# assignment_2_sentiment_analysis1

Marie Rivers

4/14/2022

## To Do

- update to use pfas .docx file
- clean artifacts of the data collection process
- add additional stop words

## Sentiment Analysis I

This assignment uses sentiment analysis to... xxx

```
library(tidyr) #text analysis in R
library(lubridate) #working with date data
library(pdftools) #read in pdfs
library(tidyverse)
library(tidytext)
library(here)
library(LexisNexisTools) #Nexis Uni data wrangling
library(sentimentr)
library(readr)
```

**Using the "IPCC" Nexis Uni data set from the class presentation and the pseudo code we discussed, recreate Figure 1A from Froelich et al. (Date x # of 1) positive, 2) negative, 3) neutral headlines):**

```
data_folder <- "/Users/marierivers/Documents/UCSB_Environmental_Data_Science/EDS_231_Text_and_Sentiment_
my_files <- list.files(pattern = ".docx", path = data_folder,
                       full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat <- lnt_read(my_files) #Object of class 'LNT output'
# lnt_read = read in a LexisNexis file
```

```
meta_df <- dat@meta
articles_df <- dat@articles
paragraphs_df <- dat@paragraphs
```

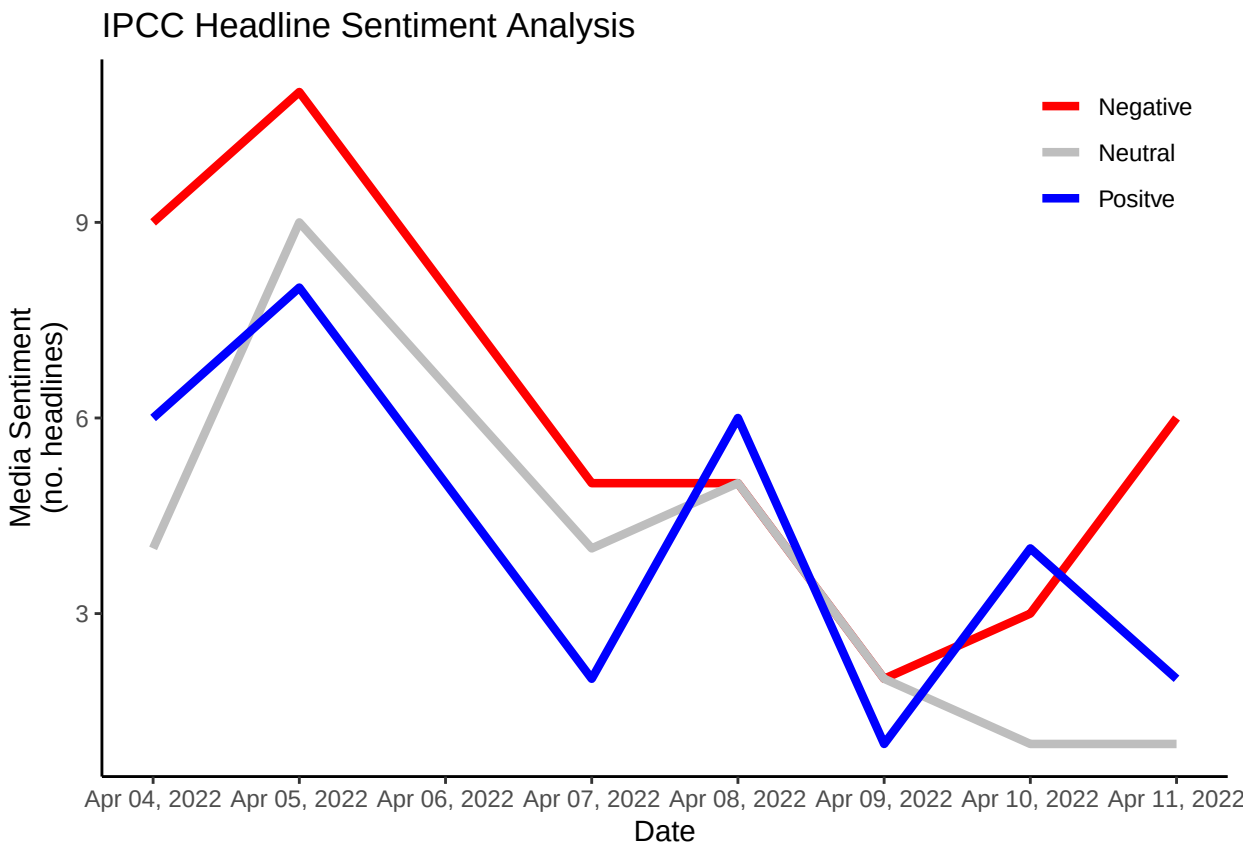
```
dat2<- data_frame(element_id = seq(1:length(meta_df$Headline)), Date = meta_df$Date, Headline = meta_df$
paragraphs_dat <- data_frame(element_id = paragraphs_df$Art_ID, Text  = paragraphs_df$Paragraph)
dat3 <- inner_join(dat2, paragraphs_dat, by = "element_id")
```

```
mytext <- get_sentences(dat2$Headline)
sent <- sentiment(mytext)

sent_df <- inner_join(dat2, sent, by = "element_id")

sentiment <- sentiment_by(sent_df$Headline)

sent_df <- sent_df%>%
  arrange(sentiment)
```

```
sent_df_summary <- sent_df %>%
  mutate(sent_category = case_when(
    sentiment < 0 ~ "negative",
    sentiment > 0 ~ "positive",
    sentiment == 0 ~ "neutral")) %>%
  group_by(Date, sent_category) %>%
  summarise(num_headlines = n())
```

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

```
ggplot(data = sent_df_summary, aes(x = Date, y = num_headlines)) +
  geom_line(aes(color = sent_category), size = 1.5) +
  scale_color_manual(name = "",
                     values = c("red", "gray", "blue"),
                     labels = c ("Negative", "Neutral", "Positve")) +
  labs(title = "IPCC Headline Sentiment Analysis",
       y = "Media Sentiment\n(no. headlines)") +
  theme_classic() +
  theme(legend.position = c(0.9, 0.9),
        legend.background = element_blank()) +
  scale_x_date(date_labels = "%b %d, %Y",
               limits = c(as.Date("2022-04-04"), as.Date("2022-04-11")),
               breaks = "1 day")
```




# Nexis Uni search of the ter, 'pfas'

```
data_folder_pfas <- "/Users/marierivers/Documents/UCSB_Environmental_Data_Science/EDS_231_Text_and_Senti
my_files_pfas <- list.files(pattern = ".docx", path = data_folder_pfas,
                      full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat_pfas <- lnt_read(my_files_pfas) #Object of class 'LNT output'
# lnt_read = read in a LexisNexis file
```

```
meta_pfas_df <- dat_pfas@meta
articles_pfas_df <- dat_pfas@articles
paragraphs_pfas_df <- dat_pfas@paragraphs
```

```
dat2_pfas<- data_frame(element_id = seq(1:length(meta_pfas_df$Headline)), Date = meta_pfas_df$Date, Hea

paragraphs_dat_pfas <- data_frame(element_id = paragraphs_pfas_df$Art_ID, Text  = paragraphs_pfas_df$Pa

dat3_pfas <- inner_join(dat2_pfas,paragraphs_dat_pfas, by = "element_id")
```

## NRC lexicon

```
nrc_sent <- get_sentiments('nrc') %>%
  filter(!sentiment %in% c("positive", "negative"))
```

```r
pfas_text_words <- dat3_pfas %>%
  unnest_tokens(output = word, input = Text, token = 'words') %>%
  anti_join(stop_words, by = 'word')
```

```r
#xxx
custom_stop_words <- bind_rows(tibble(word = c("your_word"),
                                      lexicon = c("custom")),
                               stop_words)
```

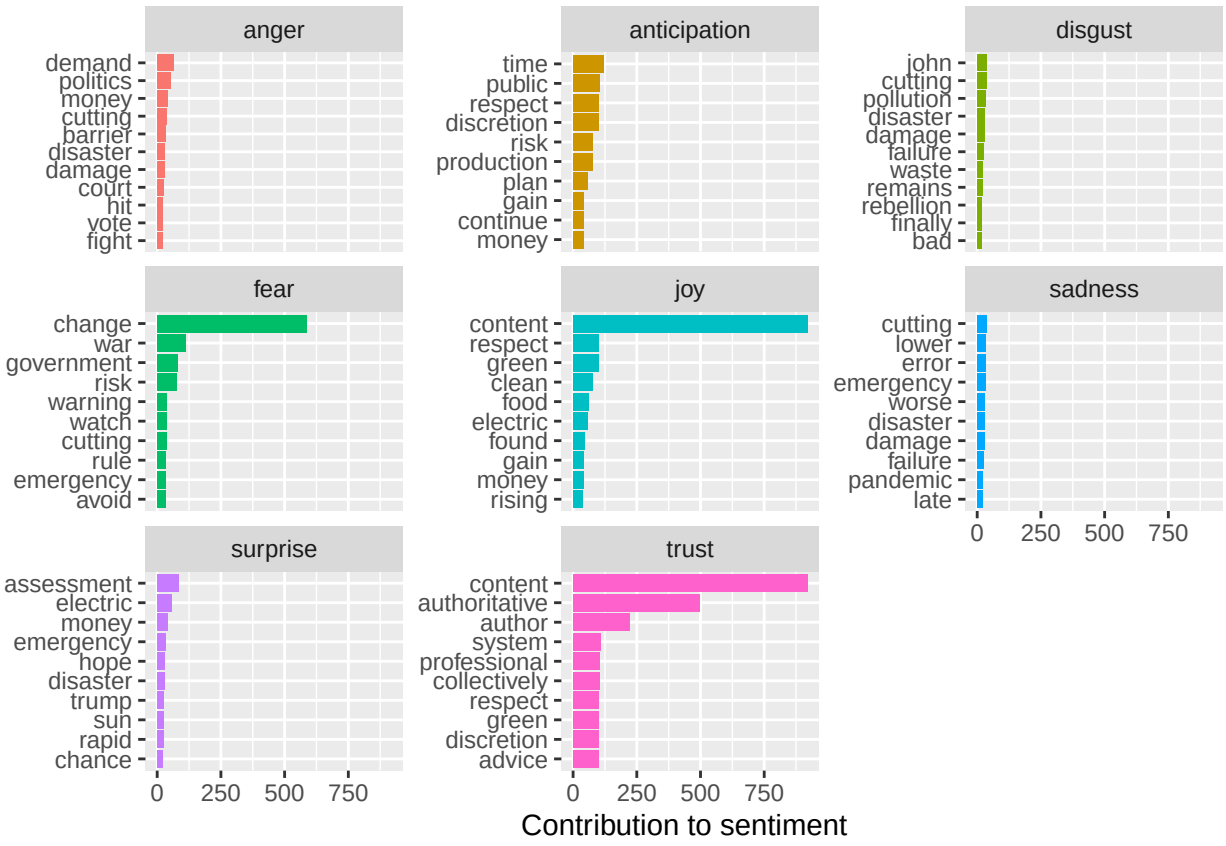
```r
pfas_nrc_sent <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

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

```r
pfas_sent_counts <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  #group_by(Date, sentiment) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

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

```r
pfas_sent_plot1 <- pfas_sent_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ggplot(aes(x = n, y = reorder(word, n), fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
pfas_sent_plot1
```




```
words_per_day <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  group_by(Date) %>%
  count(name = "words_per_day")
```

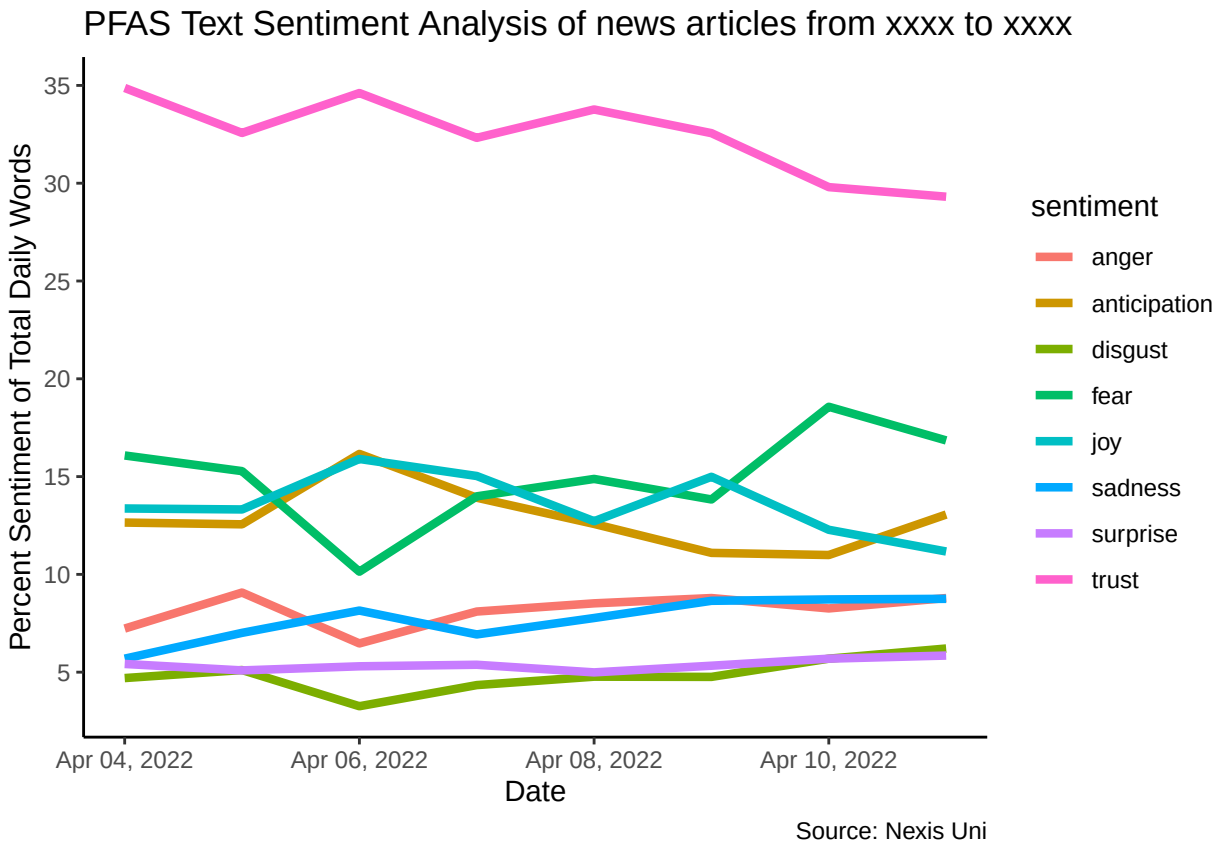
```
## Joining, by = "word"
```

```
pfas_sent_percent <- pfas_text_words %>%
  inner_join(nrc_sent) %>%
  count(Date, sentiment) %>%
  left_join(words_per_day, by = "Date") %>%
  mutate(percent = round(((n / words_per_day) * 100), 2))
```

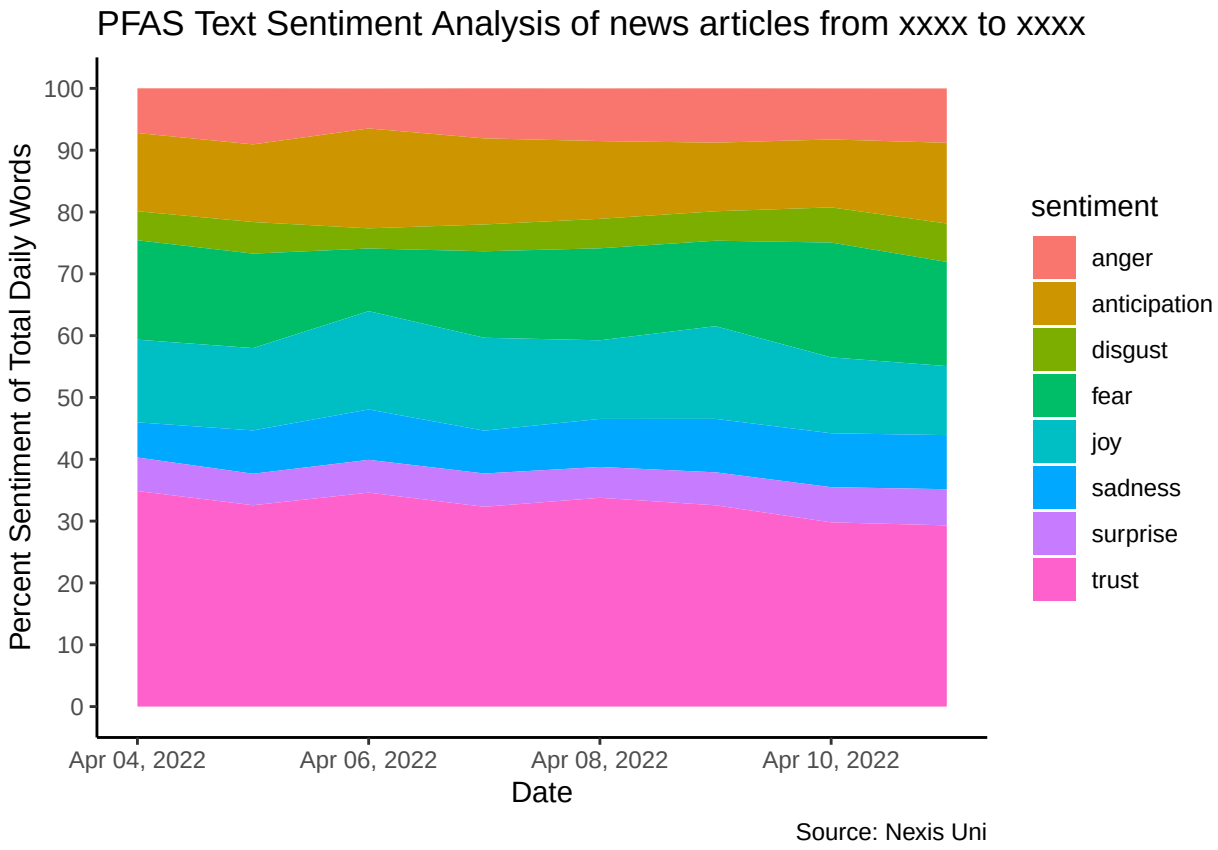
```
## Joining, by = "word"
```

```
pfas_sent_plot2 <- ggplot(data = pfas_sent_percent, aes(x = Date, y = percent)) +
  geom_line(aes(color = sentiment), size = 1.5) +
  #scale_color_manual(name = "", values = c("red", "gray", "blue"), labels = c ("Negative", "Neutral",
  labs(title = "PFAS Text Sentiment Analysis of news articles from xxxx to xxxx",
       caption = "Source: Nexis Uni",
       y = "Percent Sentiment of Total Daily Words") +
  theme_classic() +
  scale_x_date(date_labels = "%b %d, %Y",
               limits = c(as.Date("2022-04-04"), as.Date("2022-04-11")),
               breaks = "2 day") +
```

```
  scale_y_continuous(breaks = seq(0, 100, by = 5))
pfas_sent_plot2
```




```
pfas_sent_plot3 <- ggplot(data = pfas_sent_percent, aes(x = Date, y = percent, fill = sentiment)) +
  geom_area() +
  labs(title = "PFAS Text Sentiment Analysis of news articles from xxxx to xxxx",
       caption = "Source: Nexis Uni",
       y = "Percent Sentiment of Total Daily Words") +
  theme_classic() +
  scale_x_date(date_labels = "%b %d, %Y",
               limits = c(as.Date("2022-04-04"), as.Date("2022-04-11")),
               breaks = "2 day") +
  scale_y_continuous(breaks = seq(0, 100, by = 10))
pfas_sent_plot3
```


PFAS Text Sentiment Analysis of news articles from xxxx to xxxx
Percent Sentiment of Total Daily Words
100
90
80
70
60
50
40
30
20
10
0
Apr 04, 2022
Apr 06, 2022
Apr 08, 2022
Apr 10, 2022
Date
sentiment
anger
anticipation
disgust
fear
joy
sadness
surprise
trust
Source: Nexis Uni