Topic 04 - Sentiment Analysis II

Julia Parish

2022-04-26

Sentiment Analyis II

This text sentiment analysis was completed as an assignment for the course, Environmental Data Science 231: Text and Sentiment Analysis for Environmental Problems. The data was sourced from Twitter.

Original assignment instructions can be found here

Load Libraries

```
library(quanteda)
library(quanteda.sentiment)
library(quanteda.textstats)
library(tidyverse)
library(tidytext)
library(lubridate)
library(wordcloud)
library(reshape2)
library(here)
library(rtweet)
library(paletteer)
library(kableExtra)
library(sentimentr)
```

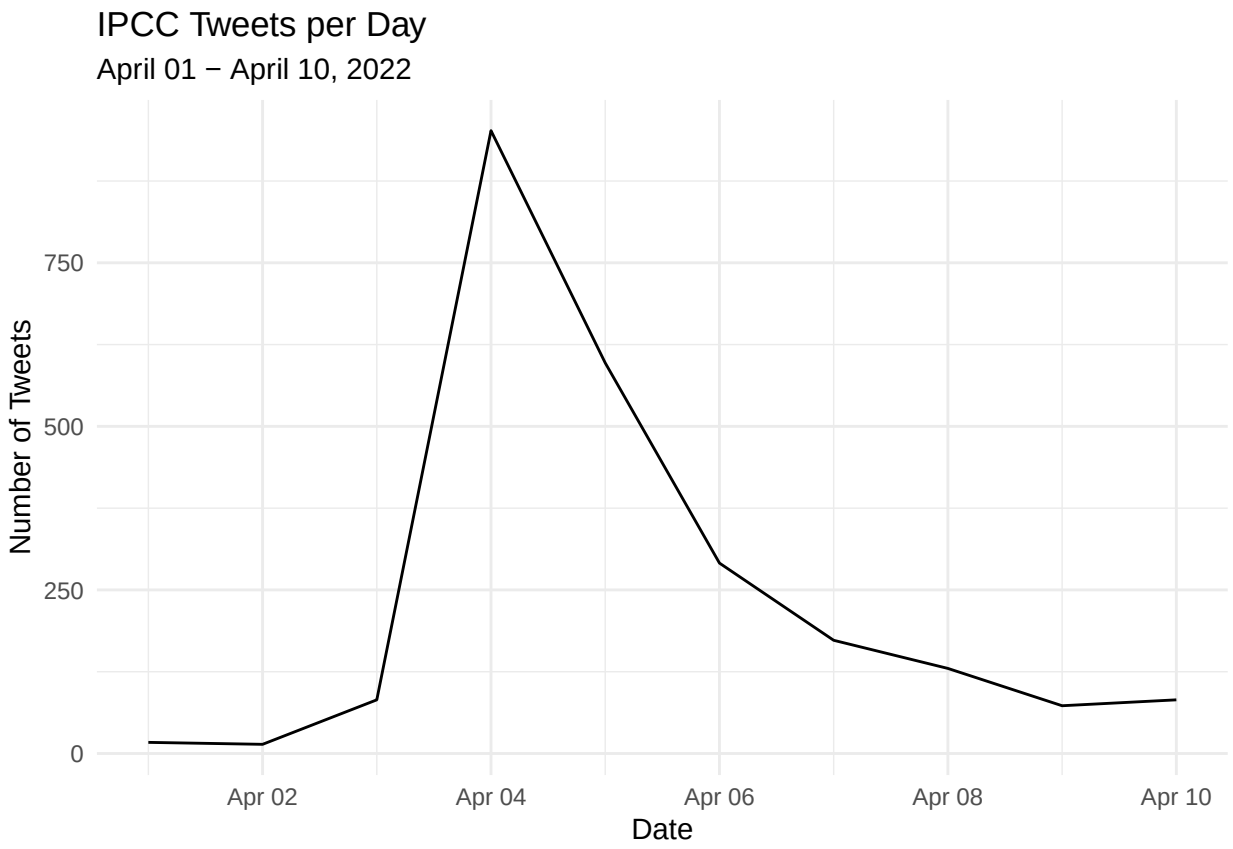
Load IPPC tweet data & create plot of data

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

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

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

head(tweets$text, n = 10)
```

```
##  [1] "thank you, followers, for the great photo suggestions for our upcoming IPCC report - on Monday
##  [2] "Greenpeace: The real solution to the climate crisis will require a rapid transition away from
##  [3] "Governments have a responsibility to ensure that #IPCCReport is grounded in rapid phaseout of
not #FalseClimateSolutions. \n\nRead more in our open letter: https://t.co/4larBPgeba https://t.co/Fv1Op
##  [4] "Next week, the IPCC will publish a new report detailing their new models and policy pathways.
##  [5] "Live stream of virtual IPCC press conference releasing the report on mitigation of climate cha
##  [6] "Attention journalists: The deadline for embargoed materials for the upcoming @IPCC_CH report o
```

```
##  [7] "The IPCC Report and "The Physics of Climate Change" https://t.co/xnxP3fup2a"
##  [8] "With time running short and most of the Summary for Policymakers yet to be approved, #IPCC Worl
##  [9] "A helpful perspective on how to talk about the scenarios discussed in the forthcoming IPCC repo
## [10] "The private sector is an integral component of the water cycle and has much to lose as critical
```

```
#simple plot of tweets per day
tweets %>%
  count(date) %>%
  ggplot(aes(x = date, y = n))+
  geom_line() +
  labs(title = "IPCC Tweets per Day",
       subtitle = "April 01 - April 10, 2022",
       x = "Date",
       y = "Number of Tweets") +
  theme_minimal()
```



Questions

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.

```
# keep original text column to track changes
tweets_clean <- tweets %>%
  mutate(text_clean = text)
```

```r
# remove mentions and website links
tweets_clean$text_clean <- str_remove(tweets_clean$text_clean, "@[a-z,A-Z]*")

tweets_clean$text_clean <- str_remove(tweets_clean$text_clean, "[:digit:]")

tweets_clean$text_clean <- gsub("http.*","", tweets_clean$text_clean)

tweets_clean$text_clean <- gsub("https.*","", tweets_clean$text_clean)

# remove punctuations
tweets_clean$text_clean <- gsub('[[:punct:]]', '', tweets_clean$text_clean)

#tokenise tweets and remove stop words
words <- tweets_clean %>%
  select(id, date, text, text_clean) %>%
  unnest_tokens(output = word, input = text_clean, token = "words") %>%
  anti_join(stop_words, by = "word")

#clean tokens
# remove numbers
clean_tokens <- str_remove_all(words$word, "[:digit:]")

# remove mentions
clean_tokens <- str_remove_all(clean_tokens, "@[a-z,A-Z]*")

# remove apostrophes
clean_tokens <- gsub("'s", '', clean_tokens)

# remove unnecessary twitter formats
clean_tokens <- str_remove_all(clean_tokens, "t.co")

# stem the token "ipcc" as there are some plural instances
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "ipcc[a-z, A-Z]*",
                                replacement = "ipcc")


# stem the token "fuel" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "fuel[a-z, A-Z]*",
                                replacement = "fuel")

# stem the token "biofuel" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "biofuel[a-z, A-Z]*",
                                replacement = "biofuel")

# stem the token "headline" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "headline[a-z, A-Z]*",
                                replacement = "headline")

# stem the token "regulation" as it may occur in the plural form
```

```r
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "regulation[a-z, A-Z]*",
                                replacement = "regulation")

# stem the token "follower" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "follower[a-z, A-Z]*",
                                replacement = "follower")

# stem the token "suggestion" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "suggestion[a-z, A-Z]*",
                                replacement = "suggestion")

# stem the token "solution" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "solution[a-z, A-Z]*",
                                replacement = "solution")

# stem the token "reduction" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "reduction[a-z, A-Z]*",
                                replacement = "reduction")

# stem the token "risk" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "risk[a-z, A-Z]*",
                                replacement = "risk")

# stem the token "scenario" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "scenario[a-z, A-Z]*",
                                replacement = "scenario")

# stem the token "submission" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "submission[a-z, A-Z]*",
                                replacement = "submission")

words$clean <- clean_tokens

# remove the empty strings
tib <-subset(words, clean != "")

#reassign
words <- tib

head(words)
```

```
## # A tibble: 6 x 5
##      id date       text                                                word  clean
##   <int> <date>     <chr>                                               <chr> <chr>
## 1     1 2022-04-01 "thank you, followers, for the great photo sugge~ foll~ foll~
## 2     1 2022-04-01 "thank you, followers, for the great photo sugge~ photo photo
```

```
## 3     1 2022-04-01 "thank you, followers, for the great photo sugge~ sugg~ sugg~
## 4     1 2022-04-01 "thank you, followers, for the great photo sugge~ upco~ upco~
## 5     1 2022-04-01 "thank you, followers, for the great photo sugge~ ipcc  ipcc
## 6     1 2022-04-01 "thank you, followers, for the great photo sugge~ repo~ repo~
```

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

```
words_freq <- words %>%
  group_by(clean) %>%
  summarise(n()) %>%
  top_n(10) %>%
  rename("freq" = "n()") %>%
  select(clean)

words_top10 <- inner_join(words_freq, words, by = "clean") %>%
  group_by(date, clean) %>%
  summarize(n()) %>%
  rename("freq" = "n()")
```

```
top10term_plot <- ggplot(data = words_top10, aes(x = date, y = freq)) +
  geom_line(aes(color = clean)) +
  geom_text(data=words_top10[34,], y = 325, label="Change", vjust=1, hjust=-0.01,
            size = 2.5, color = "grey24") +
  geom_text(data=words_top10[34,], y = 475, label="Report",vjust=1, hjust=-0.01,
            size = 2.5, color = "grey24") +
  geom_text(data=words_top10[34,], y = 600, label="Climate", hjust=-0.25,
            size = 2.5, color = "grey24") +
    geom_text(data=words_top10[34,], y = 650, label="IPCC", hjust=-0.25,
            size = 2.5, color = "grey24") +
  labs(title = "10 Most Common IPCC-related Tweet Terms",
       subtitle = "April 01 - April 10, 2022",
       caption = "Data source: Twitter",
       x = "Date",
       y = "Frequency",
       color = "Term") +
  scale_color_paletteer_d("rcartocolor::Safe") +
  theme_minimal()

top10term_plot
```

```
top10term_table = aggregate(words_top10$clean,
                            list(words_top10$date), paste, collapse=", ") %>%
  rename(Date = Group.1) %>%
  rename(top_words = x) %>%
  kable(col.names = c("Date", "Top 10 Words")) %>%
  kable_paper(full_width = TRUE)

top10term_table
```

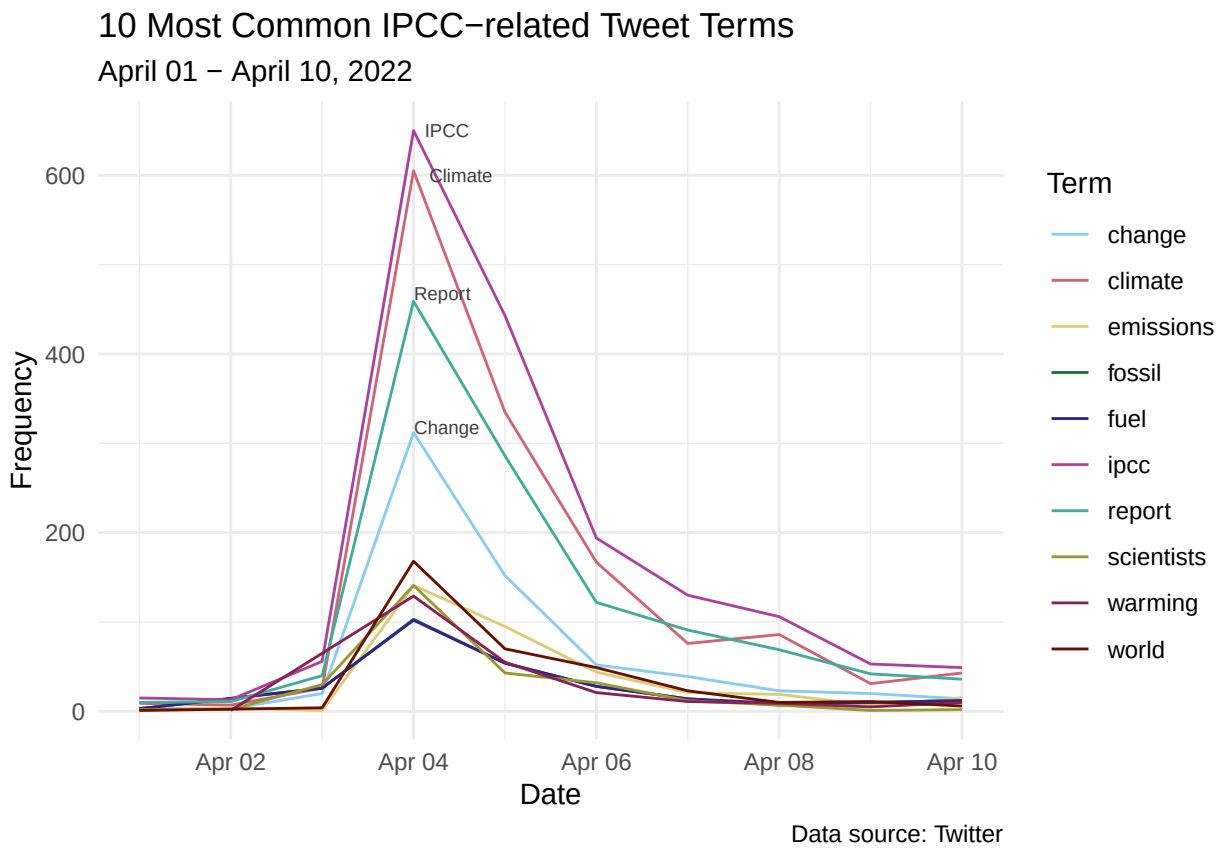


Figure 1: 10 Most Common IPCC-related Tweet Terms

Date	Top 10 Words
2022-04-01	change, climate, emissions, fossil, fuel, ipcc, report, scientists, world
2022-04-02	change, climate, emissions, ipcc, report, scientists, warming
2022-04-03	change, climate, emissions, fossil, fuel, ipcc, report, scientists, world
2022-04-04	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world
2022-04-05	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world
2022-04-06	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world
2022-04-07	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world
2022-04-08	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world
2022-04-09	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world
2022-04-10	change, climate, emissions, fossil, fuel, ipcc, report, scientists, warming, world

Answer

The Intergovernmental Panel on Climate Change (IPCC) held a virtual press conference to present a summary of the report `Climate Change 2022: Mitigation of Climate Change` on Monday, April 04, 2022. There was a large spike on the day of the press conference. More `Top 10` words in tweets occured on that day and on days preceding the press conference. The `Top 10` words found in tweets were very similar days before and after the conference, with `climate` and `change` being the most common.

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

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

```
cloud <- words %>% inner_join(get_sentiments("bing")) %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("slateblue3", "goldenrod2"),
                   max.words = 100)
```

```
cloud
```

```
## NULL
```

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
#summary(corpus)
# text: tweet ID, Types: species words, Tokens: total words
```

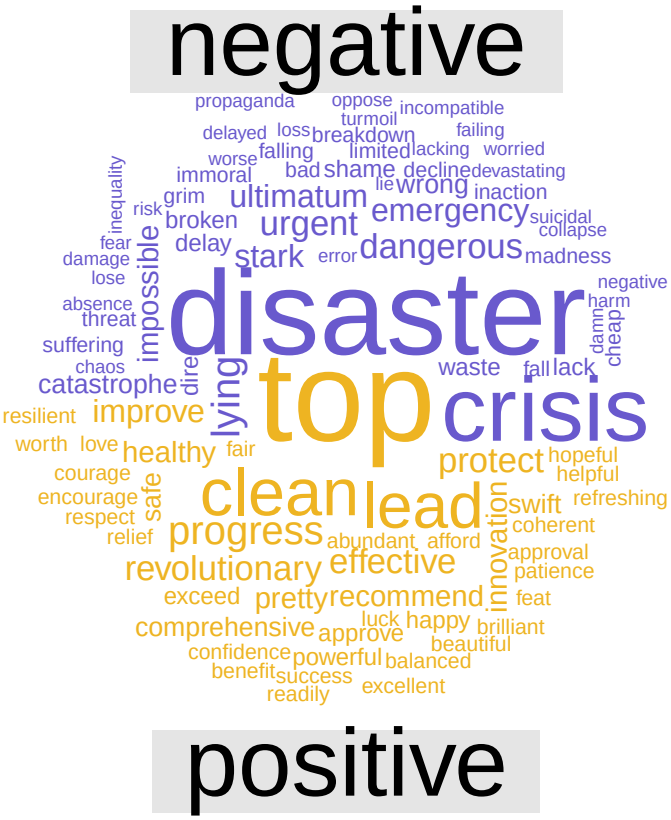


Figure 2: Sentiment word cloud of IPCC-related Tweet Terms between April 01 - April 10, 2022. Positive sentiment terms are identified in the yellow hue, while negative terms are purple.

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

# feature matrix - shows location of each features in the corpus aka located in the tweet : document fe
dfm_tags<- dfm(tagged_accts)

# frequency of hashtags
tstat_freq <- textstat_frequency(dfm_tags, n = 100)
head(tstat_freq, 10)
```

```
##             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
```

```
#tidytext gives us tools to convert to tidy from non-tidy formats
tags_tib <- tidy(dfm_tags)

tags_tib %>%
   count(term) %>%
   with(wordcloud(term, n, color = "slateblue3", max.words = 100))
```

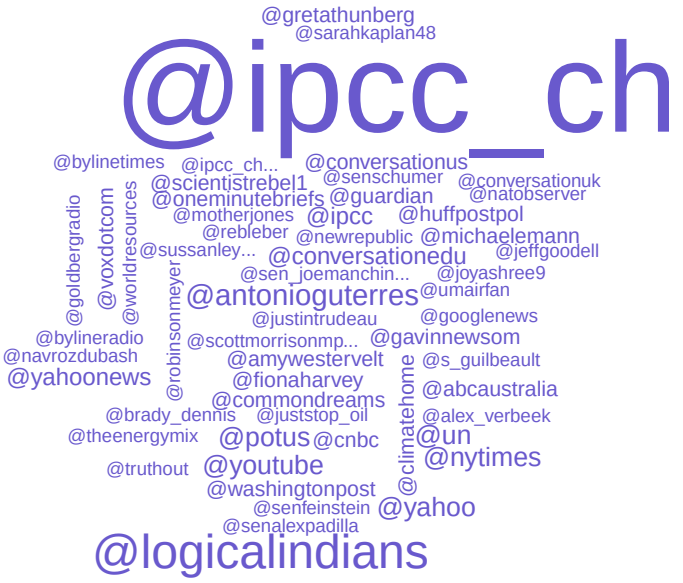


Figure 3: Word cloud of IPCC-related Twitter accounts tagged between April 01 - April 10, 2022.

```
top_tags <- tags_tib %>%
  group_by(term) %>%
  summarize(n()) %>%
  rename("freq" = "n()") %>%
```

```
  top_n(10)
```

```
top10user_plot <- top_tags %>%
  mutate(term = fct_relevel(term,
            "@ipcc_ch", "@logicalindians", "@antonioguterres", "@nytimes", "@yahoo", "@potus", "@un", "
  ggplot(aes(x = freq, y = term)) +
  geom_point(color = "slateblue2") +
  labs(title = "Top 10 Tagged IPCC-related Accounts",
       subtitle = "April 01 - April 10, 2022",
       x = "Frequency of Mentions",
       y = "Twitter Account",
       caption = "Data Source: Twitter") +
  theme_minimal()

top10user_plot
```

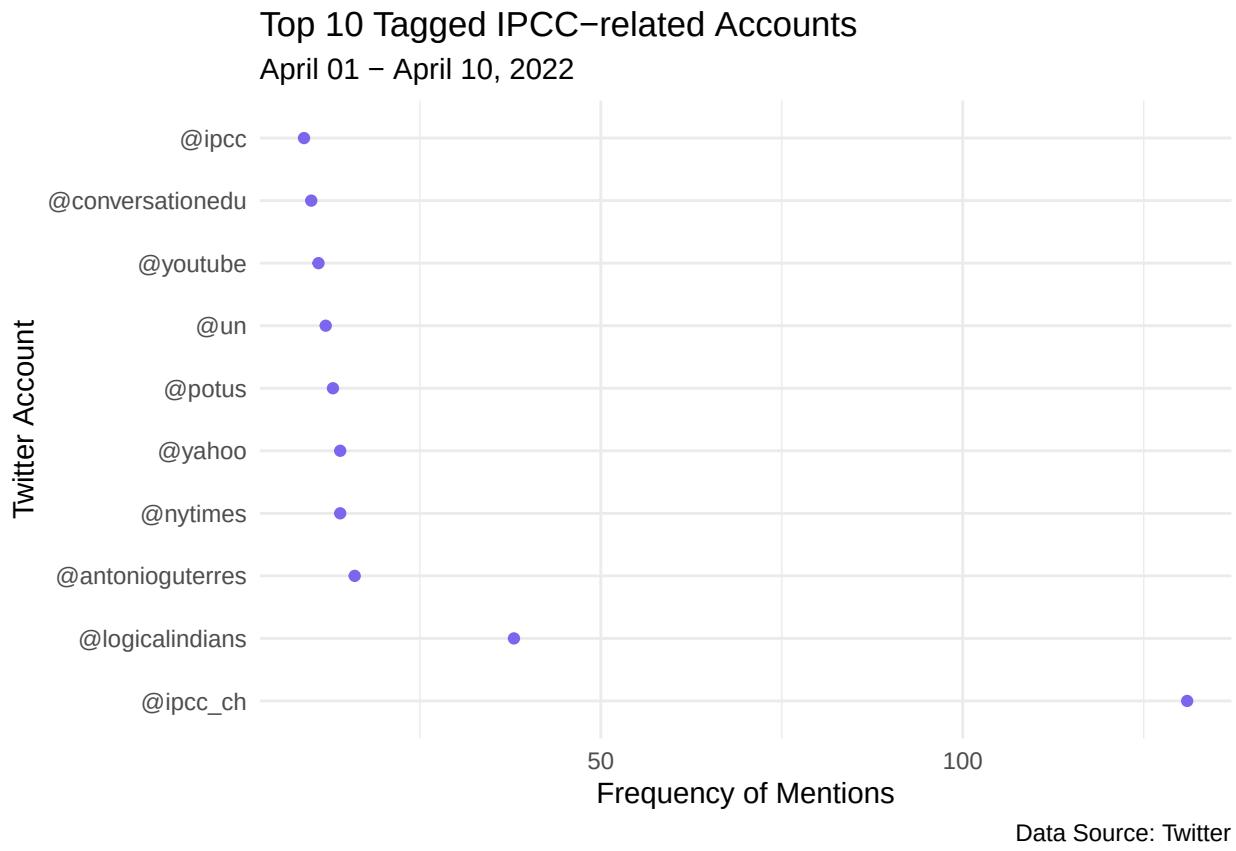


Figure 4: Top 10 Twitter Accounts tagged in IPCC related tweets between April 1 - April 10, 2022

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).

```
# Extract Date, Title, and Sentiment fields
dat2<- raw_tweets[,c(4, 6, 10)]
```

```
tweet_sentiment <- tibble(element_id = seq(1:length(dat2$Title)),
                    date = as.Date(dat2$Date,'%m/%d/%y'),
                    text = dat2$Title,
                    brandwatch_sentiment = dat2$Sentiment)
```

```
# clean tweets
tweet_sentiment$text <- gsub("http[^[:space:]]*", "",tweet_sentiment$text)

tweet_sentiment$text <- str_to_lower(tweet_sentiment$text)

tweet_sentiment$text <- gsub("@*", "", tweet_sentiment$text)
```

```
tweet_sentiment$text <- sentimentr::replace_emoji(tweet_sentiment$text)

tweet_sentiment$text <- sentimentr::replace_emoticon(tweet_sentiment$text)

tweet_sentiment$text <- gsub("<*>", "", tweet_sentiment$text)

tweet_sentiment$text <- str_remove_all(tweet_sentiment$text, "[:digit:]")

# calculate tweet sentiment
tweet_sen <- sentimentr::sentiment(tweet_sentiment$text)

# calculate tweet emotion at sentence level
tweet_emotion <- sentimentr::emotion(tweet_sentiment$text)
```

```
#join with sentence data
sentiment <- inner_join(tweet_sentiment, tweet_sen, by = "element_id")
```

```
sentiment2 <- sentiment %>%
  mutate(sent_category = case_when(
    sentiment < 0 ~ "negative",
    sentiment > 0 ~ "positive",
    sentiment == 0 ~ "neutral"))
```

```
sentiment3 = sentiment2 %>%   mutate(comparison = case_when(
   brandwatch_sentiment == sent_category & brandwatch_sentiment == "positive" ~ "positive",
   brandwatch_sentiment == sent_category & brandwatch_sentiment == "negative" ~ "negative",
   brandwatch_sentiment == sent_category & brandwatch_sentiment == "neutral" ~ "neutral",
   brandwatch_sentiment != sent_category ~ "no_match"))
```

```
sentiment4 = sentiment3 %>%
  mutate(comparison = fct_relevel(comparison, "positive", "neutral", "negative", "no_match")) %>%
  count(comparison)
```

```
sent_compare_plot <- sentiment4 %>%
  ggplot(aes(x = comparison, y = n)) +
  geom_point(color = "slateblue3", size = 5) +
  geom_text(aes(x = "positive", y = 180, label = "27"), stat = "unique",
            size = 2.5, color = "grey24") +
  geom_text(aes(x = "neutral", y = 1350, label = "1205"), stat = "unique",
            size = 2.5, color = "grey24") +
  geom_text(aes(x = "negative", y = 420, label = "290"), stat = "unique",
            size = 2.5, color = "grey24") +
  geom_text(aes(x = "no_match", y = 2990, label = "2842"), stat = "unique",
```

```
            size = 2.5, color = "grey24") +
  labs(title = "Sentiment Comparison of IPCC-related Tweets, April 2022",
       subtitle = "Evaluated when Brandwatch and sentimentR matched (or not)",
       caption = "Data Source: Brandwatch and sentimentR",
       x = "Sentiment",
       y = "Total Sentence Count") +
  theme_minimal()

sent_compare_plot
```

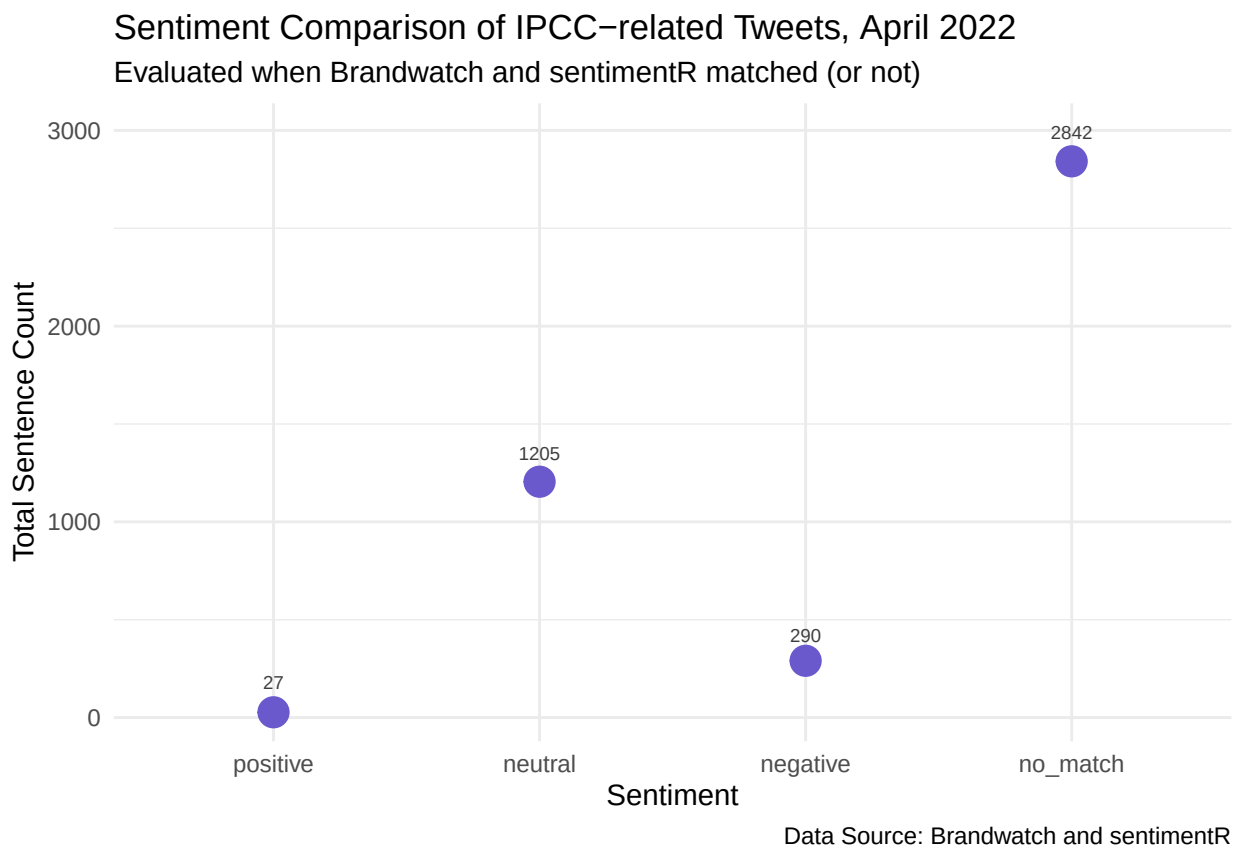


Figure 5: Compared two sentiment classification systems on IPCC related tweets posted between April 01 - April 10, 2022. Brandwatch sentiment and the package sentimentR were used for comparison. When the two sentiment classifcations were the same, they were designated by the polarity classification: positive, neutral, or negative. When the two classifications were not the same for a sentence, it was classified as: no_match.

Bonus - Emoji Frequency Exploration

```
# extract emojis from tweets
ipcc_emojis <- emojis %>%
  # for each emoji, find tweets containing this emoji
  mutate(tweet = map(code, ~grep(.x, tweets_clean$text))) %>%
  unnest(tweet) %>%
  # count the number of tweets in which each emoji was found
  count(code, description) %>%
  mutate(emoji = paste(code, description))
```

```
plot_emoji <- ipcc_emojis %>%
  top_n(5, n) %>%
  ggplot() +
  geom_col(aes(x = fct_reorder(emoji, n), y = n, fill = n),
           color = "grey58", width = 1) +
  scale_fill_gradientn("n", colors = brewer.pal(5, "Set2")) +
  labs(x = "", y = "Count",
       title = "Most Popular Emojis in IPCC tweets",
       subtitle = "April 01 - April 10, 2022") +
  coord_flip()

#plot_emoji
```

```
# save emoji plot as png
# ggsave(file = "emojiplot.png", plot = plot_emoji,
#   scale = 1, width = 6, height = 6,
#   units = "in", dpi = 300)
```