Week 3 Lab: Sentiment Analysis I

Assignment

0. 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):
1. Access the Nexis Uni database through the UCSB library: https://www.library.ucsb.edu/research/db/211
2. Choose a key search term or terms to define a set of articles.
3. Use your search term along with appropriate filters to obtain and download a batch of at least 100 full text search results (.docx).
4. Read your Nexis article document into RStudio and parse with pdftools.
5. This time use the full text of the articles for the analysis. First clean any artifacts of the data collection process (hint: this type of thing should be removed: "Apr 04, 2022(Biofuels Digest: http://www.biofuelsdigest.com/ Delivered by Newstex"))
6. Explore your data a bit and try to replicate some of the analyses above presented in class if you'd like (not necessary).
7. Plot the amount of emotion words (the 8 from nrc) as a percentage of all the emotion words used each day (aggregate text from articles published on the same day). How does the distribution of emotion words change over time? Can you think of any reason this would be the case?

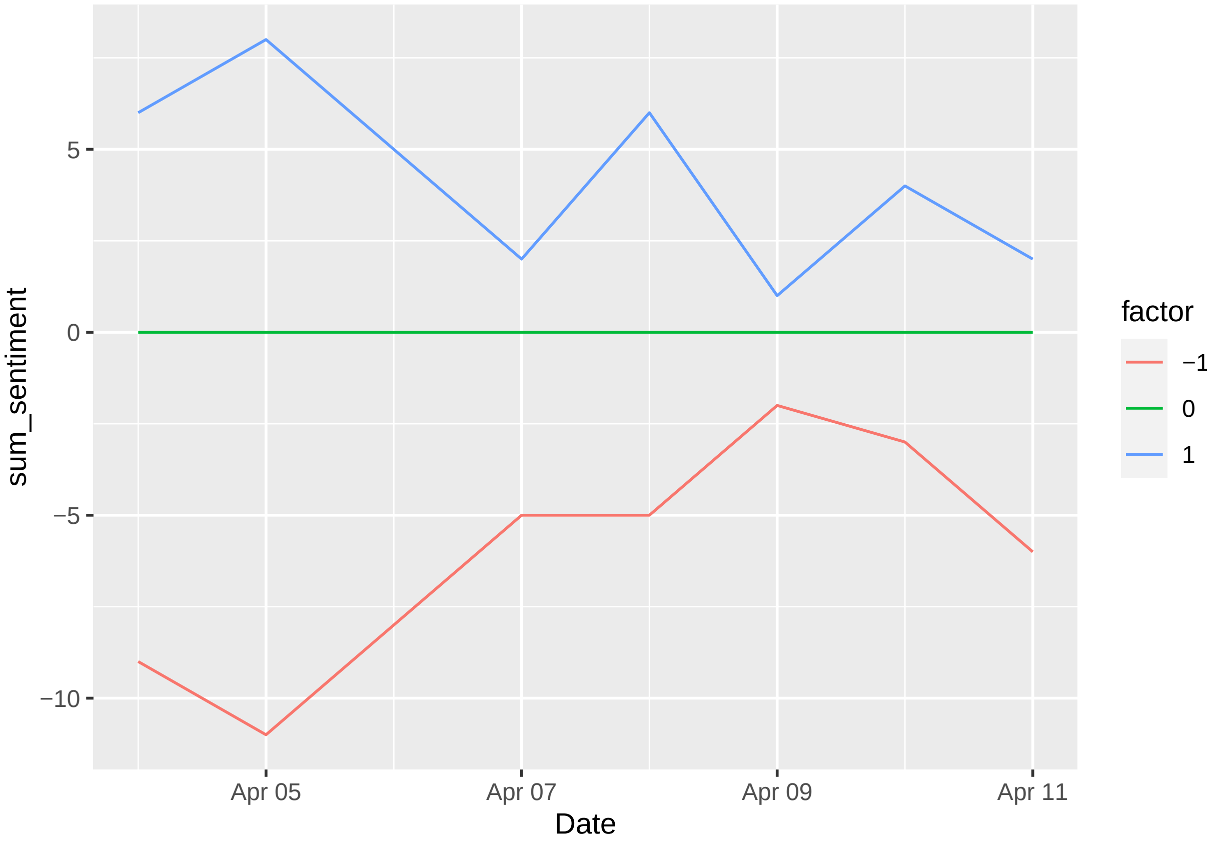
```r
library(tidyr) #text analysis in R
library(lubridate) #working with date data
library(pdftools) #read in pdfs
library(tidyverse)
library(tidytext)
library(here)
library(LexisNexisTools) #Nexis Uni data wrangling
library(sentimentr)
library(readr)
```

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):

```r
my_files <- list.files(pattern = "IPCC", path = here(),
                       full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat <- lnt_read(my_files) #Object of class 'LNT output'
```

```r
# lets make the tibbles in the LNToutput into individual df

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$
```

```r
IPCC_sent <- sentiment(get_sentences(dat2$Headline)) # use sentimentr get sentiment of each headline

IPCC_sent <-  inner_join(x = dat2,
                         y = IPCC_sent,
                         by = "element_id") %>% # join back with dat2
  as_tibble() %>% # make tibble for tidyverse
    mutate(category = case_when( # make score
        sentiment < 0 ~ "-1",
        sentiment == 0 ~ '0',
        sentiment > 0 ~ '1'),
        category = as.integer(category),
        factor = as.factor(category)) %>%
  group_by(Date, category, factor) %>%
  summarise(sum_sentiment = sum(category)) %>%
  ggplot(aes(x = Date,
             y = sum_sentiment,
             color = factor)) +
  geom_line(position = "dodge")

IPCC_sent
```



This is not quite it but this is taking to much time so I will move on.

Set up Sentiment Words

```
nrc_sent <- get_sentiments('nrc') #requires downloading a large dataset via prompt

nrc_fear <- get_sentiments("nrc") %>%
  filter(sentiment == "fear")
```

Key Search Term = decarbonization

- Search term = decarbonization
- Narrow by: news and English language

Download Batch of data and Read in

```
my_files <- list.files(pattern = ".docx", path = here("data/decarbonization/"),
                       full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

dat <- lnt_read(my_files) #Object of class 'LNT output'

# lets make the tibbles in the LNToutput into individual df
```

```
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")
```

Use the full text of the articles for the analysis. First clean any artifacts of the data collection process (hint: this type of thing should be removed: "Apr 04, 2022(Biofuels Digest: http://www.biofuelsdigest.com/ Delivered by Newstex"))

```
#this is get rid of any links and anything that contains less than 20 words
cleanpars <- dat3 %>%
  mutate(link = str_detect(dat3$Text, "http", negate = TRUE)) %>%
  filter(link == TRUE & nchar(dat3$Text) > 20)
```

Explore your data a bit and try to replicate some of the analyses above presented in class if you'd like (not necessary).

```
#most common words by sentiment
fear_words <- cleanpars  %>%
  unnest_tokens(output = word, input = Text, token = 'words') %>%
  inner_join(nrc_fear) %>%
  count(word, sort = TRUE)
head(fear_words)
```

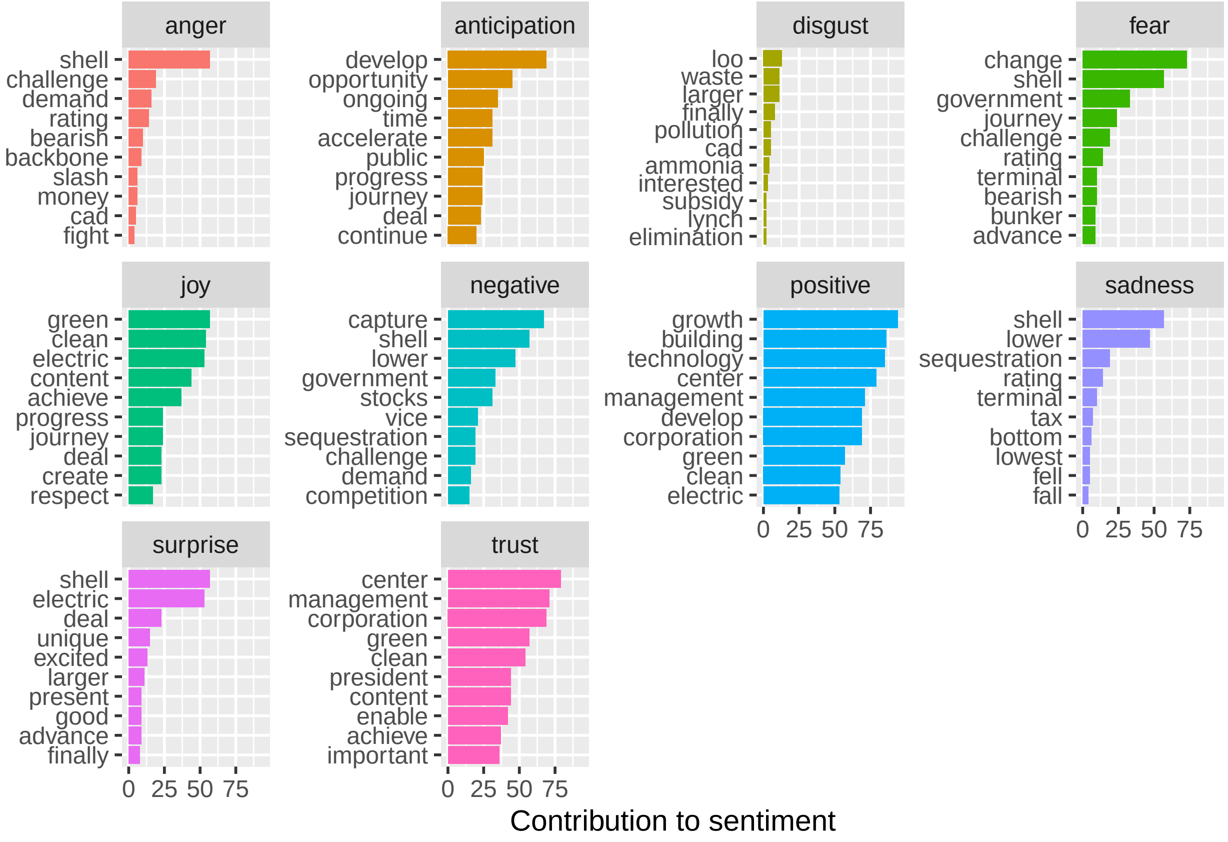
```
## # A tibble: 6 x 2
##   word           n
##   <chr>      <int>
## 1 change        73
## 2 shell         57
## 3 government    33
## 4 journey       24
## 5 challenge     19
## 6 rating        14
```

```
#unnest to word-level tokens, remove stop words, and join sentiment words
text_words <- cleanpars  %>%
  unnest_tokens(output = word, input = Text, token = 'words')

nrc_word_counts <- text_words %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()

head(nrc_word_counts)
```

```
## # A tibble: 6 x 3
##   word       sentiment     n
##   <chr>      <chr>     <int>
## 1 growth     positive     94
## 2 building   positive     86
## 3 technology positive     85
## 4 center     positive     79
## 5 center     trust        79
## 6 change     fear         73
```

Let's break it out and plot the contributions by particular words different sentiment categories

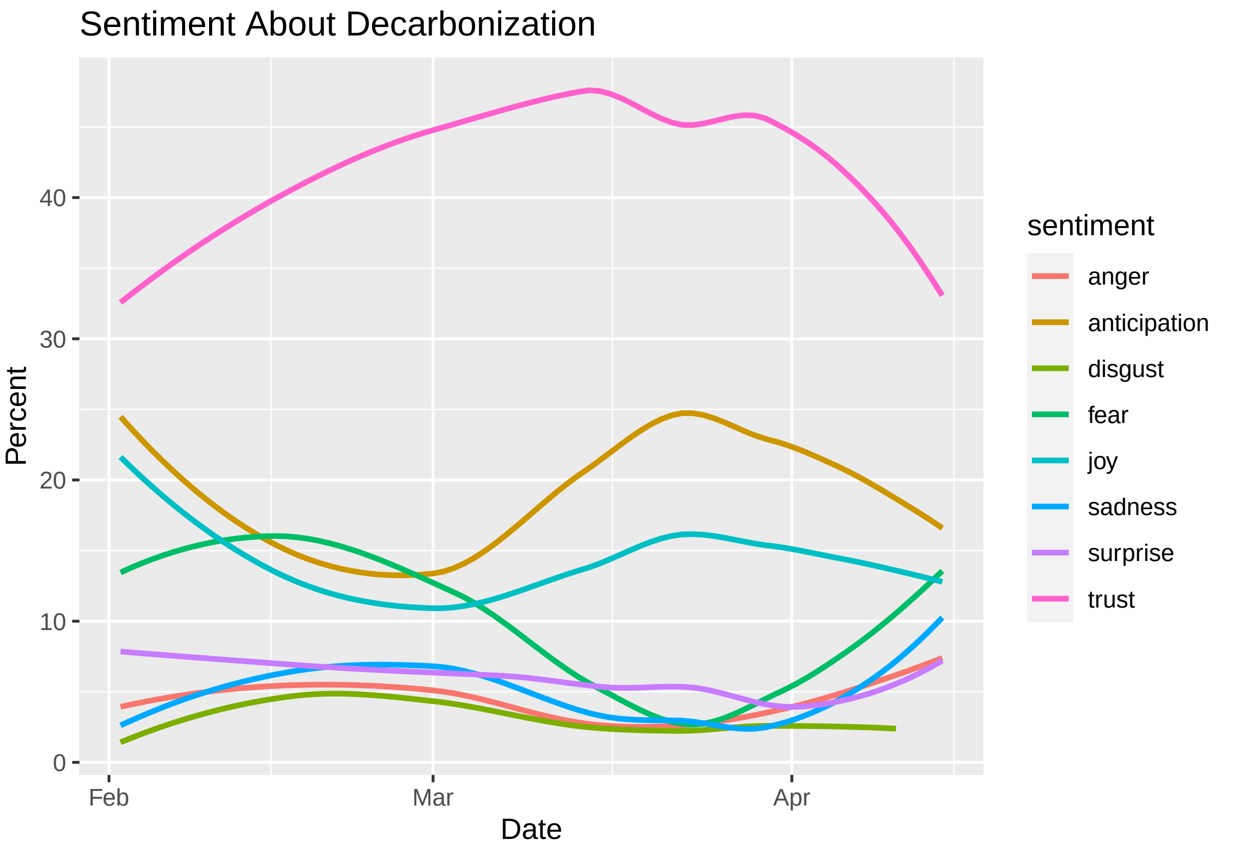
```
decarb_sent_counts <- text_words %>%
        group_by(element_id) %>%
        inner_join(get_sentiments("nrc")) %>%
        group_by(sentiment) %>%
        count(word, sentiment, sort = TRUE) %>%
        ungroup()

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



Plot the amount of emotion words (the 8 from nrc) as a percentage of all the emotion words used each day (aggregate text from articles published on the same day). How does the distribution of emotion words change over time? Can you think of any reason this would be the case?

```
# need to reset because things have gotten messy

sent_words <- text_words %>% #break into individual words again
  anti_join(stop_words, by = 'word') %>%
  inner_join(nrc_sent, by = 'word') %>%
  filter(!sentiment %in% c("positive", "negative")) %>%
  mutate(Date = as_date(Date))

sent_word_count <- sent_words %>%
  group_by(Date, sentiment) %>%
  count(sentiment) %>%
  ungroup() %>%
  group_by(Date) %>%
  mutate(n_max = sum(n),
         percent = round((n / n_max) * 100, 2))
```

```
ggplot(data = sent_word_count) +
  geom_smooth(aes(x = Date, y = percent, color = sentiment),
              se = FALSE) +
  labs(title = "Sentiment About Decarbonization",
       y = "Percent",
       x = "Date")
```



There is an increase in anticipation words at the beginning of March so maybe there was some legislation in the works about decarbonization strategies. There is a high percentage of trust words likely due to the fact that decarbonization takes a lot of cooperation of government bodies. There is an interesting back and forth between fear and joy at opposite times and I wonder if that coincides with IPCC report releases.