Final project

Katherine Liu

6/6/2019

```
# load all required packages
library(newsanchor)
library(robotstxt)
library(httr)
library(rvest)
library(dplyr)
library(stringr)
library(tidytext)
library(xml2)
library(tidyverse)
library(lubridate)
library(tm)
library(tidyr)
```

Webscrape New York Times articles and people.cn articles (See Final Project Code)

To make sure that the news are covering the same topics, I selected a period of time for both news source. Since the news are usually reported in a timely manner, selecting the same period of time will ensure the topics and issues covered are the same for both news sources. For web scraping articles from New York Times, I used the News API and the code from Jan Dix (Dix, Jan. Scrape New York Times Online Articles. March 05, 2019. https://cran.r-project.org/web/packages/newsanchor/vignettes/scrape-nyt.html.).

Load articles from NYT and people.cn articles

```
articles <- read.csv("articles.csv")[-1]
df_people <- read.csv("people.csv")[-1]
```

Pre-process the data for both NYT news articles and People.cn news articles

```
nytdocs <- VCorpus(VectorSource(articles$body))
nytdocs <- tm_map(nytdocs, removePunctuation)
nytdocs <- tm_map(nytdocs, content_transformer(tolower))
nytdocs <- tm_map(nytdocs, removeWords, stopwords("en"))
nytdocs <- tm_map(nytdocs, stemDocument)
nytdocsTDM <- DocumentTermMatrix(nytdocs)
nytdocsTDM <- removeSparseTerms(nytdocsTDM, 0.99)
nytdocsTidy <- tidy(nytdocsTDM)
nyttf_idf <- nytdocsTidy %>%
  bind_tf_idf(term, document, count)
```

```
ppldocs <- VCorpus(VectorSource(df_people$body))
ppldocs <- tm_map(ppldocs, removePunctuation)
ppldocs <- tm_map(ppldocs, content_transformer(tolower))
ppldocs <- tm_map(ppldocs, removeWords, stopwords("en"))
ppldocs <- tm_map(ppldocs, stemDocument)
ppldocsTDM <- DocumentTermMatrix(ppldocs)
ppldocsTDM <- removeSparseTerms(ppldocsTDM, 0.99)
ppldocsTidy <- tidy(ppldocsTDM)
ppltf_idf <- ppldocsTidy %>%
  bind_tf_idf(term, document, count)
```

Word Frequency

```
# top 20 most commonly occurring terms across news in NYT
nytdocsTidy %>%
  group_by(term) %>%
  summarize(frequency = sum(count)) %>%
  arrange(desc(frequency)) %>%
  top_n(20)
```

```
## Selecting by frequency
```

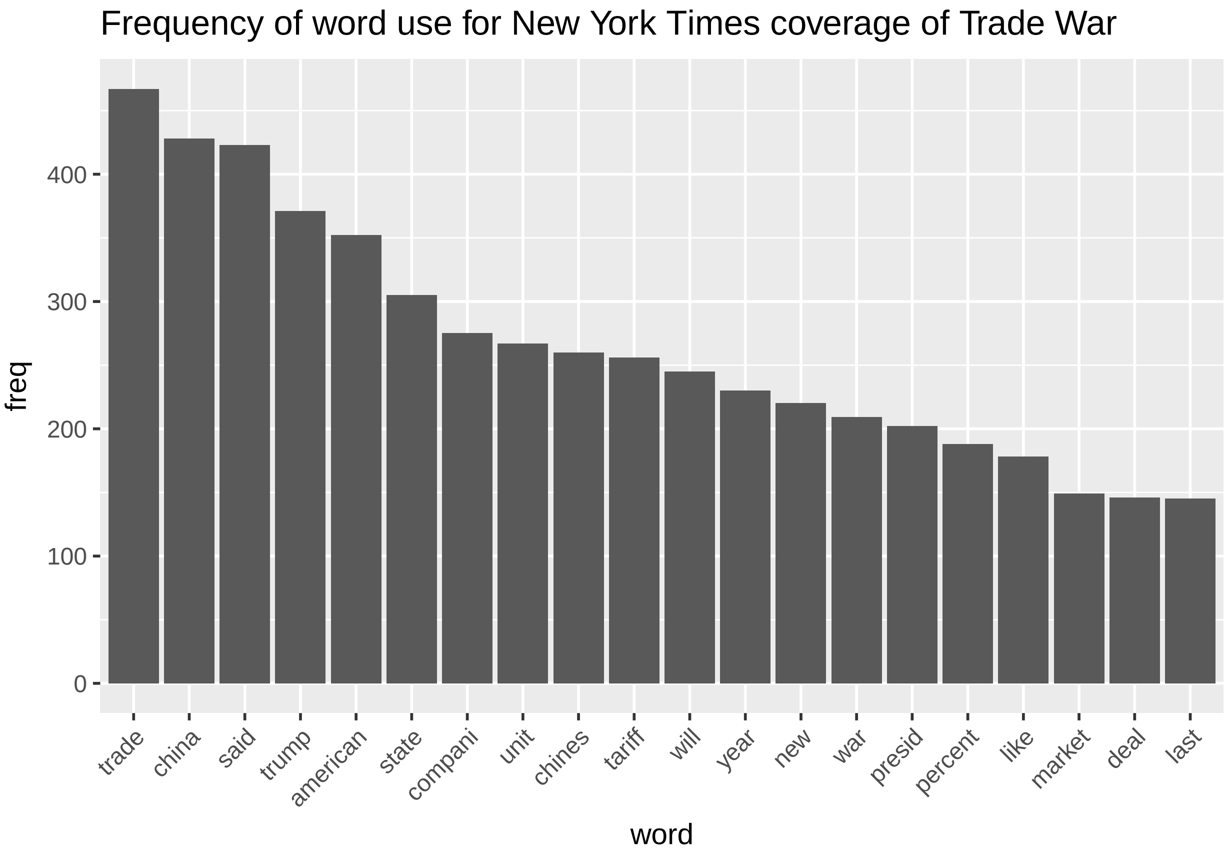
```
## # A tibble: 20 x 2
##    term     frequency
##    <chr>        <dbl>
##  1 trade          467
##  2 china          428
##  3 said           423
##  4 trump          371
##  5 american       352
##  6 state          305
##  7 compani        275
##  8 unit           267
##  9 chines         260
## 10 tariff         256
## 11 will           245
## 12 year           230
## 13 new            220
## 14 war            209
## 15 presid         202
## 16 percent        188
## 17 like           178
## 18 market         149
## 19 deal           146
## 20 last           145
```

```
# top 20 most commonly occurring terms across news in People.cn
ppldocsTidy %>%
  group_by(term) %>%
  summarize(frequency = sum(count)) %>%
  arrange(desc(frequency)) %>%
  top_n(20)
```

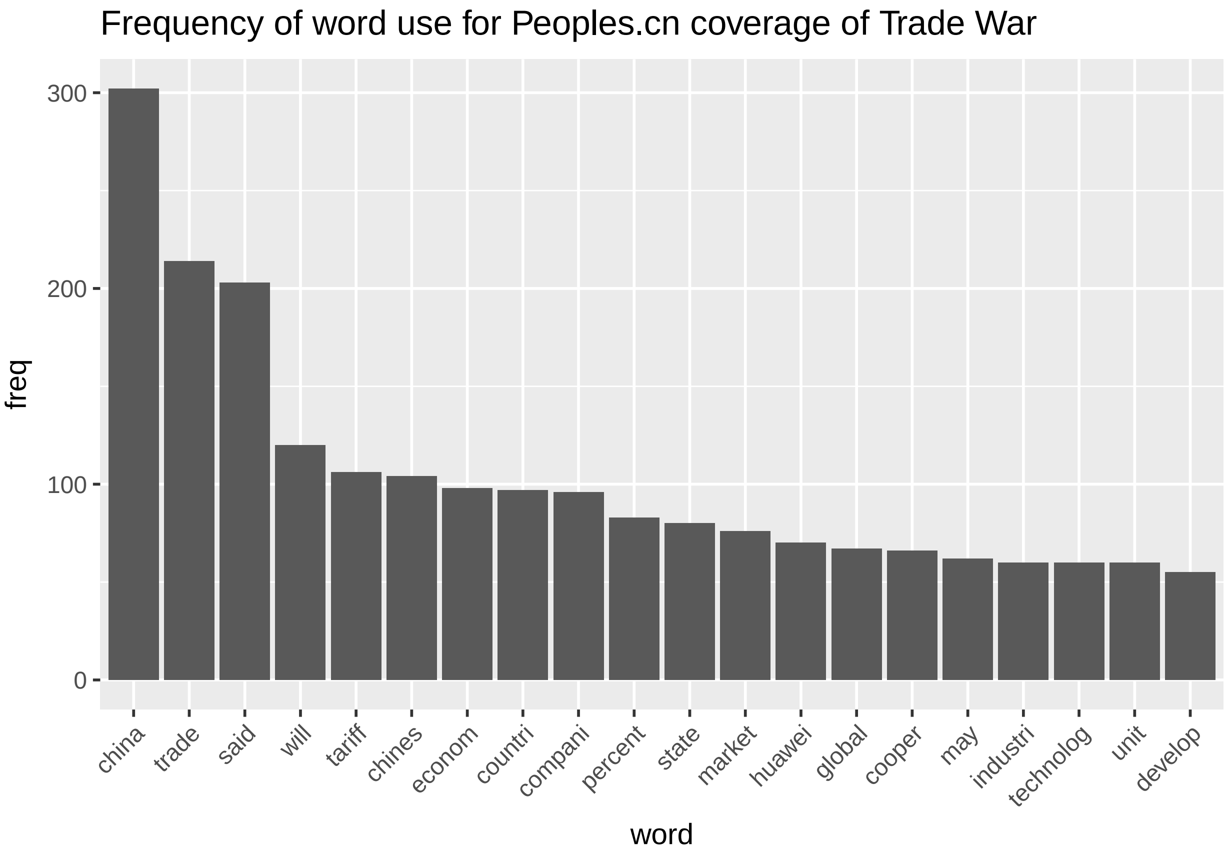
```
## Selecting by frequency
```

```
## # A tibble: 20 x 2
##    term      frequency
##    <chr>         <dbl>
##  1 china           302
##  2 trade           214
##  3 said            203
##  4 will            120
##  5 tariff          106
##  6 chines          104
##  7 econom           98
##  8 countri          97
##  9 compani          96
## 10 percent          83
## 11 state            80
## 12 market           76
## 13 huawei           70
## 14 global           67
## 15 cooper           66
## 16 may              62
## 17 industri         60
## 18 technolog        60
## 19 unit             60
## 20 develop          55
```

```r
#plot the frequency
nytdocsTidy %>%
  group_by(term) %>%
  summarize(freq = sum(count)) %>%
  top_n(20, freq) %>%
  arrange(desc(freq)) %>%
  ggplot(aes(reorder(term, -freq), freq)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle=45, hjust=1)) + xlab("word") +
  ggtitle("Frequency of word use for New York Times coverage of Trade War")
```



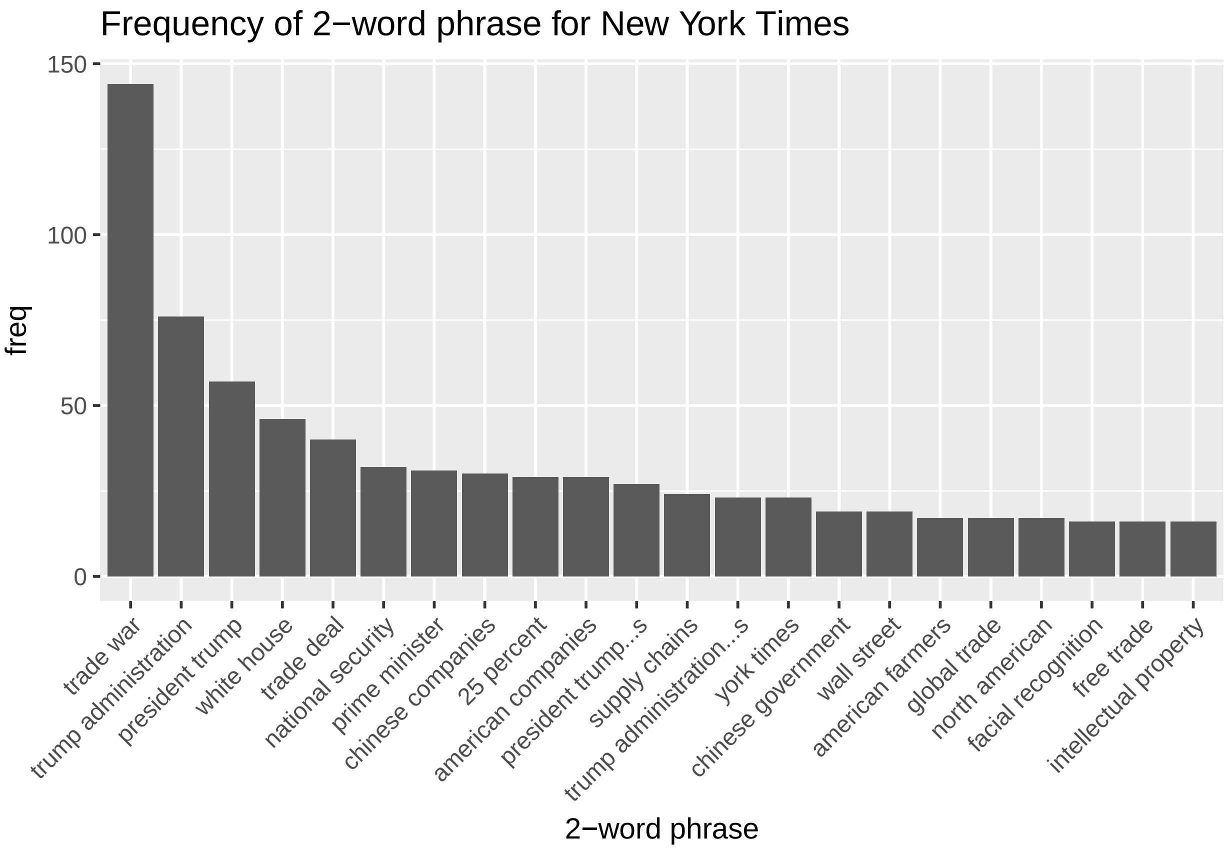
```
ppldocsTidy %>%
  group_by(term) %>%
  summarize(freq = sum(count)) %>%
  top_n(20, freq) %>%
  arrange(desc(freq)) %>%
  ggplot(aes(reorder(term, -freq), freq)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle=45, hjust=1)) + xlab("word") +
  ggtitle("Frequency of word use for Peoples.cn coverage of Trade War")
```



We can see that the New York Times focuses more on the political impact of the trade war and Trump's impact on the trade war whereas the Chinese news source focuses more on China's economy. The Chinese news source also focuses on Huawei and the technology side of the trade war, which is relatively not that important for the New York Times.

Relationships between words: n-grams

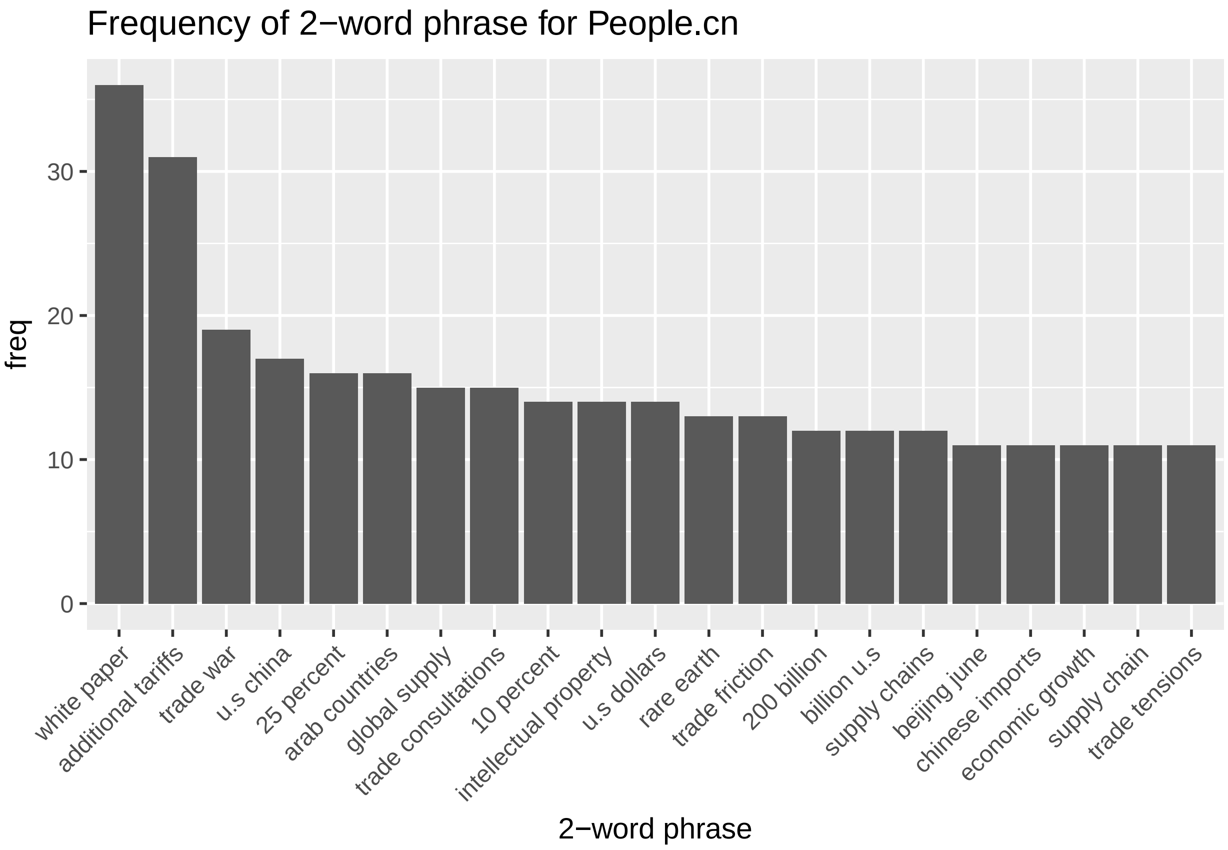
```
## for New York Times articles
articles <- articles %>%
  select(-content)

nyt_bigrams <- articles %>%
  unnest_tokens(bigram, body, token = "ngrams", n = 2)

nytbigrams_separated <- nyt_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

nytbigrams_filtered <- nytbigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

nytbigrams_united <- nytbigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")

nytbigram_counts <- nytbigrams_united %>%
  count(bigram, sort = TRUE) %>%
  mutate(freq = n)

nytbigram_counts %>%
  top_n(20, freq) %>%
  arrange(desc(freq)) %>%
  ggplot(aes(reorder(bigram, -freq), freq)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle=45, hjust=1)) + xlab("2-word phrase") +
  ggtitle("Frequency of 2-word phrase for New York Times")
```



```
ppl_bigrams <- df_people %>%
  unnest_tokens(bigram, body, token = "ngrams", n = 2)

pplbigrams_separated <- ppl_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

pplbigrams_filtered <- pplbigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)



pplbigrams_united <- pplbigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")

pplbigram_counts <- pplbigrams_united %>%
  count(bigram, sort = TRUE) %>%
  mutate(freq = n)
pplbigram_counts
```

```
## # A tibble: 3,461 x 3
##    bigram                                  n  freq
##    <chr>                               <int> <int>
##  1 addthis_config data_track_addressbar    48    48
##  2 data_track_addressbar false             48    48
##  3 var addthis_config                      48    48
##  4 white paper                             36    36
##  5 additional tariffs                      31    31
##  6 trade war                               19    19
##  7 u.s china                               17    17
##  8 25 percent                              16    16
##  9 arab countries                          16    16
## 10 global supply                           15    15
## # ... with 3,451 more rows
```

```
#Notice that the top three bigrams are codelines.
#Therefore, we want to remove these top three
pplbigram_counts %>%
  filter(! bigram %in% c('addthis_config data_track_addressbar', 'data_track_addressbar false', 'var ado
  top_n(20, freq) %>%
  arrange(desc(freq)) %>%
  ggplot(aes(reorder(bigram, -freq), freq)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle=45, hjust=1)) + xlab("2-word phrase") +
  ggtitle("Frequency of 2-word phrase for People.cn")
```



Similar to the previous discovery, we can observe that the news in New York Times focus more on the political side of Trade War.

Sentiment Analysis

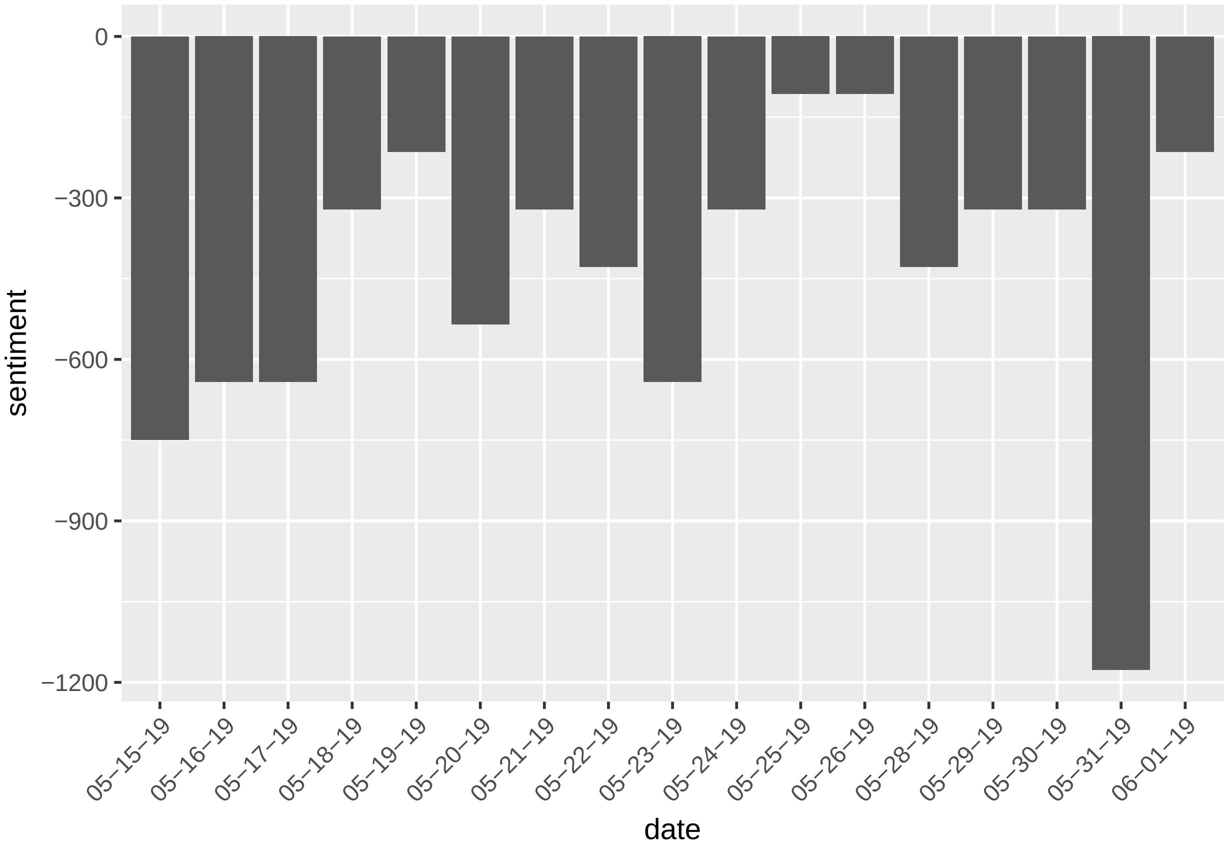
```
for (i in 1:nrow(articles)){
  articles$time[i] = toString(articles$published_at[i])
}

articles <- articles %>%
  mutate(datetime = mdy_hm(time),
         date = format(datetime, format="%m-%d-%y")
         )

df_people <- df_people %>%
  mutate(datetime = as.Date(df_people$published_at),
         date = format(datetime, format="%m-%d-%y")
         )
nyt_body <- articles %>%
  select(body, date) %>%
  filter((! is.na(body))) %>%
  filter(body != "") %>%
  mutate(text = toString(body))

tidy_nyt <- nyt_body %>%
  mutate(article_id = row_number()) %>%
  unnest_tokens(word, text)

nyt_sentiment <- tidy_nyt %>%
  inner_join(get_sentiments("bing")) %>%
  count(date, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

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

```
nyt_sentiment
```

```
## # A tibble: 17 x 4
##    date     negative positive sentiment
##    <chr>       <dbl>    <dbl>     <dbl>
##  1 05-15-19    17101    16352      -749
##  2 05-16-19    14658    14016      -642
##  3 05-17-19    14658    14016      -642
##  4 05-18-19     7329     7008      -321
##  5 05-19-19     4886     4672      -214
##  6 05-20-19    12215    11680      -535
##  7 05-21-19     7329     7008      -321
##  8 05-22-19     9772     9344      -428
##  9 05-23-19    14658    14016      -642
## 10 05-24-19     7329     7008      -321
## 11 05-25-19     2443     2336      -107
## 12 05-26-19     2443     2336      -107
## 13 05-28-19     9772     9344      -428
## 14 05-29-19     7329     7008      -321
## 15 05-30-19     7329     7008      -321
## 16 05-31-19    26873    25696     -1177
## 17 06-01-19     4886     4672      -214
```

```
ggplot(nyt_sentiment, aes(date, sentiment)) +
  geom_col(show.legend = FALSE) +
  theme(axis.text.x = element_text(angle=45, hjust=1))
```



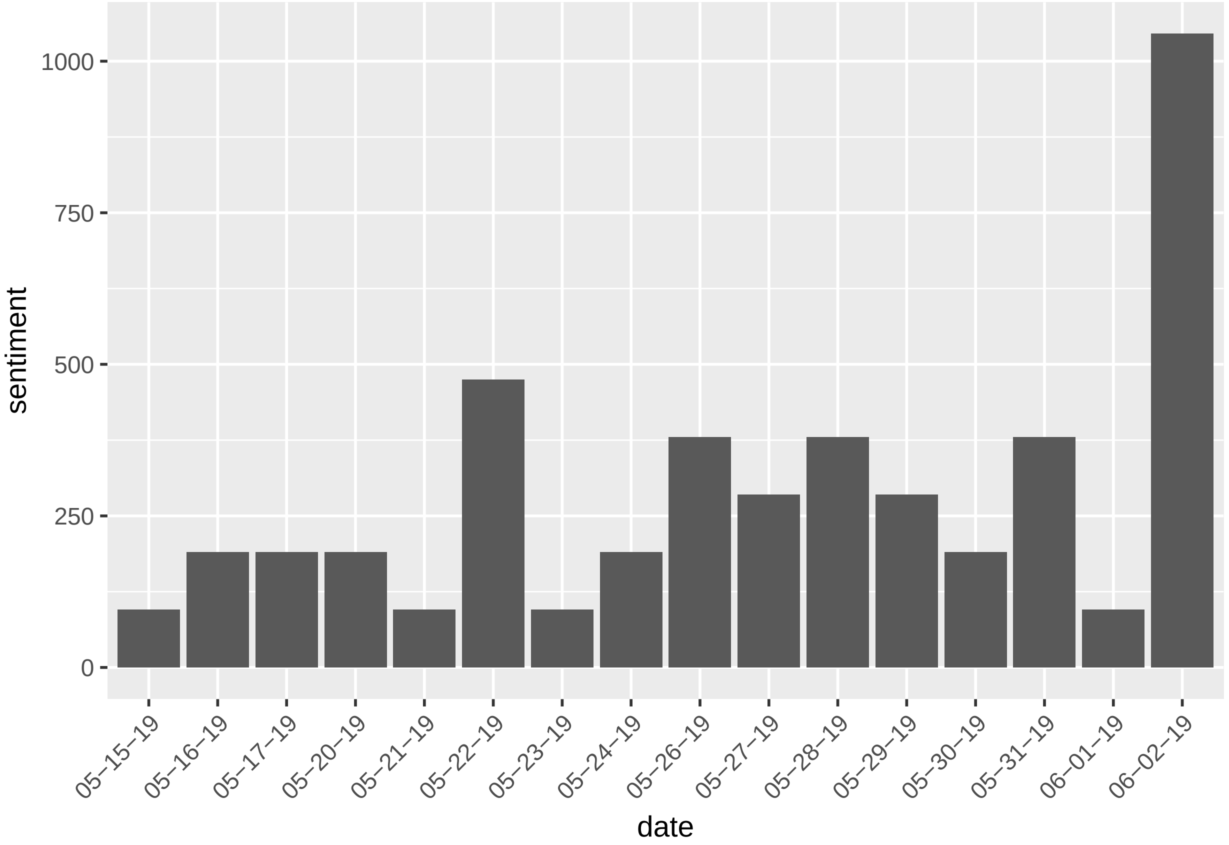
```
ppl_body <- df_people %>%
  select(body, date) %>%
  filter((! is.na(body))) %>%
  filter(body != "") %>%
  mutate(text = toString(body))

tidy_ppl <- ppl_body %>%
  mutate(article_id = row_number()) %>%
  unnest_tokens(word, text)

ppl_sentiment <- tidy_ppl %>%
  inner_join(get_sentiments("bing")) %>%
  count(date, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

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

```
ppl_sentiment
```

```
## # A tibble: 16 x 4
##    date     negative positive sentiment
##    <chr>       <dbl>    <dbl>     <dbl>
##  1 05-15-19      516      611        95
##  2 05-16-19     1032     1222       190
##  3 05-17-19     1032     1222       190
##  4 05-20-19     1032     1222       190
##  5 05-21-19      516      611        95
##   6 05-22-19     2580      3055       475
##   7 05-23-19      516       611        95
##   8 05-24-19     1032      1222       190
##   9 05-26-19     2064      2444       380
##  10 05-27-19     1548      1833       285
##  11 05-28-19     2064      2444       380
##  12 05-29-19     1548      1833       285
##  13 05-30-19     1032      1222       190
##  14 05-31-19     2064      2444       380
##  15 06-01-19      516       611        95
##  16 06-02-19     5676      6721      1045
```

```
ggplot(ppl_sentiment, aes(date, sentiment)) +
  geom_col(show.legend = FALSE) +
  theme(axis.text.x = element_text(angle=45, hjust=1))
```



The reason that I decided to use the count of the sentiment rather than use the average method is that the number of articles and the length of the articles are significant. We are more likely to see an increase in the number of articles or the length of the articles when important changes happen and I want to capture this effect. Through the sentiment analysis, we can find out that interestingly, the news coverage in People.cn is more positive in its descriptive tone while that in the New York Times is more negative. One possible reason could be that the Chinese government wants to assure the public that the Trade War situation is not that bad through news propaganda.