IPCC Tweets April 2022

Peter Menzies

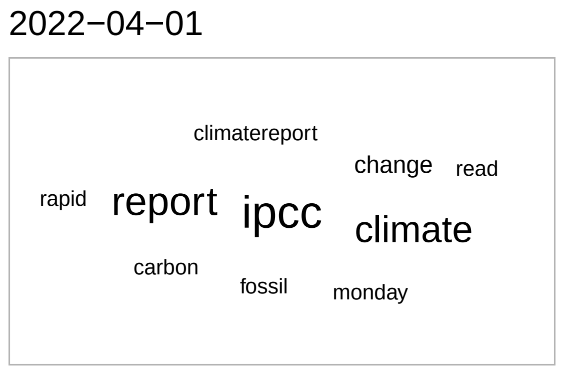
2022-04-20

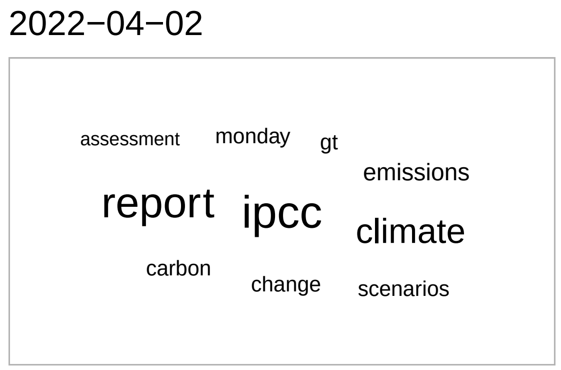
1. Cleaning tweets

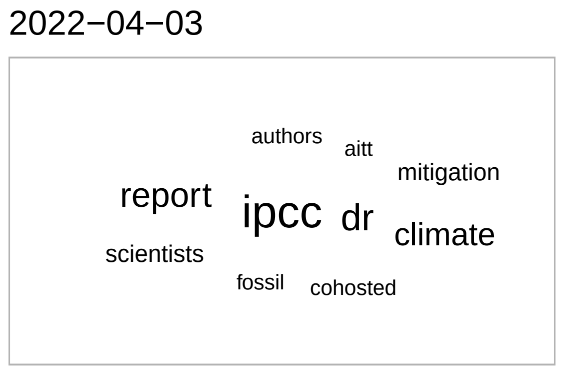
```r
data <- read_csv(here("data", "IPCC_tweets_April1-10_sample.csv"))

tweets <- data[2:11] %>%
  clean_names() %>%
  rename(text = title) %>%
  mutate(date = as.Date(date,'%m/%d/%y'),
         id = seq_along(text),
         text = tolower(text))

# URLs
tweets$text <- gsub("(http\\S+)|(www\\S+)", "", tweets$text)
# Twitter accounts
tweets$text <- gsub("@\\S+", "", tweets$text)
# Numbers and punctuation
tweets$text <- gsub("[[:digit:][:punct:]]", "", tweets$text)
# Emojis ("{So}" = Unicode "Other_Symbol")
tweets$text <- gsub("\\p{So}", "", tweets$text, perl = TRUE)
```

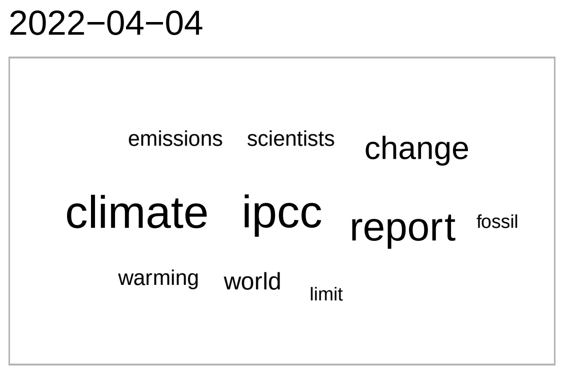
2. Common terms

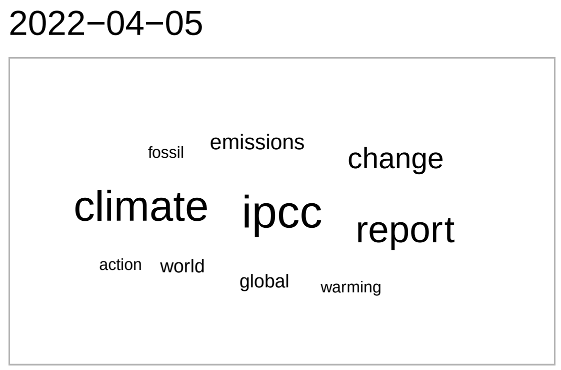
```r
words <- tweets %>%
  select(id, date, text) %>%
  unnest_tokens(output = word, input = text, token = "words") %>%
  anti_join(stop_words, by = "word")

dates <- sort(unique(words$date))

for (i in seq_along(dates)) {

  cloud <- words %>%
    filter(date == dates[i]) %>%
    count(word) %>%
    slice_max(n, n = 10, with_ties = FALSE) %>%
    ggplot(aes(label = word, size = n)) +
    geom_text_wordcloud() +
    labs(title = as.character(dates[i])) +
    scale_size_area(max_size = 6) +
    theme_light()

  name <- paste0("cloud_", i)

  assign(name, cloud)

  plot(get(name))

}
```

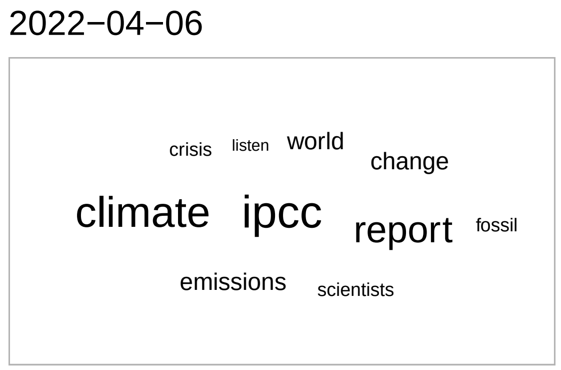


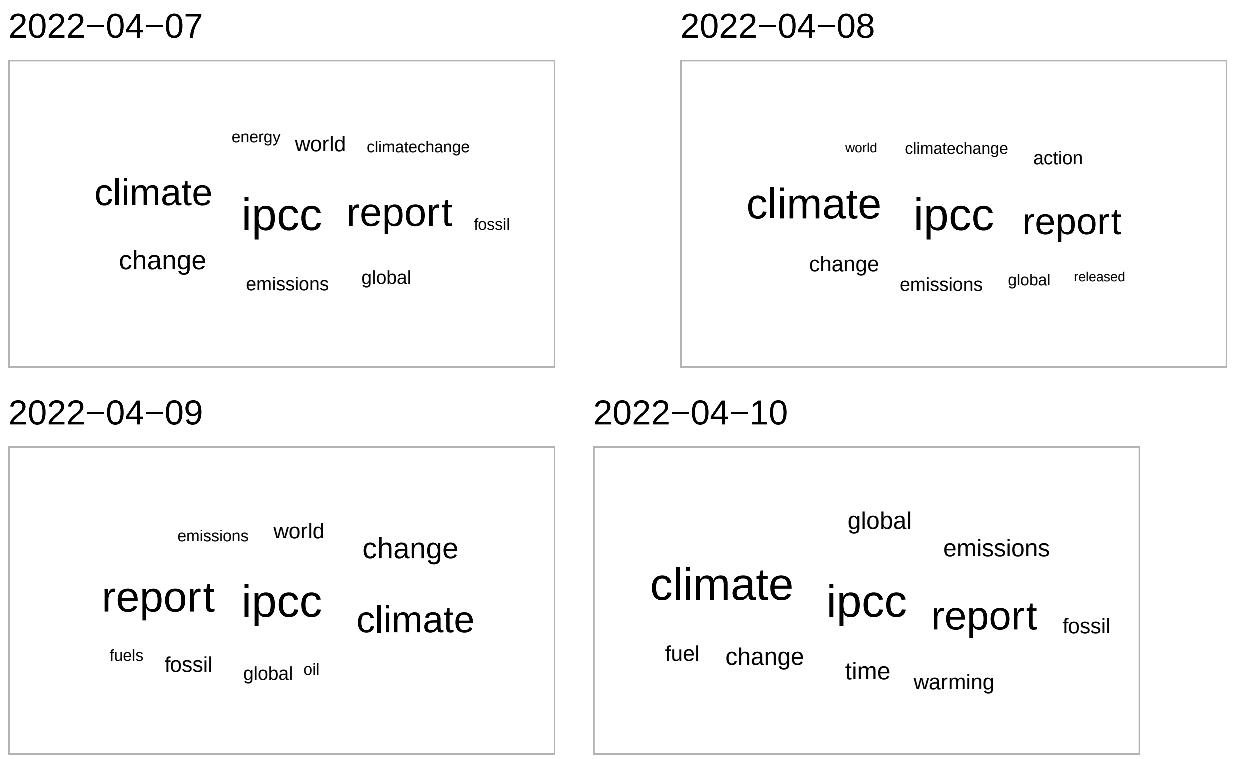








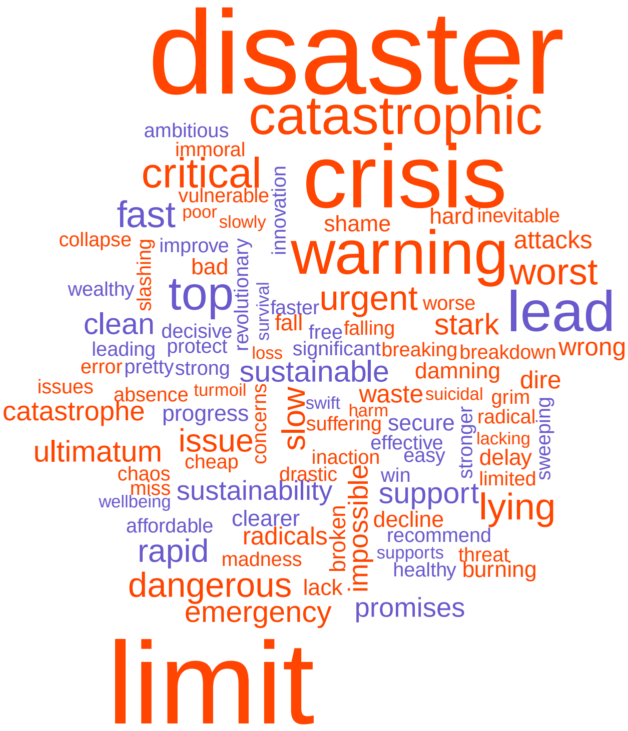




Looking at the ten most common words each day, I don't notice any particularly apparent trends. The most striking word in my opinion is "crisis" which appears as a top ten word after the report was released—this would align with the sentiment analysis.

3. Wordcloud with colors based on sentiment

```r
words %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  mutate(color = case_when(sentiment == "positive" ~ "slateblue3",
                           sentiment == "negative" ~ "orangered")) %>%
  with(wordcloud(word, n, max.words = 100, colors = color,
                 ordered.colors = TRUE))
```



4. Top 10 most tagged accounts

```
corpus <- corpus(data$Title)

tokens <- tokens(corpus)

tokens <- tokens %>%
  tokens(remove_punct = TRUE,remove_numbers = TRUE) %>%
  tokens_select(stopwords('english'),selection='remove') %>%
  tokens_tolower()

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

dfm <- dfm(accounts)

account_freq <- textstat_frequency(dfm, n = 10) %>%
  rename(account = feature)

account_freq[,1:3] %>% gt::gt()
```

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

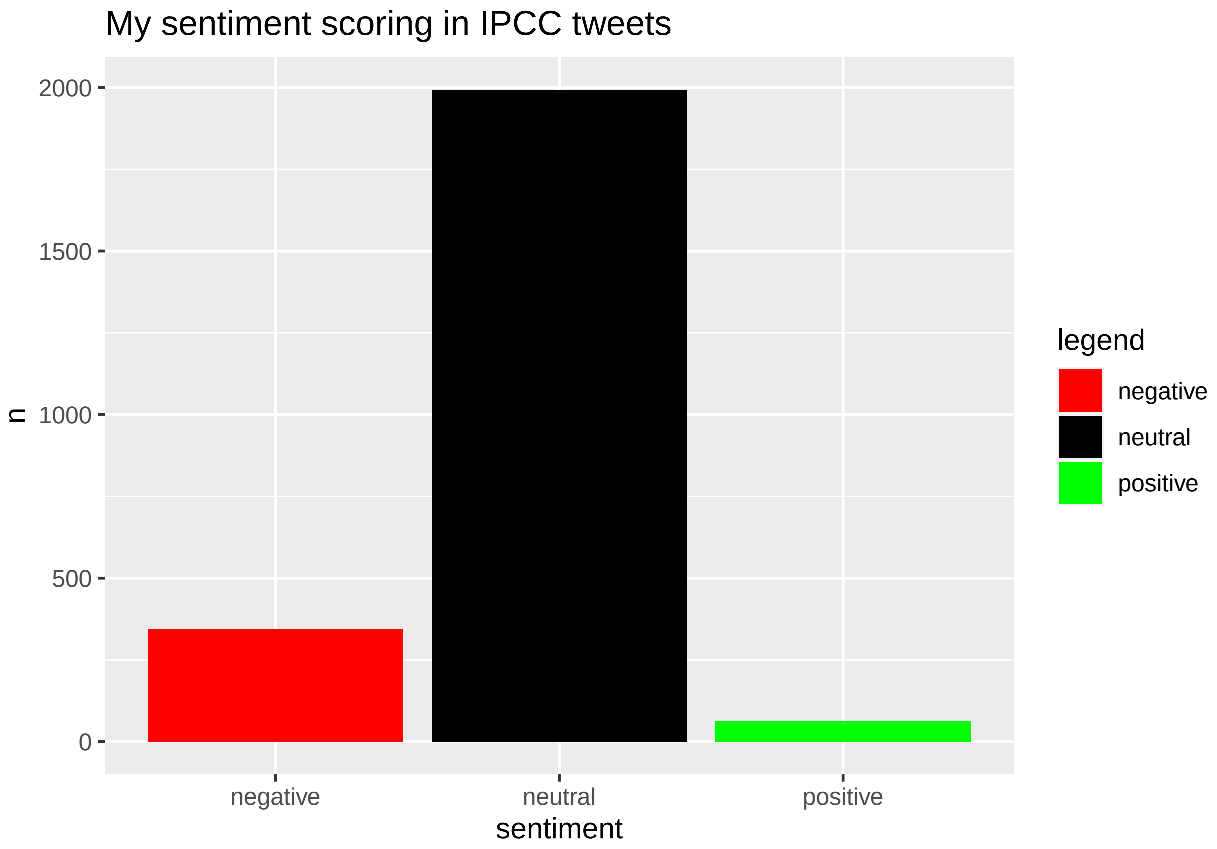
5. Comparing polarity scoring

I'm categorizing polarity by giving positive words a value of 1, negative -1, and neutral (those not in `bing`) a value of 0. I sum all sentiment values in each tweet, and then consider any sums (scores) greater than 1 to be positive, less than -1 to be negative, and everything in between neutral.

```
sent_words <- words %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  mutate(value = case_when(sentiment == "positive" ~ 1,
                           sentiment == "negative" ~ -1)) %>%
  group_by(id) %>%
  mutate(score = sum(value)) %>%
  ungroup() %>%
  mutate(overall = case_when(score >= -1 & score <= 1 ~ "neutral",
                             score > 1 ~ "positive",
                             score < -1 ~ "negative"))
```

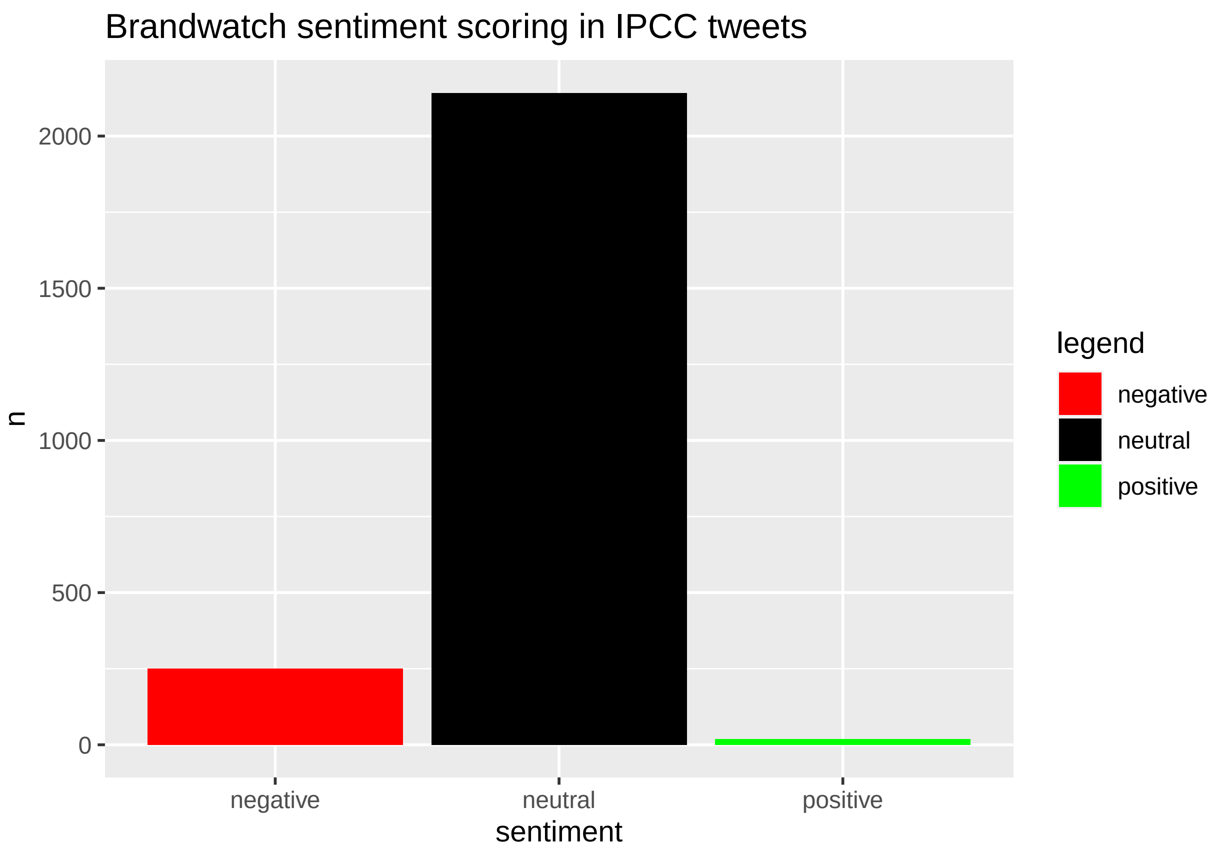
```
sent_words <- words %>%
  left_join(get_sentiments("bing"), by = "word") %>%
  mutate(value = case_when(sentiment == "positive" ~ 1,
                           sentiment == "negative" ~ -1,
                           is.na(sentiment) ~ 0)) %>%
  group_by(id) %>%
  summarize(score = sum(value)) %>%
  mutate(overall = case_when(score >= -1 & score <= 1 ~ "neutral",
                             score > 1 ~ "positive",
                             score < -1 ~ "negative"))
```

```
sent_counts <- sent_words %>%
  group_by(overall) %>%
  count()

ggplot(sent_counts, aes(x = overall,y = n))+
  geom_bar(stat = "identity", aes(fill = overall))+
  scale_fill_manual("legend", values = c("negative" = "red", "neutral" = "black", "positive" = "green")
  labs(x = "sentiment", title = "My sentiment scoring in IPCC tweets")
```



Brandwatch sentiment distribution

```
brandwatch <- tweets %>%
  group_by(sentiment) %>%
  count()

ggplot(brandwatch, aes(x = sentiment,y = n))+
  geom_bar(stat = "identity", aes(fill = sentiment))+
  scale_fill_manual("legend", values = c("negative" = "red", "neutral" = "black", "positive" = "green")
  ggtitle("Brandwatch sentiment scoring in IPCC tweets")
```



My approach seems to have categorized tweets quite similarly to that of Brandwatch, with the only difference being that my method assigned a few more polar labels than Brandwatch did.