EDS 231: Assignment 2

Scout Leonard

04/18/2022

Objective

Load Libararies

```r
# load packages
packages=c("tidyr",
           "lubridate",
           "pdftools",
           "pdftools",
           "tidytext",
           "here",
           "LexisNexisTools",
           "sentimentr",
           "readr",
           "textdata",
           "dplyr",
           "stringr",
           "janitor",
           "ggplot2",
           "MetBrewer",
           "kableExtra")

for (i in packages) {
  if (require(i,character.only=TRUE)==FALSE) {
    install.packages(i,repos='http://cran.us.r-project.org')
  }
  else {
    require(i,character.only=TRUE)
  }
}
```

Part 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):

```
#to follow along with this example, download this .docx to your working directory:
#https://github.com/MaRo406/EDS_231-text-sentiment/blob/main/nexis_dat/Nexis_IPCC_Results.docx
ipcc_files <- list.files(pattern = ".docx", path = here("data/ipcc"),
                         full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

ipcc_dat <- lnt_read(ipcc_files) #Object of class 'LNT output'

ipcc_meta_df <- ipcc_dat@meta
ipcc_articles_df <- ipcc_dat@articles
ipcc_paragraphs_df <- ipcc_dat@paragraphs

ipcc_dat2<- data_frame(element_id = seq(1:length(ipcc_meta_df$Headline)),
                       Date = ipcc_meta_df$Date,
                       Headline = ipcc_meta_df$Headline)
```

Use the Bing sentiment analysis lexicon.

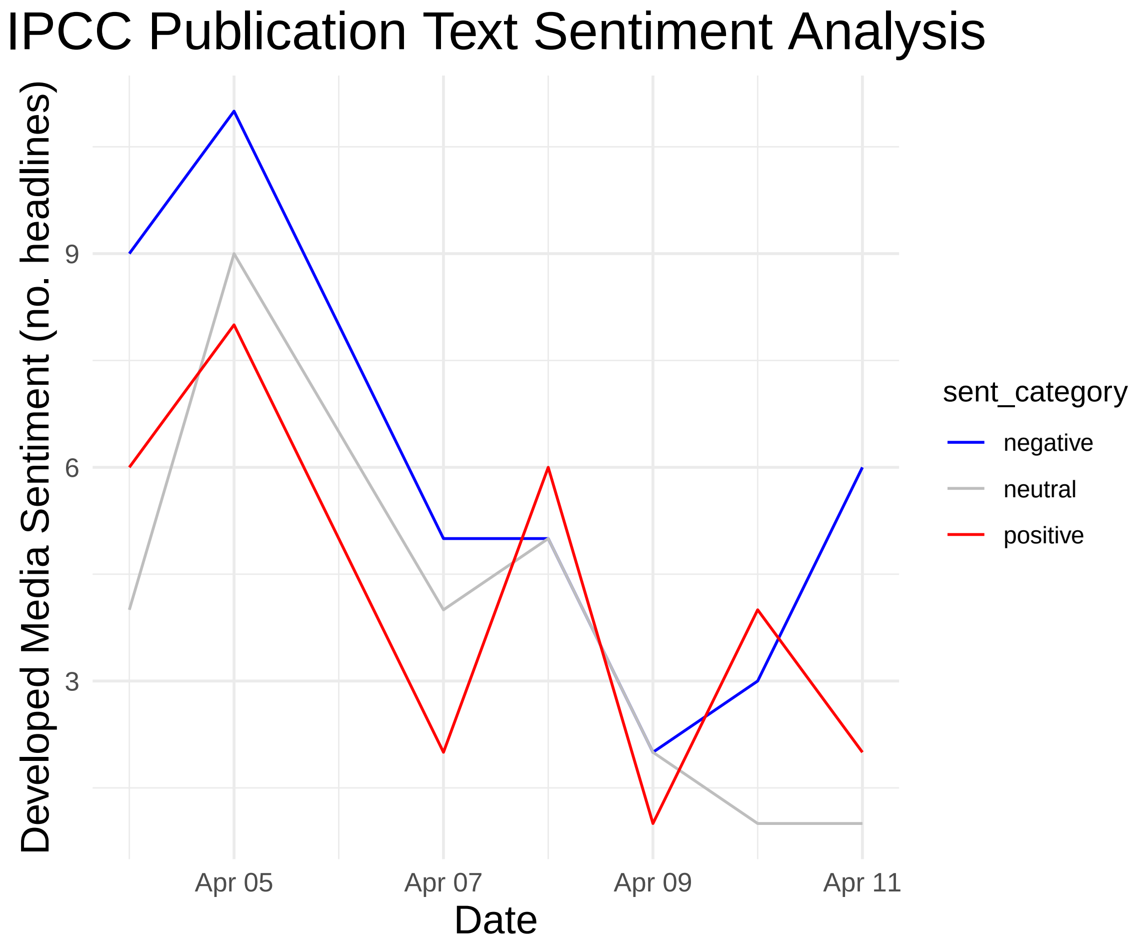
```
bing_sent <- get_sentiments('bing') #grab the bing sentiment lexicon from tidytext

#test
head(bing_sent, n = 5) %>%
  kable()
```

word	sentiment
2-faces	negative
abnormal	negative
abolish	negative
abominable	negative
abominably	negative

```
#get sentences from ipcc headlines
ipcc_text <- get_sentences(ipcc_dat2$Headline)

#get the sentiment of each sentence in each headline
ipcc_sent <- sentiment(ipcc_text)

#join with sentence data
ipcc_sent <- inner_join(ipcc_dat2, ipcc_sent, by = "element_id")

# catgorize for ggplot
ipcc_sent <- ipcc_sent %>%
  mutate(sent_category = case_when(
    sentiment < 0 ~ "negative",
    sentiment == 0 ~ "neutral",
    sentiment > 0 ~ "positive"
  ))

head(ipcc_sent, 5) %>%
  kable()
```

element_id	Date	Headline
1	2022-04-05	IPCC says it’s not too late to avoid climate catastrophe
2	2022-04-08	IPCC Adaptation Report: It’s Later Than You Think
3	2022-04-07	PRESS RELEASE: Heartland Institute Climate Experts React to Latest UN IPCC Report
4	2022-04-07	’Now or never’ to keep within 1.5C, reports IPCC
5	2022-04-06	ExxonMobil Announces $10 Billion Oil Investment the Same Day IPCC Signals End for Foss

```
#generate counts of sentiment headlines by date to plot
ipcc_sent_plot <- ipcc_sent %>%
  count(sent_category, Date)

#plot it UP
ggplot(data = ipcc_sent_plot, aes(x = Date, y = n)) +
         geom_line(aes(color = sent_category)) +
  theme_minimal() +
  labs(y = "Developed Media Sentiment (no. headlines)",
       x = "Date",
       title = "IPCC Publication Text Sentiment Analysis") +
  scale_color_manual(values = c("blue", "grey", "red")) +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.title = element_text(size = 15),
        axis.text = element_text(size = 10))
```



Part 1

[Access the Nexis Uni database through the UCSB library] (https://www.library.ucsb.edu/research/db/211)

Got it!

Part 2

Choose a key search term or terms to define a set of articles.

Done! I chose the term, “school lunch.” My MEDS cohort knows I love talking about the USDA National School Lunch Program…

Part 3

Use your search term along with appropriate filters to obtain and download a batch of at least 100 full text search results (.docx)..

Sweet! All downloaded.

Part 4

Read your Nexis article document into RStudio.

Now for some coding…

```r
#read in my Lexis Nexis files
lunch_files <- list.files(pattern = ".docx",
                          path = here("data/lunch"),
                          full.names = TRUE,
                          recursive = TRUE,
                          ignore.case = TRUE)

lunch_dat <- lnt_read(lunch_files) #Object of class 'LNT output'

#pull the metadata, articles, and text
lunch_meta_df <- lunch_dat@meta
lunch_articles_df <- lunch_dat@articles
lunch_paragraphs_df <- lunch_dat@paragraphs

#make a df with headlines by date
lunch_dat2<- data.frame(element_id = seq(1:length(lunch_meta_df$Headline)),
                        Date = lunch_meta_df$Date,
                        Headline = lunch_meta_df$Headline)

#test
head(lunch_dat2, 5) %>%
  kable()
```

element_id	Date	Headline
1	2022-03-24	SCHOOL LUNCH & BREAKFAST
2	2022-04-07	SCHOOL LUNCH MENUS
3	2022-04-13	Monroe School Board Learns School Lunch Program Details
4	2022-02-11	Healthy School Lunch Programme
5	2022-03-10	SCHOOL LUNCH & Breakfast Menus

Part 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”)).

```
lunch_paragraphs_dat <- data.frame(element_id = lunch_paragraphs_df$Art_ID,
                                   Text = lunch_paragraphs_df$Paragraph)

lunch_dat3 <- inner_join(lunch_dat2,
                     lunch_paragraphs_dat,
                     by = "element_id") %>%
  clean_names()

#unnest to word-level tokens, remove stop words, and join sentiment words
 lunch_text_words <- lunch_dat3  %>%
  unnest_tokens(output = word,
                input = text,
                token = 'words')

lunch_text_words <- lunch_text_words %>%
  anti_join(stop_words) #removes the stop words

 #remove numbers
clean_lunch_words <- str_remove_all(lunch_text_words$word, "[:digit:]")

#removes apostrophes
clean_lunch_words <- gsub("’s", '', clean_lunch_words)

lunch_text_words$clean <- clean_lunch_words

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

#reassign
lunch_words_tokenized <- tib

#test
head(lunch_words_tokenized) %>%
  kable()
```

element_id	date	headline	word	clean
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	breakfast	breakfast
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	lunch	lunch
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	menu	menu
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	palmyra	palmyra
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	school	school
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	district	district

Part 6

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

Part 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
nrc_sent <- get_sentiments('nrc') %>%
  filter(sentiment != "negative" & sentiment != "positive")

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

#test
head(text_words, 5) %>%
   kable()
```

element_id	date	headline	word
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	breakfast
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	lunch
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	menu
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	palmyra
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	school

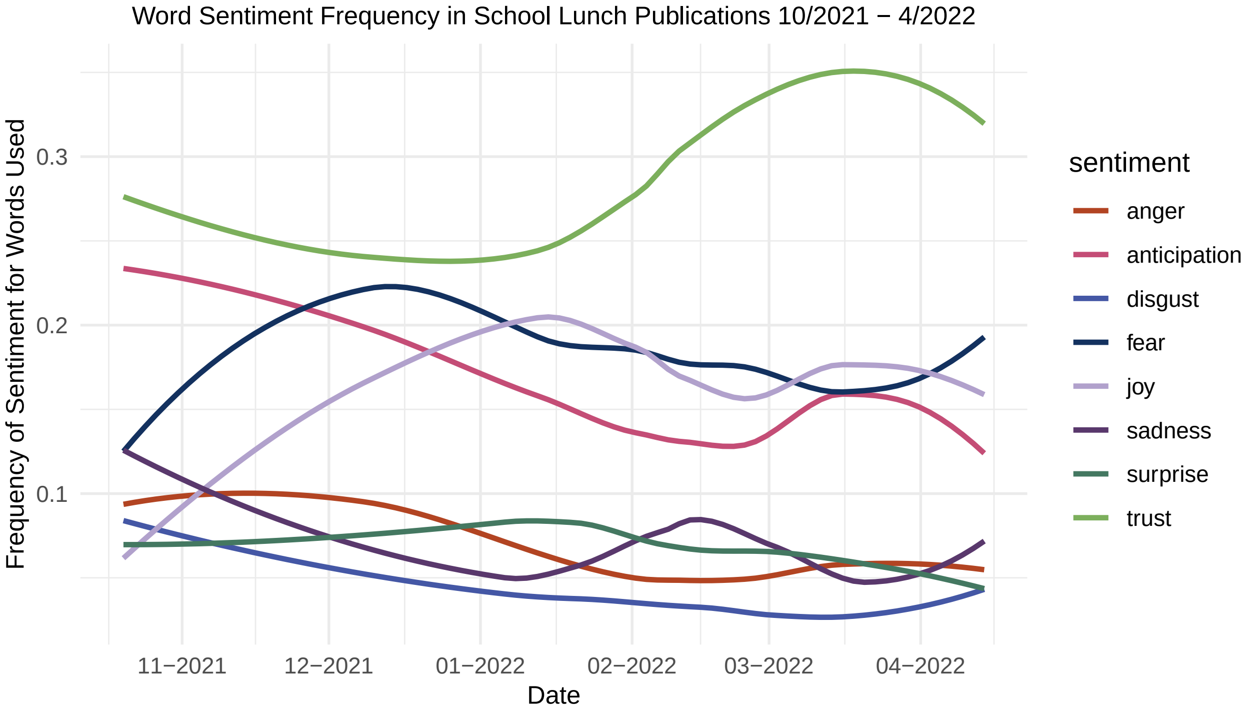
```
lunch_nrc_word_counts <- text_words %>%
  inner_join(nrc_sent)

#test
head(lunch_nrc_word_counts, 5) %>%
  kable()
```

element_id	date	headline	word	sentiment
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	school	trust
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	change	fear
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	understanding	trust
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	chocolate	anticipation
1	2022-03-24	SCHOOL LUNCH & BREAKFAST	chocolate	joy

```
lunch_sentiment_freq <- lunch_nrc_word_counts %>%
  group_by(date, sentiment) %>%
  summarise(count = n()) %>%
  mutate(freq = formattable::percent(count / sum(count)))

#head
head(lunch_sentiment_freq, 5) %>%
  kable()
```

date	sentiment	count	freq
2021-10-20	anger	3	8.33%
2021-10-20	anticipation	7	19.44%
2021-10-20	disgust	3	8.33%
2021-10-20	fear	5	13.89%
2021-10-20	joy	2	5.56%

```
ggplot(data = lunch_sentiment_freq, aes(x = date, y = freq)) +
  # geom_point(aes(color = sentiment)) +
  geom_smooth(aes(color = sentiment), se = FALSE) +
  scale_x_date(date_breaks = "1 month", date_labels = "%m-%Y") +
  scale_color_manual(values = met.brewer("Thomas")) +
  theme_minimal() +
  labs(title = "Word Sentiment Frequency in School Lunch Publications 10/2021 - 4/2022",
       x = "Date",
       y = "Frequency of Sentiment for Words Used") +
  theme(axis.title = element_text(size = 10),
        plot.title = element_text(size = 10, hjust = 0.5))
```



It seems that there are just fewer sentiments and less change in frequencies from day to day in December. I think this is because, while I was looking for articles and publications about the national school lunch program and access to nutrition for kids, Lexis Nexis returned a lot of school meal menus for random districts in the United States. I think the frequency of words in general decreased in December due to school breaks for the winter holidays. The frequency of menus in my data is kind of a bummer, but it is also interesting to see that so many food words are associated with trust. I think a lot of child ed words are, as well.