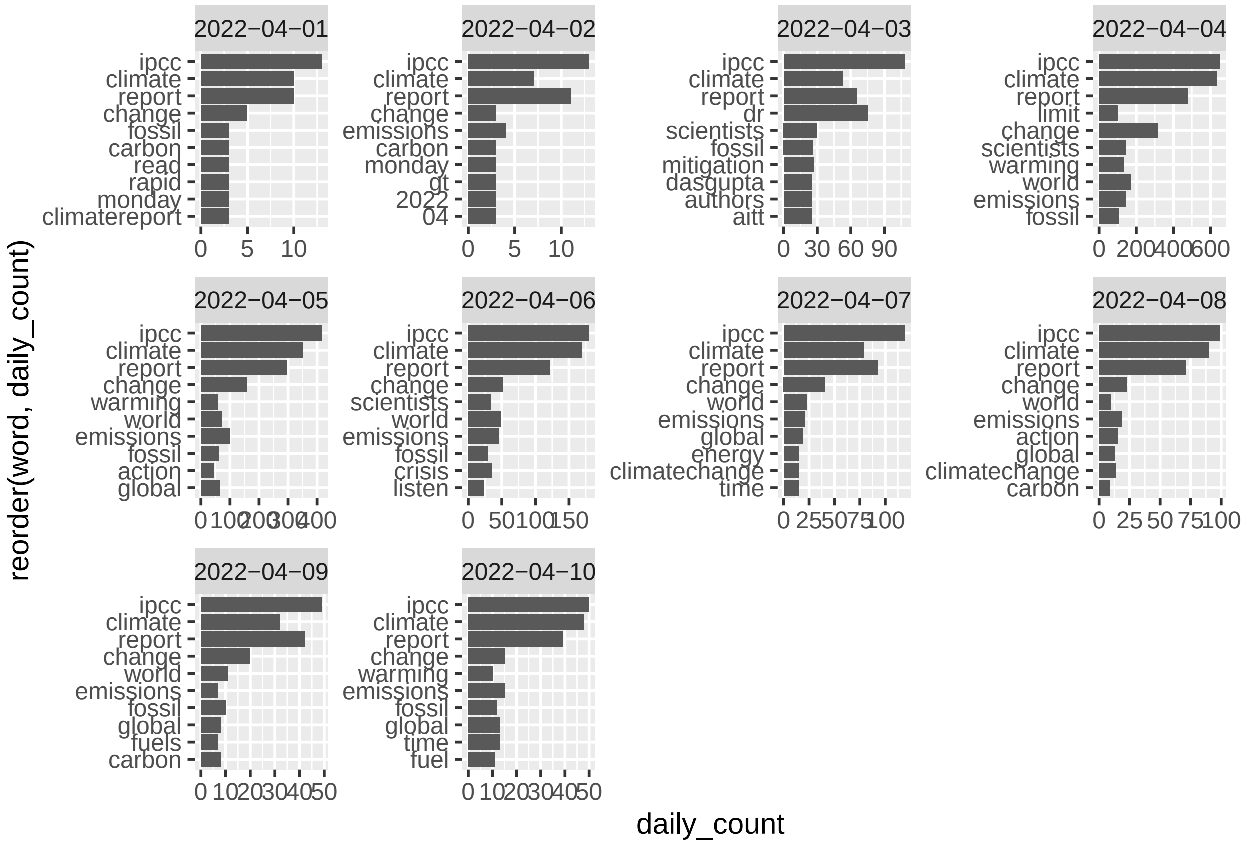
```
# clean up the URLs and tagged accounts from the tweets
clean_twt = tweets

clean_twt$text <- gsub("http[^[:space:]]*", "",clean_twt$text)
clean_twt$text <- gsub("@[^[:space:]]*", "", clean_twt$text)
clean_twt$text <- str_to_lower(clean_twt$text)
```

Most Common Words by Day

```
pop = clean_twt %>%
  unnest_tokens(output = word, input = text, token = "words") %>%
  anti_join(stop_words, by = "word") %>%
  group_by(date, word) %>%
  summarise(daily_count = n()) %>%
  slice_max(daily_count, n=10, with_ties = FALSE)
```

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

```r
ggplot(data = pop, aes(x=daily_count, y=reorder(word, daily_count))) +
  geom_bar(stat = "identity") +
  facet_wrap(facets = "date", scales = "free")
```



The words `ipcc`, `climate`, and `report` are consistently the most frequently used words per day, which makes sense in a query for "IPCC." There is a distinct pattern in the count of the most common words each day: on April 1-2 there are very few mentions, probably an indicator of fewer tweets overall that fit the query; April 3rd the numbers get higher, with the term `ipcc` showing up about 100 times. Count of these words max out on April 4-5, with many hundreds of uses of `ipcc`, `climate`, and `report`; then usage steadily declines again from April 6-10.

Wordcloud

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

#tokenize tweets to individual words
words <- clean_twt %>%
  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")
```

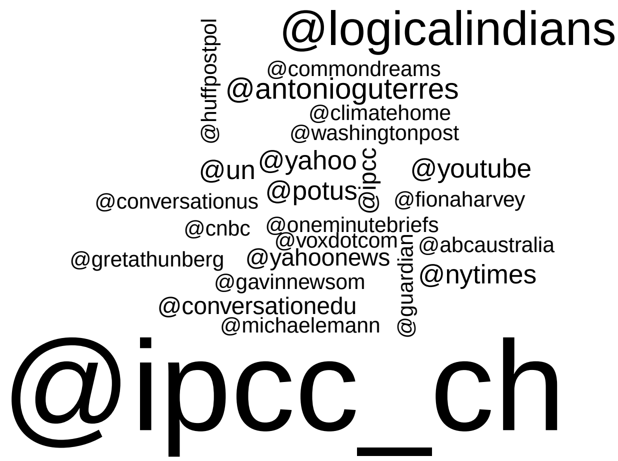
```
words %>%
inner_join(get_sentiments("bing")) %>%
count(word, sentiment, sort = TRUE) %>%
acast(word ~ sentiment, value.var = "n", fill = 0) %>%
comparison.cloud(colors = c("red", "green"),
                 max.words = 100)
```

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

Most Tagged Accounts

```
# using quanteda and corpus() for alternate analyses

corpus = corpus(dat$Title)

tokens = tokens(corpus, remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_remove(stopwords("english")) %>%
  tokens_remove("http[^[:space:]]*", valuetype = "regex") %>%
  tokens_tolower()

tokens_at = tokens(corpus, remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_keep(pattern = "@*")
```

```r
dfm_at = dfm(tokens_at)

#this is useful!
tstat_freq <- textstat_frequency(dfm_at, n = 100)

tidy_at <- tidy(dfm_at) %>%
   count(term) %>%
   with(wordcloud(term, n, max.words = 25))
```



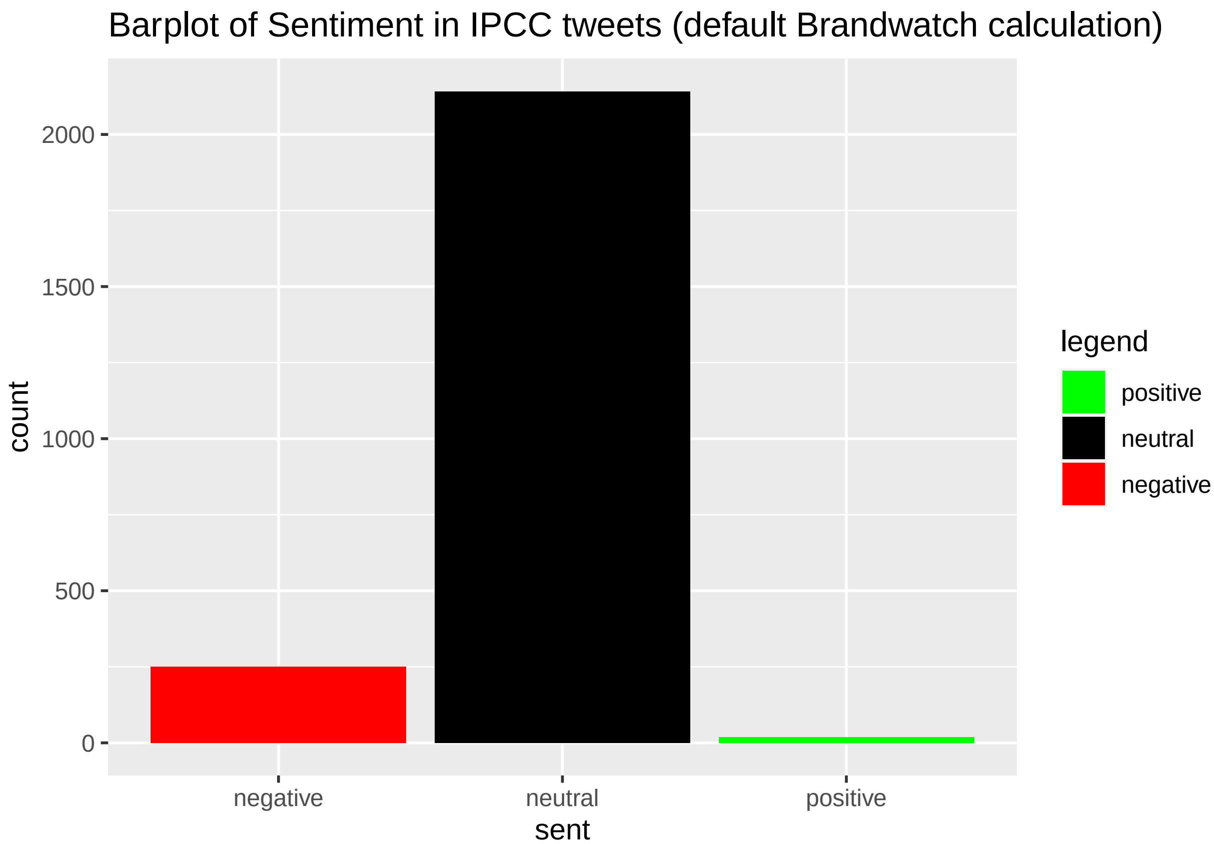

```r
# Show the top 10 accounts
topten = data.frame(tstat_freq) %>%
  filter(rank < 11) %>%
   select(-group) #%>%
  # datatable() # incompatible with pdf output
topten
```

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

Alternate Sentiment Comparison

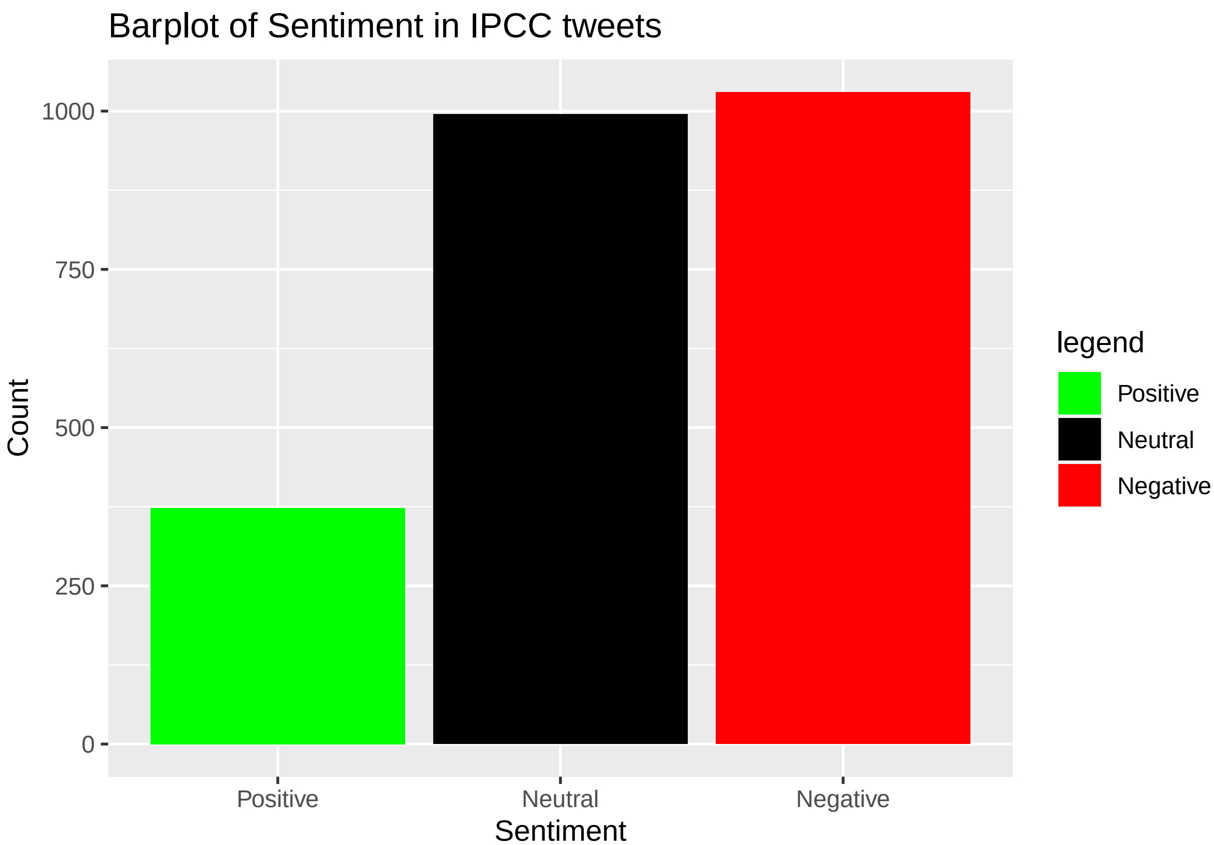
```r
raw_sent <- raw_tweets[,c(4,6,10)] # Extract Date and Title fields

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

# default sentiment
```

```r
bwatch = tweetsent %>%
  group_by(sent) %>%
  summarise(count = n())

ggplot(bwatch, aes(x=sent,y=count))+
  geom_bar(stat = "identity", aes(fill = sent)) +
  ggtitle("Barplot of Sentiment in IPCC tweets (default Brandwatch calculation)") +
  scale_fill_manual("legend",
                  values = c("positive" = "green",
                             "neutral" = "black",
                             "negative" = "red"))
```




```r
# the comparison
tweets_sent <- words %>%
      mutate(sent_score = replace_na(sent_score, 0)) %>%
      group_by(id) %>%
      summarize(sent_score = mean(sent_score, na.rm = T))

neutral <- length(which(tweets_sent$sent_score == 0))
positive <- length(which(tweets_sent$sent_score > 0))
negative <- length(which(tweets_sent$sent_score < 0))

# graph overall sentiments
Sentiment <- c("Positive","Neutral","Negative")
Count <- c(positive,neutral,negative)
output <- data.frame(Sentiment,Count)
output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
```

```
ggplot(output, aes(x=Sentiment,y=Count)) +
  geom_bar(stat = "identity",
           aes(fill = Sentiment)) +
  scale_fill_manual("legend",
                    values = c("Positive" = "green",
                               "Neutral" = "black",
                               "Negative" = "red")) +
  ggtitle("Barplot of Sentiment in IPCC tweets")
```



In a comparison of the default Brandwatch sentiment values and a manual calculation using the `bing` sentiment lexicon, there is a clear mismatch. The Brandwatch analysis labels almost all of the tweets as `neutral`, with a few `negative` and almost no `positive` tweets. In contrast, using the `bing` lexicon to identify all positive and negative words yields 1030 negative and 373 positive tweets. Even by classifying all tweets that had no sentiment words (in addition to tweets that had equal amounts of positive and negative words in the `bing` lexicon) as neutral, this method yielded a total of 995 neutral tweets, less than half of Brandwatch's 2142 `neutral` tweets.