Thomas Fire Sentiment Analysis

Joe Desacaro, Connor Flynn, Grace Lewin, Shale Hunter, Steven Cognac

2022-05-31

```
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(here)
```

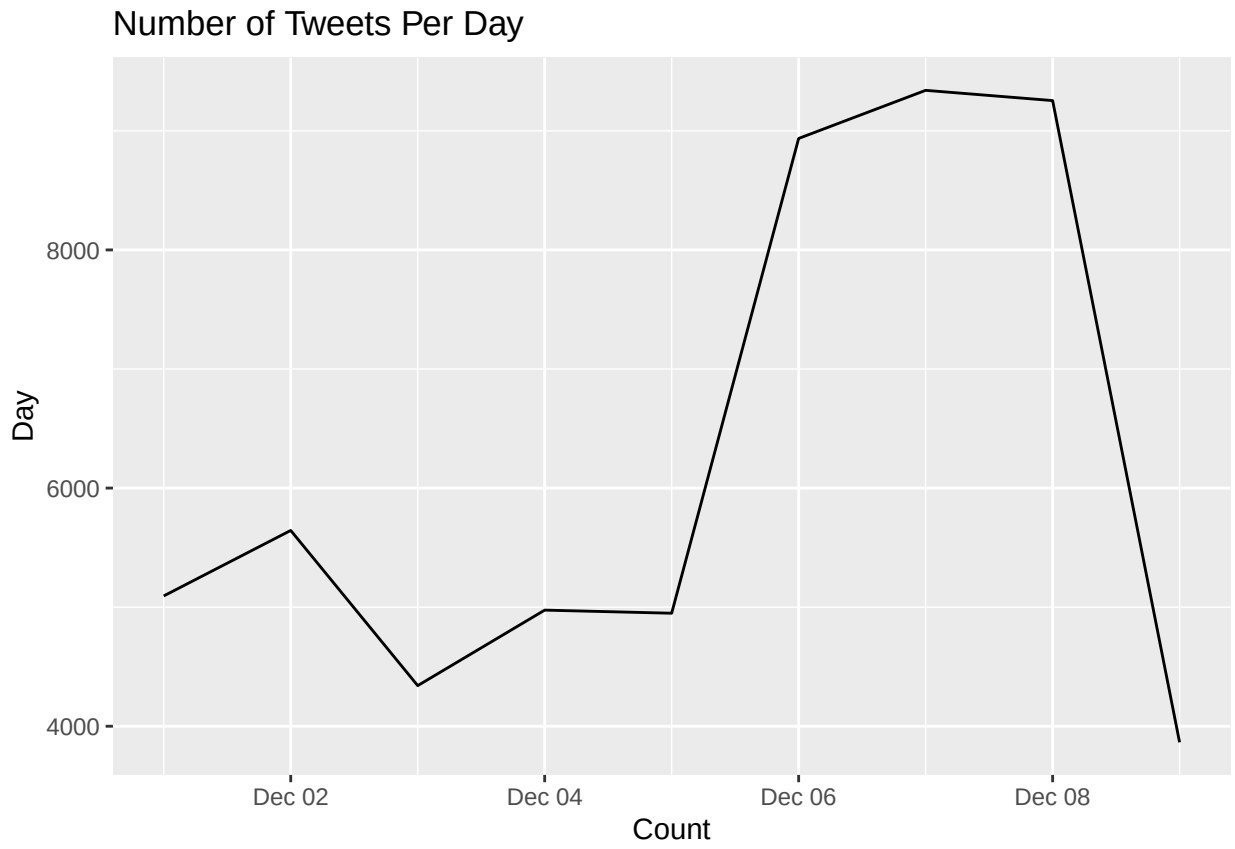
Thomas Fire Sentiment Analysis - Twitter Read in the data

```
all_tweet_files <- list.files(path = here("data/"),
                              pattern = "*.csv")
```

```
temp_df <- data.frame() # make empty df to put new data into

for (i in 1:length(all_tweet_files)){
  current_file <- read_csv(file = here("data",all_tweet_files[i]), # read current csv
                           skip = 6) # skip first 6 lines of csv because they are not helpful
  temp_df <- rbind(temp_df, # bind to df
                   current_file)
  print(paste("Done with",i))
}
```

```
## [1] "Done with 1"
## [1] "Done with 2"
## [1] "Done with 3"
## [1] "Done with 4"
## [1] "Done with 5"
## [1] "Done with 6"
## [1] "Done with 7"
## [1] "Done with 8"
## [1] "Done with 9"
## [1] "Done with 10"
## [1] "Done with 11"
## [1] "Done with 12"
```

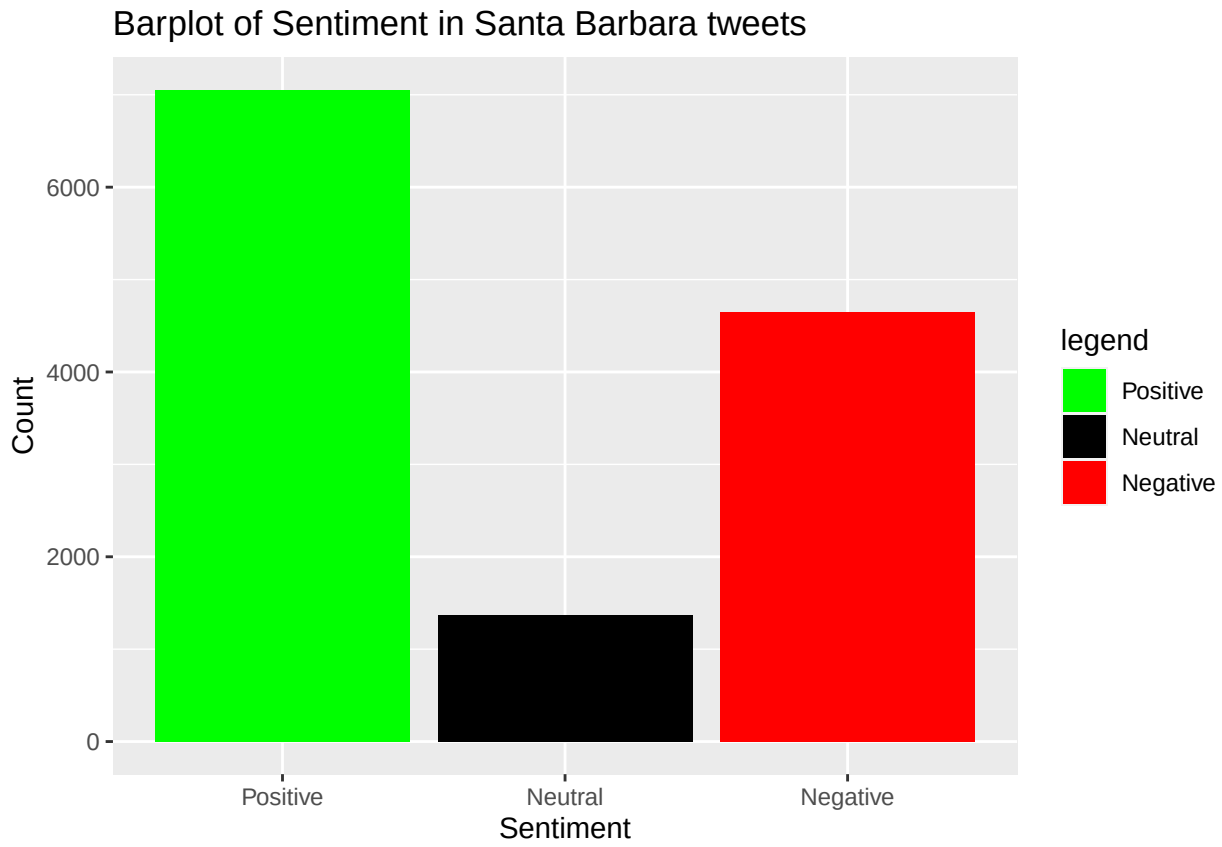
```r
data <- temp_df[,c(4,5)] # Extract Date and Title fields from temp_df

tweets <- tibble(text = data$Title, # make tweet text column
                 id = seq(1:length(data$Title)), # make id sequence
                 date = as.Date(data$Date,'%m/%d/%y')) %>%  # make date column
  mutate(text = str_replace(string = text,
                            pattern = "http.*[:space:]",
                            replacement = ""),
         text = str_replace(string = text,
                            pattern = "http.*$",
                            replacement = ""),
         text = str_replace(string = text,
                            pattern = "@.*[:space:]",
                            replacement = ""),
         text = str_replace(string = text,
                            pattern = "@.*$",
                            replacement = ""),
         text = str_replace_all(string = text,
                                pattern = "rt",
                                replacement = ""),
         text = str_to_lower(text))

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

#simple plot of tweets per day
tweets %>%
  count(date) %>%
  ggplot(aes(x = date, y = n)) +
  geom_line() +
  labs(title = "Number of Tweets Per Day",
       x = "Count",
       y = "Day")
```




```
#let's clean up the URLs from the tweets
# tweets$text <- gsub("http[^[:space:]]*", "",tweets$text) # pull out https and urls and convert to bla

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

#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(nrc_sent, by = "word") %>%
  left_join(
    tribble(
      ~sentiment, ~sent_score,
      "positive", 1,
      "negative", -1),
    by = "sentiment") %>%
  filter(word != "rt") # remove rt as a word
```

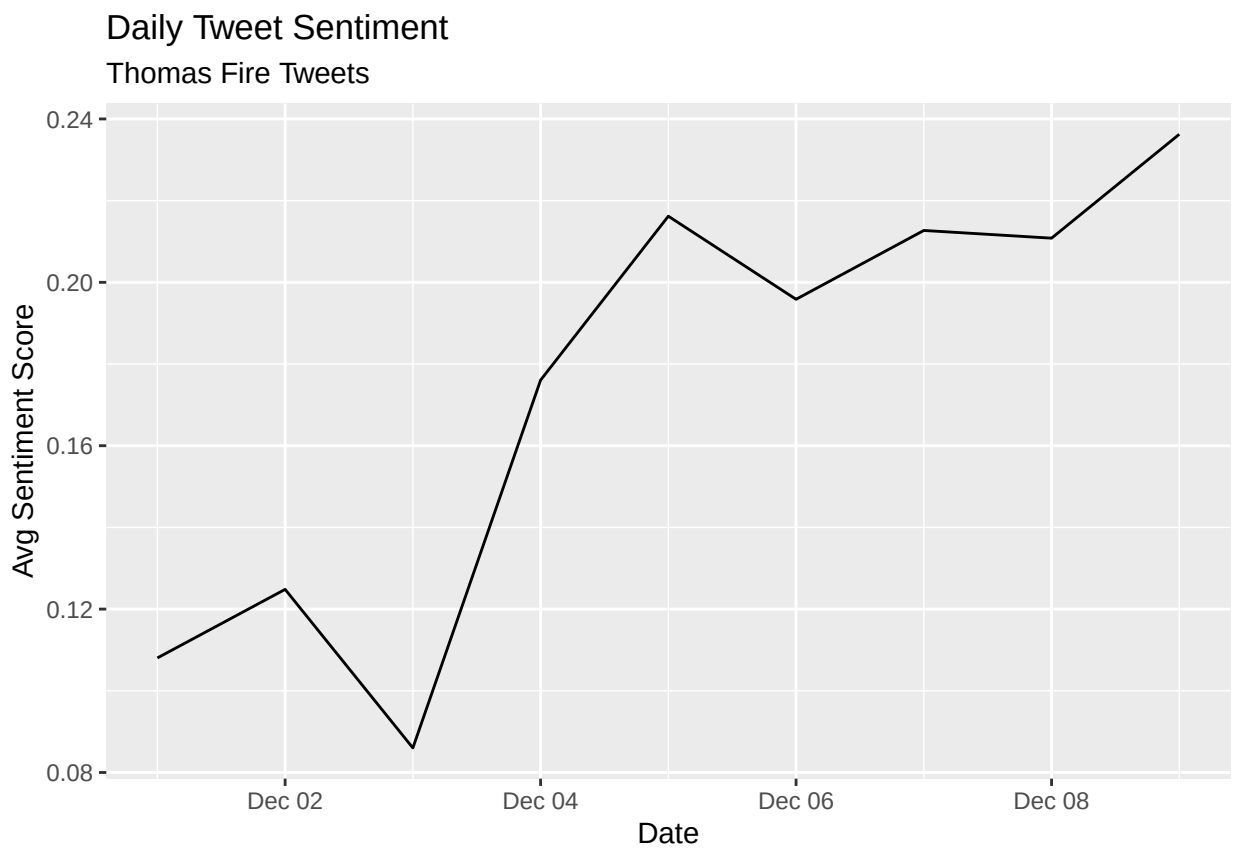
```
#take average sentiment score by tweet
tweets_sent <- tweets %>%
  left_join(
    words %>%
      group_by(id) %>%
```

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

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

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 Santa Barbara tweets")
```




```
# tally sentiment score per day
daily_sent <- tweets_sent %>%
  group_by(date) %>%
  summarize(sent_score = mean(sent_score, na.rm = T))

daily_sent %>%
  ggplot( aes(x = date, y = sent_score)) +
  geom_line() +
```

```
    labs(x = "Date",
    y = "Avg Sentiment Score",
    title = "Daily Tweet Sentiment",
    subtitle = "Thomas Fire Tweets")
```




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

The quanteda package quanteda is a package (actually a family of packages) full of tools for conducting text analysis. quanteda.sentiment (not yet on CRAN, download from github) is the quanteda modular package for conducting sentiment analysis.

quanteda has its own built in functions for cleaning text data. Let's take a look at some. First we have to clean the messy tweet data:

```
## Corpus consisting of 56398 documents, showing 100 documents:
##
##     Text Types Tokens Sentences
##    text1    43     65         5
##    text2    19     19         2
##    text3     6      6         1
##    text4    27     30         3
##    text5     5      5         1
##    text6    16     16         2
##    text7    38     41         2
##    text8    19     25         1
##    text9     6      6         1
##   text10    10     13         3
##   text11    12     12         1
##   text12    39     45         4
##   text13    28     33         1
##   text14    12     12         2
##   text15     6      6         1
##   text16    38     39         1
```

```
##   text17     9     10         2
##   text18    21     23         1
##   text19    52     56         4
##   text20    31     35         1
##   text21     5      5         1
##   text22    18     20         2
##   text23    38     47         6
##   text24    12     12         1
##   text25    16     16         1
##   text26    39     44         2
##   text27    22     24         2
##   text28    41     50         3
##   text29    33     42         2
##   text30    32     36         1
##   text31    50     58         4
##   text32     6      6         1
##   text33    28     31         2
##   text34    31     35         1
##   text35     7      7         1
##   text36    10     10         2
##   text37    38     48         2
##   text38     6      6         1
##   text39    31     35         1
##   text40    10     10         1
##   text41     9      9         1
##   text42    29     34         2
##   text43    20     22         1
##   text44     9      9         1
##   text45    12     13         1
##   text46    31     32         1
##   text47    20     22         1
##   text48     9      9         1
##   text49    25     29         2
##   text50     4     12         2
##   text51    43     50         5
##   text52    10     10         1
##   text53    29     33         4
##   text54    17     17         1
##   text55    48     60         2
##   text56    22     26         3
##   text57    39     41         3
##   text58    16     16         1
##   text59    41     49         3
##   text60    34     36         2
##   text61    31     34         1
##   text62    13     13         1
##   text63    11     12         2
##   text64     4      4         1
##   text65    13     15         1
##   text66     5      5         2
##   text67    14     14         1
##   text68    22     23         1
##   text69    17     17         1
##   text70     9      9         1
```

```
##   text71    14     15         1
##   text72    13     13         1
##   text73     8      9         1
##   text74     7      8         1
##   text75    44     52         2
##   text76    21     27         4
##   text77    25     26         2
##   text78    10     10         1
##   text79    12     12         1
##   text80    12     15         1
##   text81    16     26         3
##   text82    10     13         1
##   text83    15     15         1
##   text84    20     30         3
##   text85    19     19         1
##   text86    42     55         5
##   text87    42     53         5
##   text88    35     36         1
##   text89    30     31         2
##   text90     9      9         1
##   text91     8      8         1
##   text92    19     30         3
##   text93    36     39         3
##   text94     9      9         1
##   text95    26     29         1
##   text96    45     51         4
##   text97    26     29         2
##   text98    36     43         4
##   text99    16     17         2
##  text100    15     19         2
```

```
tokens <- tokens(corpus) #tokenize the text so each doc (page, in this case) is a list of tokens (words

#examine the uncleaned version
tokens
```

```
## Tokens consisting of 56,398 documents.
## text1 :
##  [1] "RT"             "@Golden_Noonas" "Pdogg"          "really"
##  [5] "said"           "\""             "Let's"          "make"
##  [9] "better"         "albums"         "together"       "\""
## [ ... and 53 more ]
##
## text2 :
##  [1] "Wondering"    "about"        "if"           "breast"       "feeding"
##  [6] "is"           "possible"     "after"        "augmentation" "?"
## [11] "Learn"        "more"
## [ ... and 7 more ]
##
## text3 :
## [1] "@maggslin"               "Not"
## [3] "seeing"                  "the"
## [5] "relevance"               "https://t.co/cpwzVm09rn"
##
```

```
## text4 :
##  [1] "@SenatorCollins" "@AARP"           "@alzassociation" "@autismspeaks"
##  [5] "No"              "please"          "don't"           "do"
##  [9] "this"            "!"               "It"              "is"
## [ ... and 18 more ]
##
## text5 :
## [1] "Feeling" "like"    "the"     "one"     "."
##
## text6 :
##  [1] "RT"            "@Grammarly"    "@ryanmomalley" "That's"
##  [5] "great"         "to"            "hear"          "!"
##  [9] "Keep"          "up"            "the"           "awesome"
## [ ... and 4 more ]
##
## [ reached max_ndoc ... 56,392 more documents ]
```

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

tokens <- tokens_select(tokens, stopwords('english'),selection='remove') #stopwords lexicon built in to

#tokens <- tokens_wordstem(tokens) #stem words down to their base form for comparisons across tense and
# since we are doing visual analysis we are leaving this out but if we were doing rigorous analysis we

tokens <- tokens_tolower(tokens)
```

I don't think we need this

```
# hash_tweets <- tokens(corpus, remove_punct = TRUE) %>%
#                 tokens_keep(pattern = "#*")
#
# dfm_hash<- dfm(hash_tweets)
#
# tstat_freq <- textstat_frequency(dfm_hash, n = 100)
# head(tstat_freq, 10)
#
# #tidytext gives us tools to convert to tidy from non-tidy formats
# hash_tib<- tidy(dfm_hash)
#
# hash_tib %>%
#    count(term) %>%
#    with(wordcloud(term, n, max.words = 100))
```

Create the sparse matrix representation known as the document-feature matrix. quanteda's textstat_polarity function has multiple ways to combine polarity to a single score. The sent_logit value to fun argument is the log of (pos/neg) counts.

```
dfm <- dfm(tokens)

topfeatures(dfm, 12)
```

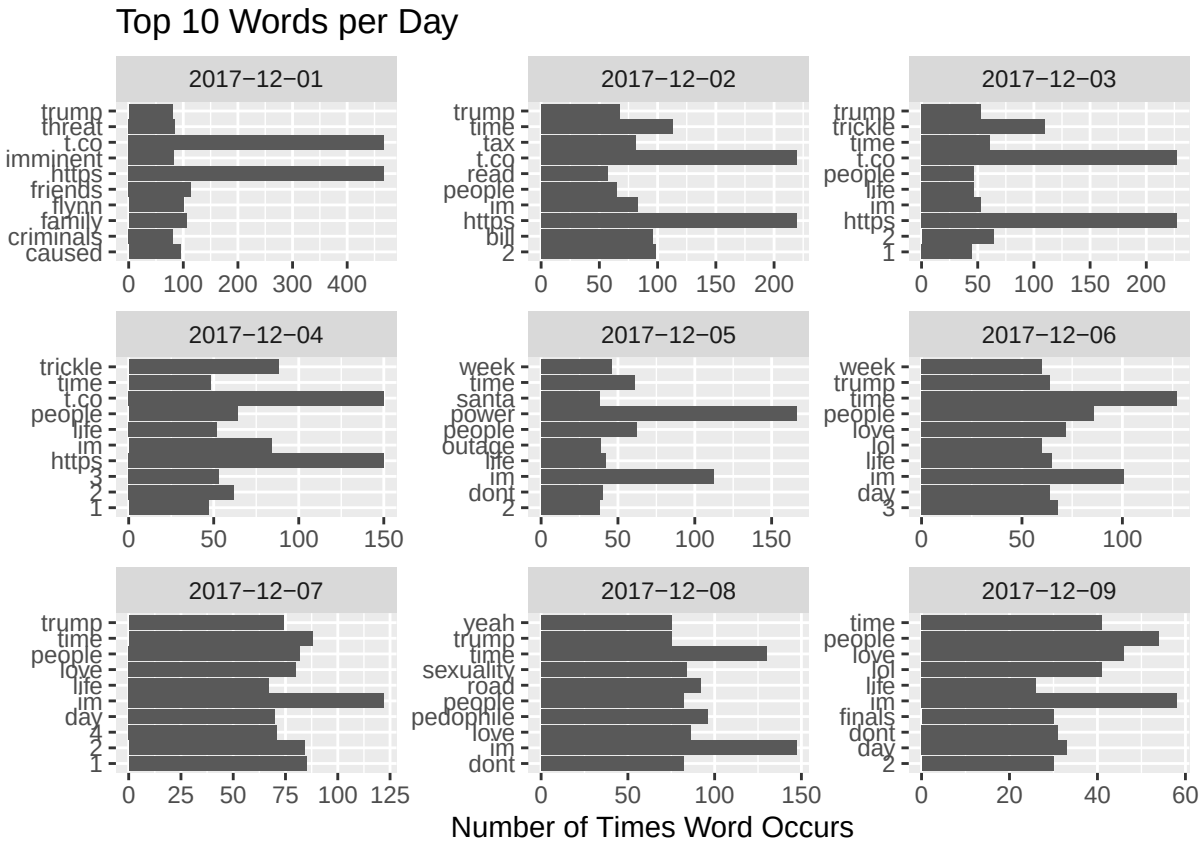
```
##     rt   just   like    get people    can    now    one  trump   time   love
##  34012   3730   3222   2602   2581   2495   2471   2366   2060   2044   1854
##   know
##   1787
```

```
dfm.sentiment <- dfm_lookup(dfm, dictionary = data_dictionary_LSD2015)

head(textstat_polarity(tokens, data_dictionary_LSD2015, fun = sent_logit))
```

```
##   doc_id sentiment
## 1  text1  1.466337
## 2  text2  1.098612
## 3  text3  0.000000
## 4  text4  1.098612
## 5  text5  1.098612
## 6  text6  1.609438
```

```
#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")  %>%
  filter(word != "rt")
```

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


ten_words %>%
  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")
```



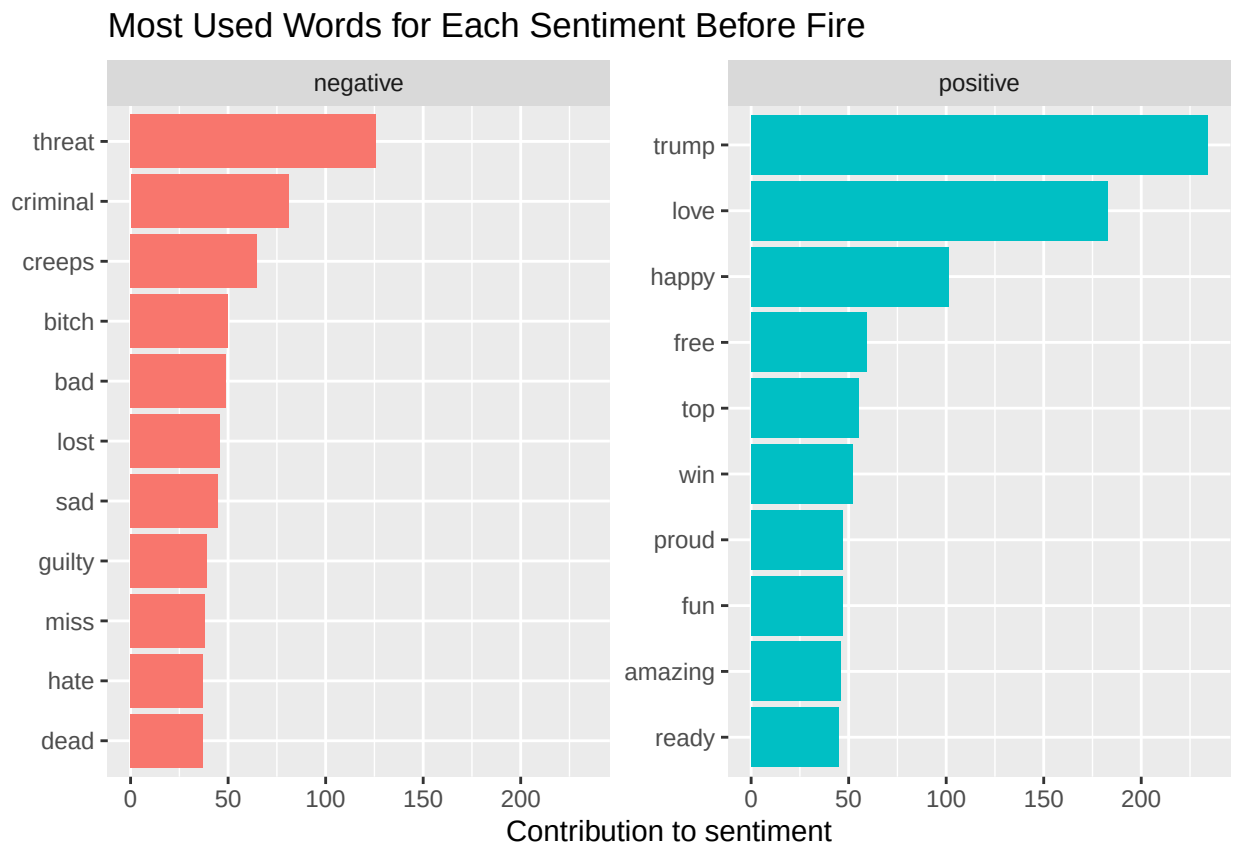
Make word groups

```
words_before <- words %>%
  filter(date < "2017-12-05")

words_during <- words %>%
  filter(date >= "2017-12-05")
```

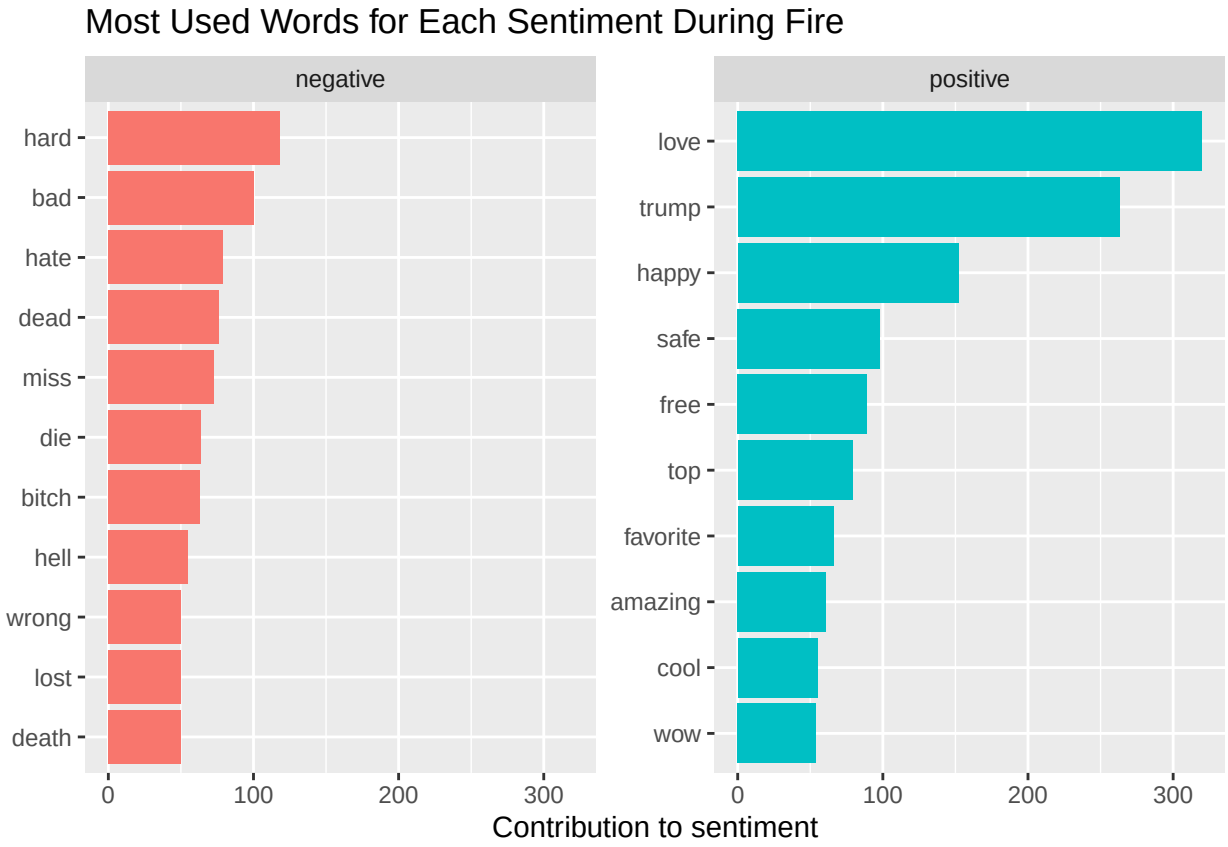
Make some charts of before and during

```
before_counts <- words_before %>%
  group_by(sentiment) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  filter(!is.na(sentiment)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
```

```
  labs(x = "Contribution to sentiment",
       y = NULL,
       title = "Most Used Words for Each Sentiment Before Fire")

before_counts
```

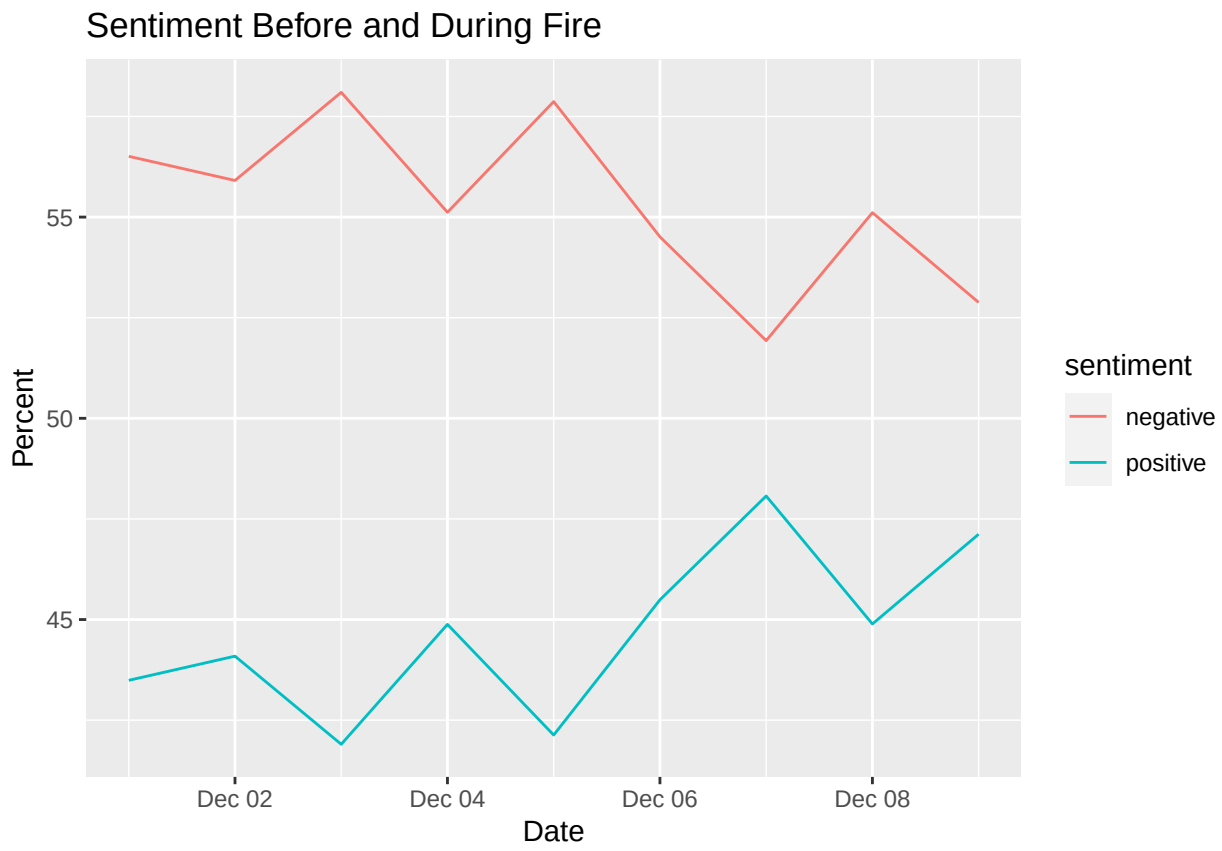



```
during_counts <- words_during %>%
  group_by(sentiment) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  filter(!is.na(sentiment)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL,
       title = "Most Used Words for Each Sentiment During Fire")

during_counts
```



Make plot that shows sentiment over time

```
sent_word_count <- words %>%
  group_by(date, sentiment) %>%
  count(sentiment) %>%
  ungroup() %>%
  filter(!is.na(sentiment)) %>%
  group_by(date) %>%
  mutate(n_max = sum(n),
         percent = round((n / n_max) * 100, 2))

ggplot(data = sent_word_count) +
  geom_line(aes(x = date, y = percent, color = sentiment)) +
  labs(title = "Sentiment Before and During Fire",
       y = "Percent",
       x = "Date")
```


Sentiment Before and During Fire
Percent
55
50
45
Dec 02
Dec 04
Dec 06
Dec 08
Date
sentiment
negative
positive