Homework 3, Sentiment Analysis

Steven Cognac

4/19/2022

Part A.

Re-create plot from Fig. 1 in the Froelich et al. 2017 paper titled "Public Perceptions of Aquaculture: Evaluating Spatiotemporal Patterns of Sentiment around the World".

Read in IPCC File

```
# read in the .docx file
ipcc_file <- here("Nexis_IPCC_Results.docx")
```

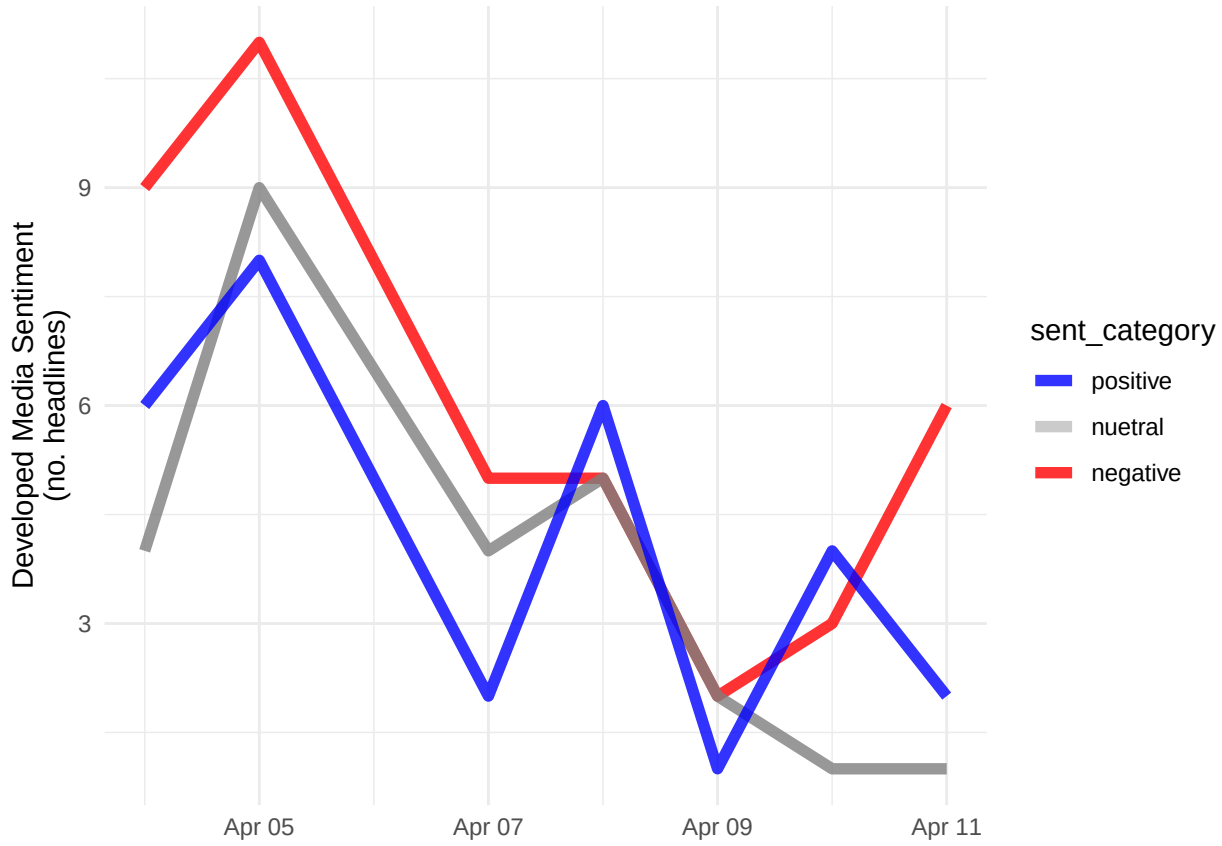
Create datafrome from IPCC data

```
ipcc_meta <- ipcc@meta
ipcc_df <- data_frame(
  element_id = seq(1:length(ipcc_meta$Headline)),
  Date = ipcc_meta$Date,
  Headline = ipcc_meta$Headline)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
bing_sent <- get_sentiments('bing')

ipcc_sent <- ipcc_df$Headline %>%
  get_sentences() %>%
  sentiment() %>%
  inner_join(ipcc_df, by = "element_id") %>%
  mutate(
    sent_category = case_when(
      sentiment < 0 ~ "negative",
      sentiment > 0 ~ "positive",
      T ~ "neutral")) %>%
  count(sent_category, Date)
```

```
ggplot(data = ipcc_sent, aes(x = Date, y = n, color = sent_category)) +
  geom_line(size = 2, alpha = 0.8) +
  labs(y = "Developed Media Sentiment\n(no. headlines)",
       x = NULL) +
  scale_color_manual(values = c(
    "positive" = "blue",
    "nuetral" = "grey",
    "negative" = "red")) +
  theme_minimal()
```



Part B. Nexis Uni Database Search

Import files

- keyword search 'inaturalist'
- iNaturalist is a nature app that helps a user identify plants and animals through the camera on their smartphone.

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

Let's look at some of the data of our articles.

```
# isolating matedata, articles, and paragraphs
meta_df <- dat@meta
articles_df <- dat@articles
paragraphs_df <- dat@paragraphs

# put the meta_df, articles_df, and paragraphs_df into a dataframe
dat2 <- tibble(element_id = seq(1:length(meta_df$Headline)),
                  Date = meta_df$Date,
                  Headline = meta_df$Headline)

# May be of use for assignment: using the full text from the articles
paragraphs_dat <- data_frame(element_id = paragraphs_df$Art_ID,
                             Text  = paragraphs_df$Paragraph)

# join dat2 with paragraphs_df
dat3 <- inner_join(dat2, paragraphs_dat, by = "element_id")
```

Cleaning the Data

Now lets clean up dat3 in the code chunks below. First we need to unnest the `text` column to the word level so we can extract the individual words. Next we'll remove unwanted character patterns.

We want to remove unwanted data in the `Text` column as these are not actual paragraphs. Patterns we'll search for and remove include:

1. Web address pattern "[1]: http. . . "
2. Paragraphs less than 40 characters

```
# remove paragraphs that contain website links and are shorter than 41 characters
dat4 <- dat3 %>%
  mutate(link = str_detect(dat3$Text, "1]: http://", negate = FALSE),
         txt_short = nchar(dat3$Text) < 40) %>%
  filter(link == FALSE & txt_short == FALSE) %>%
  select(!c(link, txt_short))

# Let'ts check the number of rows removed
rows_all = nrow(dat3)
rows_new = nrow(dat4)
rows_remove = rows_all - rows_new
```

The original dataframe included 17429 paragraphs of data. Filtering out paragraphs allowed for the removal of 3954 rows of data for a new paragraph total of 13475.

Tokenizing

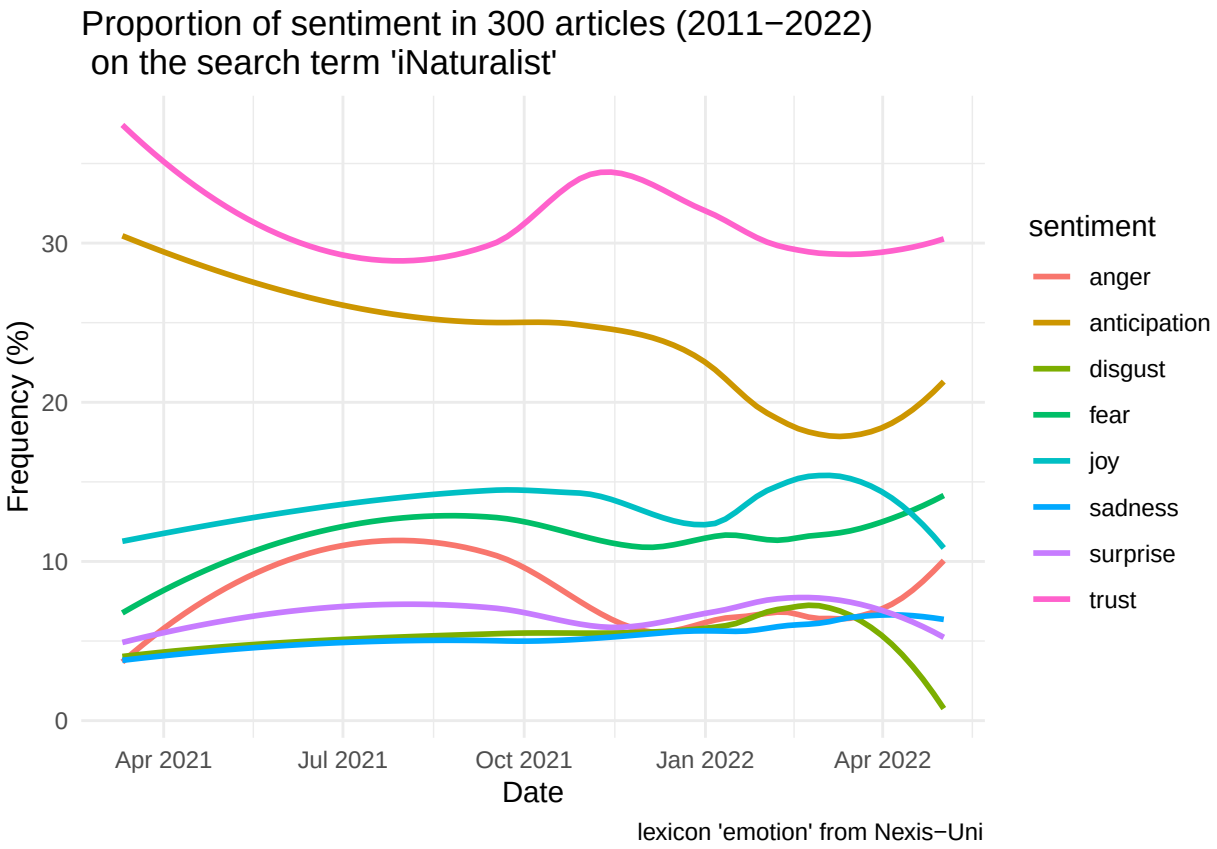
We need to unnest the filtered 'Text' column to the word level so we can label the individual sentiment words. Let's also remove stop words as standard text cleaning procedure. Note: Not every English word is in the lexicons because many English words are pretty neutral.

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

# Grabbing sentiment for paragraphs using NRC Emotions
# unnest to word-level tokens remove stop words, and join sentiment words
emotions <- dat4 %>%
  unnest_tokens(word, Text) %>%
  inner_join(emotion_words) %>%
  na.omit() %>%
  group_by(Date, sentiment) %>%

  # count number of sentiment words per day
  count() %>%
  ungroup() %>%
  group_by(Date) %>%

  # add total word
  mutate(n_max_day = sum(n),
         percent = round((n/n_max_day)*100, 2))
```

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

```r
ggplot(data = emotions, aes(x = Date, y = percent, color = sentiment)) +
  geom_smooth(se = FALSE) +
#  geom_point() +
  labs(title = "Proportion of sentiment in 300 articles (2011-2022) \n on the search term 'iNaturalist'"
       x = "Date",
       y = "Frequency (%)",
       caption = "lexicon 'emotion' from Nexis-Uni") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


Proportion of sentiment in 300 articles (2011−2022)
on the search term 'iNaturalist'
Frequency (%)
30
20
10
0
Apr 2021
Jul 2021
Oct 2021
Jan 2022
Apr 2022
Date
lexicon 'emotion' from Nexis−Uni
sentiment
anger
anticipation
disgust
fear
joy
sadness
surprise
trust