PS239 Project

Bergliot Christensen

December 8, 2016

Getting data:

```
setwd("C:/Users/bergliotc/Dropbox/Berkeley/PS239T/17_project")

data_1 <- read.csv('Snippets_0616_0820.csv')
data_2 <- read.csv('Snippets_2008_2010.csv')
data_3 <- read.csv("f_docs_0_120 - Copy.csv")
data_4 <- read.csv("all_formatted_20151220_20160820.csv")
data_5 <- read.csv("all_formatted_20151220_20160120_v2.csv")
data_6 <- read.csv("all_formatted_20151220_20160120.csv")
data_7 <- read.csv("Snippets_2020_2012_ut101.csv")

data_all <- rbind(data_1, data_2, data_3, data_4, data_5, data_6, data_7) # merging to 1 dataset
data_all <- unique(data_all) # making sure I only have uniqe values
```

I now chunk the data into months:

```
jun_15 <- filter(data_all, grepl("2015-06", data_all$date) == TRUE)
jul_15 <- filter(data_all, grepl("2015-07", data_all$date) == TRUE)
aug_15 <- filter(data_all, grepl("2015-08", data_all$date) == TRUE)
sep_15 <- filter(data_all, grepl("2015-09", data_all$date) == TRUE)
oct_15 <- filter(data_all, grepl("2015-10", data_all$date) == TRUE)
nov_15 <- filter(data_all, grepl("2015-11", data_all$date) == TRUE)
dec_15 <- filter(data_all, grepl("2015-12", data_all$date) == TRUE)
jan_16 <- filter(data_all, grepl("2016-01", data_all$date) == TRUE)
feb_16 <- filter(data_all, grepl("2016-02", data_all$date) == TRUE)
mar_16 <- filter(data_all, grepl("2016-03", data_all$date) == TRUE)
apr_16 <- filter(data_all, grepl("2016-04", data_all$date) == TRUE)
may_16 <- filter(data_all, grepl("2016-05", data_all$date) == TRUE)
jun_16 <- filter(data_all, grepl("2016-06", data_all$date) == TRUE)
jul_16 <- filter(data_all, grepl("2016-07", data_all$date) == TRUE)
aug_16 <- filter(data_all, grepl("2016-08", data_all$date) == TRUE)
sep_16 <- filter(data_all, grepl("2016-09", data_all$date) == TRUE)
oct_16 <- filter(data_all, grepl("2016-10", data_all$date) == TRUE)
nov_16 <- filter(data_all, grepl("2016-11", data_all$date) == TRUE)

# I check to see if I got everything by checking if the length of all my chunks matches the length of d
nrow(rbind(jun_15, jul_15, aug_15, sep_15, oct_15, nov_15, dec_15, jan_16, feb_16, mar_16, apr_16, may_
```

```
## [1] 8217
```

I now preproces every chuck to create corpi:

```
c_jun_15 <- Corpus(VectorSource(jun_15$lead_paragraph))
c_jul_15 <- Corpus(VectorSource(jul_15$lead_paragraph))
c_aug_15 <- Corpus(VectorSource(aug_15$lead_paragraph))
c_sep_15 <- Corpus(VectorSource(sep_15$lead_paragraph))
c_oct_15 <- Corpus(VectorSource(oct_15$lead_paragraph))
c_nov_15 <- Corpus(VectorSource(nov_15$lead_paragraph))
```

```r
c_dec_15 <- Corpus(VectorSource(dec_15$lead_paragraph))
c_jan_16 <- Corpus(VectorSource(jan_16$lead_paragraph))
c_feb_16 <- Corpus(VectorSource(feb_16$lead_paragraph))
c_mar_16 <- Corpus(VectorSource(mar_16$lead_paragraph))
c_apr_16 <- Corpus(VectorSource(apr_16$lead_paragraph))
c_may_16 <- Corpus(VectorSource(may_16$lead_paragraph))
c_jun_16 <- Corpus(VectorSource(jun_16$lead_paragraph))
c_jul_16 <- Corpus(VectorSource(jul_16$lead_paragraph))
c_aug_16 <- Corpus(VectorSource(aug_16$lead_paragraph))
c_sep_16 <- Corpus(VectorSource(sep_16$lead_paragraph))
c_oct_16 <- Corpus(VectorSource(oct_16$lead_paragraph))
c_nov_16 <- Corpus(VectorSource(nov_16$lead_paragraph))

# I now turn to creating document term matrices and processing each corpus.
# I initially remove my own stop words, I want to remove the name of Donald Trump as well as 'president
# Other words like "campaign", "Hillary" "elect" etc are not removed. I wish to see whether the NYT nor
my_stopwords <- c("Donald", "Trump", "president")
c_jun_15 <- tm_map(c_jun_15, removeWords, my_stopwords)
c_jul_15 <- tm_map(c_jul_15, removeWords, my_stopwords)
c_aug_15 <- tm_map(c_aug_15, removeWords, my_stopwords)
c_sep_15 <- tm_map(c_sep_15, removeWords, my_stopwords)
c_oct_15 <- tm_map(c_oct_15, removeWords, my_stopwords)
c_nov_15 <- tm_map(c_nov_15, removeWords, my_stopwords)
c_dec_15 <- tm_map(c_dec_15, removeWords, my_stopwords)
c_jan_16 <- tm_map(c_jan_16, removeWords, my_stopwords)
c_feb_16 <- tm_map(c_feb_16, removeWords, my_stopwords)
c_mar_16 <- tm_map(c_mar_16, removeWords, my_stopwords)
c_apr_16 <- tm_map(c_apr_16, removeWords, my_stopwords)
c_may_16 <- tm_map(c_may_16, removeWords, my_stopwords)
c_jun_16 <- tm_map(c_jun_16, removeWords, my_stopwords)
c_jul_16 <- tm_map(c_jul_16, removeWords, my_stopwords)
c_aug_16 <- tm_map(c_aug_16, removeWords, my_stopwords)
c_sep_16 <- tm_map(c_sep_16, removeWords, my_stopwords)
c_oct_16 <- tm_map(c_oct_16, removeWords, my_stopwords)
c_nov_16 <- tm_map(c_nov_16, removeWords, my_stopwords)

# I then create document term matrices:
dtm_jun_15 <- DocumentTermMatrix(c_jun_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_jul_15 <- DocumentTermMatrix(c_jul_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_aug_15 <- DocumentTermMatrix(c_aug_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_sep_15 <- DocumentTermMatrix(c_sep_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_oct_15 <- DocumentTermMatrix(c_oct_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_nov_15 <- DocumentTermMatrix(c_nov_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_dec_15 <- DocumentTermMatrix(c_dec_15, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_jan_16 <- DocumentTermMatrix(c_jan_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_feb_16 <- DocumentTermMatrix(c_feb_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_mar_16 <- DocumentTermMatrix(c_mar_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_apr_16 <- DocumentTermMatrix(c_apr_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_may_16 <- DocumentTermMatrix(c_may_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_jun_16 <- DocumentTermMatrix(c_jun_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_jul_16 <- DocumentTermMatrix(c_jul_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_aug_16 <- DocumentTermMatrix(c_aug_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_sep_16 <- DocumentTermMatrix(c_sep_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_oct_16 <- DocumentTermMatrix(c_oct_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
dtm_nov_16 <- DocumentTermMatrix(c_nov_16, control = list(tolower = TRUE,
                          removePunctuation = TRUE, removeNumbers = TRUE,
                          stopwords = TRUE, stemming=TRUE))
```

I now have a DTM for every month, and I can begin analyzing the frequency of different words and how these frequencies change over time:

```
# I now create a loop that spits out the most frequent terms in every month:

months_chunked_dtm <- list(dtm_jun_16, dtm_jul_16, dtm_aug_15, dtm_sep_15, dtm_oct_15, dtm_nov_15, dtm_

# Define a function to take as its inptút a list of DTMs and return their frequencies relative to the
get_freq <- function(dtm){
  findFreqTerms(dtm, lowfreq= (dtm$ncol/100))
}

# apply this function to all DTMs
# freq_terms <- lapply(months_chunked_dtm, get_freq)
# freq_terms blanked out due to space issues
```

Unfortunately, eyeballing the most frequent terms in each month shows that all words are fairly uncontroversial, i.e. no ‘liar’, ‘demagogue’ or even ‘assault’. This means that there is little point in moving forward with an explorative analysis of the change of the most frequent terms.

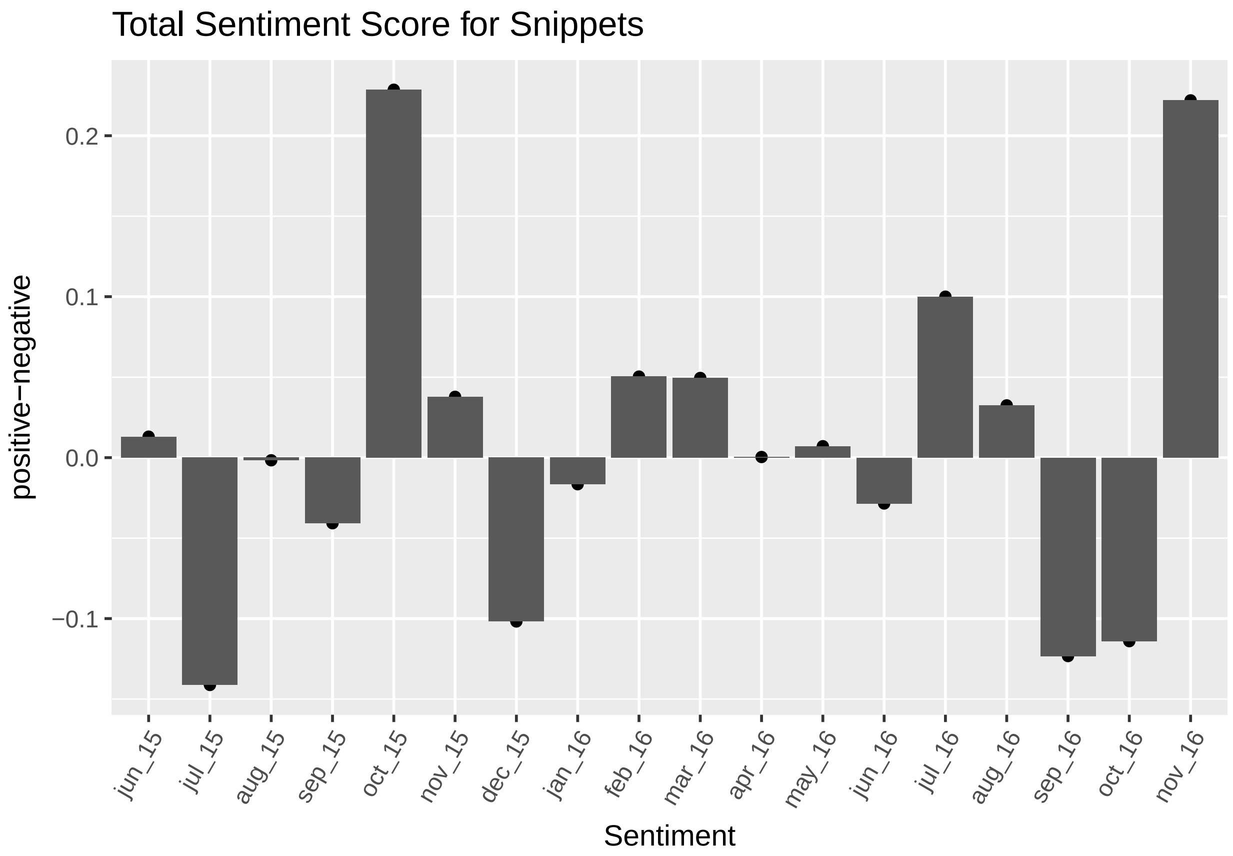
I am able, however, to find the frecuency of specific words, for example the following words:

```
# z <- inspect( dtm_nov_16[dimnames(dtm_nov_16)$Docs, c("lie", "accus", "putin", "assault", "women", "d
# colSums(z)
```

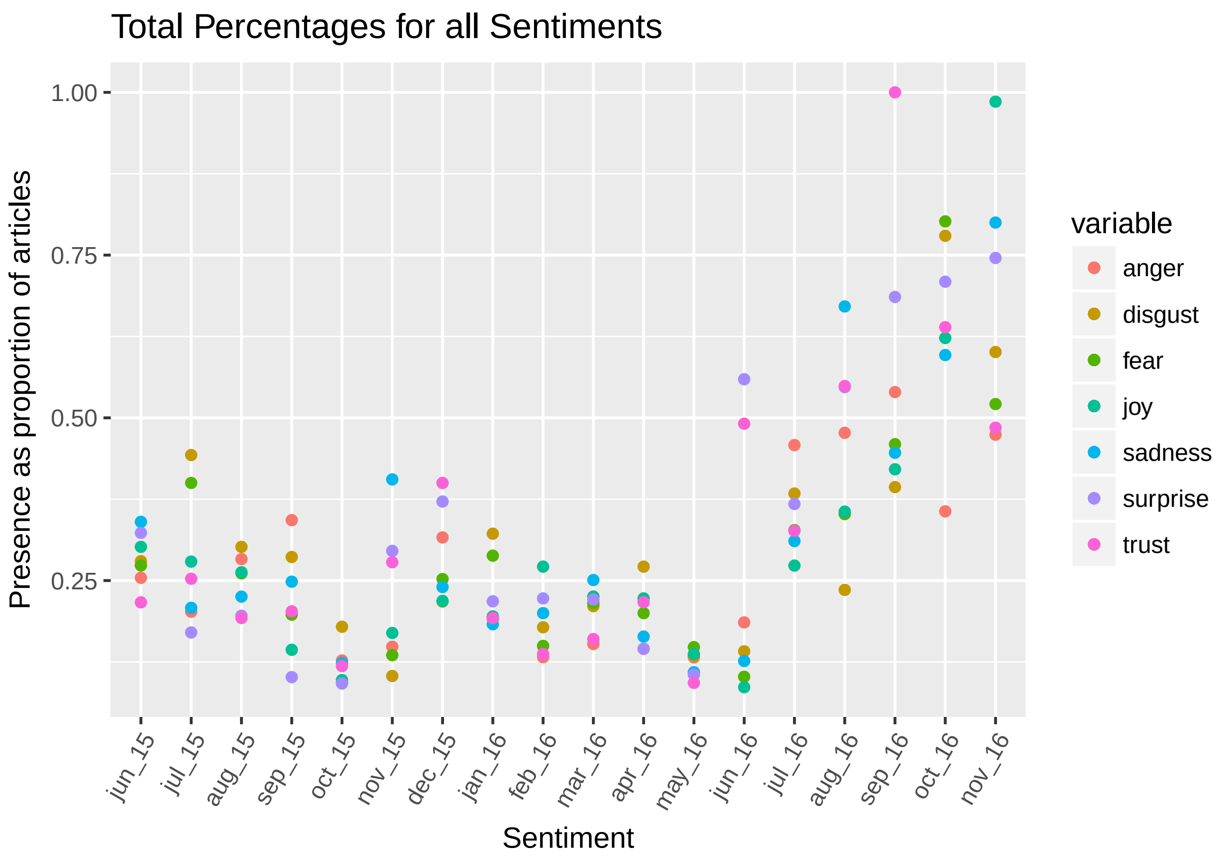
Due to the stemming of the words, it is hard to predict what form specific key words will take. For that reason, extracting frequencies of words of interest is hard to do with an apriori word list. Also, the above function, ‘inspect’, throws an error instead of 0 in cases where the word does not appear at all. This means that I will have to create specialized word-lists for all different chunks of months and cannot loop through all DTM’s at once.

ANALYZING SENTIMENTS

```
# I create a list of all the different months:
months_chunked <- list(jun_15, jul_15, aug_15, sep_15, oct_15, nov_15, dec_15, jan_16, feb_16, mar_16, a

# Define a function to take as its input a list of month-dataframes and return the sentiments og the mo
get_sentiment <- function(chunk){
  snippet_df <- data.frame(chunk$snippet) # get snippet as dataframe
  snippet_vec <-  as.vector(snippet_df$chunk.snippet) # get snippet out of this dataframe as vector
  snippet_char <- paste(c(snippet_vec), collapse=', ' ) #collapse into character
  sentiment <- get_nrc_sentiment(snippet_char) # get sentiment
  sentiment/(nrow(chunk)) # divide by number of rows to get percentage
}

# Apply this function to get sentiment for every month:
sentiment <- sapply(months_chunked, get_sentiment)

# This results in a wide dataset, and I want to transpose columns and rows::
sentiment <- as.data.frame(t(sentiment))

# This object is still not a dataframe, I therefore use the following command to create a dataframe:
sentiment_df <- data.frame(matrix(unlist(sentiment), nrow=18, byrow=T))

# I add date-column:
sentiment_df$date <- as.vector(c('jun_15', 'jul_15', 'aug_15', 'sep_15', 'oct_15', 'nov_15', 'dec_15',

# I code the date as a factor, so that ggplot will not sort it for me in the plot:
sentiment_df$date <- factor(sentiment_df$date, levels = sentiment_df$date)

colnames(sentiment_df) <- c('anger', 'anticipation', 'disgust', 'fear', 'joy', 'sadness', 'surprise', '
rownames(sentiment_df) <- c('jun_15', 'jul_15', 'aug_15', 'sep_15', 'oct_15', 'nov_15', 'dec_15', 'jan_

# I get average sentiment-score by subtracting negative from positive:
sentiment_df$sentiment <- sentiment_df$positive - sentiment_df$negative

# I now try to plot the development of net positive sentiment over time:
ggplot(data = sentiment_df, aes(x = date, y = sentiment)) +
  geom_point()+
  geom_bar(stat = "identity")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  xlab("Sentiment") + ylab("positive-negative") + ggtitle("Total Sentiment Score for Snippets")
```



```
# also want to see the development of the individual sentiments:
ggplot(data = sentiment_df, aes(x = date, y = value, color = variable)) +
  geom_point((aes(y = anger, col = "anger")))+
  geom_point((aes(y = disgust, col = "disgust")))+
  geom_point((aes(y = fear, col = "fear")))+
  geom_point((aes(y = joy, col = "joy")))+
  geom_point((aes(y = sadness, col = "sadness")))+
  geom_point((aes(y = surprise, col = "surprise")))+
  geom_point((aes(y = trust, col = "trust")))+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  xlab("Sentiment") + ylab("Presence as proportion of articles") + ggtitle("Total Percentages for all S
```

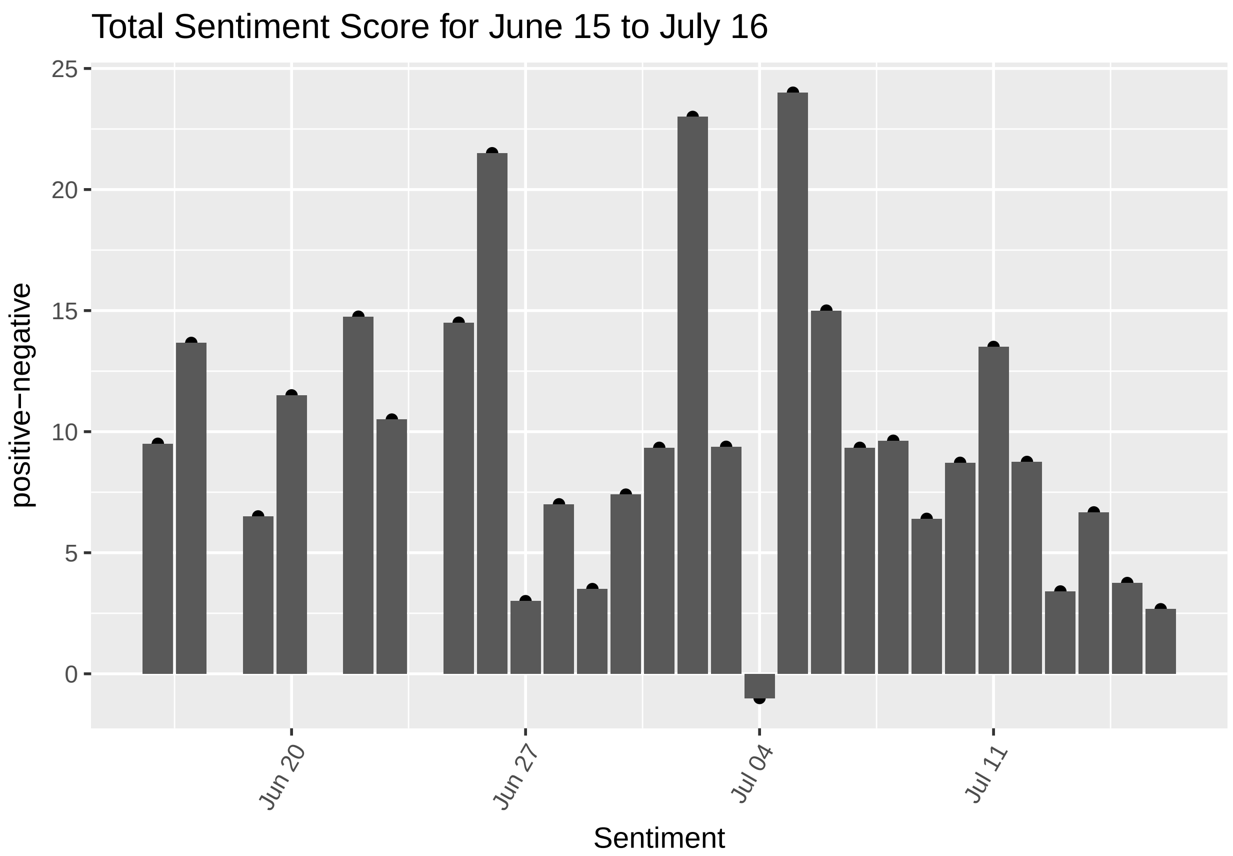


ANALYZING FULL LENGTH ARTICLES

```
full_jun <- as.data.frame(read.csv("June_articles2.csv"))
dates_jun <- read.csv("June_date2.csv")
dates_jun <- as.Date(dates_jun$X1.of, format = "%m/%d")


# I get sentiments:
df_june_sentiments <- lapply(as.vector(full_jun[,1]), get_nrc_sentiment)
june_sentiment <- as.data.frame(t(df_june_sentiments))
sentiment_df <- data.frame(matrix(unlist(june_sentiment), nrow=104, byrow=T))


df_june <- cbind(full_jun, dates_jun, sentiment_df)
june_df <- data.frame(matrix(unlist(df_june), nrow=104, byrow=F))

colnames(df_june) <- c('articles', 'date', 'anger', 'anticipation', 'disgust', 'fear', 'joy', 'sadness'

# sentiment analysis
df_june$sentiment <- df_june$positive - df_june$negative
# There are multiple articles pr day, so I get the mean sentiment pr day using dplyr:
sentiment_day <- df_june %>% # I put in my dataset
  group_by(date)  %>% # grouping by date
  summarise(date.sentiment=mean(sentiment, na.rm=T)) # name new variable and specify function

as.data.frame(sentiment_day)
```

```
##          date date.sentiment
## 1  2016-06-16       9.500000
## 2  2016-06-17      13.666667
## 3  2016-06-19       6.500000
## 4  2016-06-20      11.500000
## 5  2016-06-22      14.750000
## 6  2016-06-23      10.500000
## 7  2016-06-25      14.500000
## 8  2016-06-26      21.500000
## 9  2016-06-27       3.000000
## 10 2016-06-28       7.000000
## 11 2016-06-29       3.500000
## 12 2016-06-30       7.400000
## 13 2016-07-01       9.333333
## 14 2016-07-02      23.000000
## 15 2016-07-03       9.375000
## 16 2016-07-04      -1.000000
## 17 2016-07-05      24.000000
## 18 2016-07-06      15.000000
## 19 2016-07-07       9.333333
## 20 2016-07-08       9.625000
## 21 2016-07-09       6.400000
## 22 2016-07-10       8.714286
## 23 2016-07-11      13.500000
## 24 2016-07-12       8.750000
## 25 2016-07-13       3.400000
## 26 2016-07-14       6.666667
## 27 2016-07-15       3.750000
## 28 2016-07-16       2.666667
```

```
# I now graph the average sentiment:
ggplot(data =sentiment_day, aes(x = date, y = date.sentiment)) +
  geom_point()+
  geom_bar(stat = "identity")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  xlab("Sentiment") + ylab("positive-negative") + ggtitle("Total Sentiment Score for June 15 to July 16
```



```
mean(sentiment_day$date.sentiment)
```

```
## [1] 9.851105
```

The graph shows that coverage of Trump was mostly positive. All dates but one had positive coverage, and the mean score was 9.8, meaning that articles had app. 9 more positively attributed words/sentences than negative ones. Moreover, the graph shows that coverage actually grew less positive over the course of the summer.

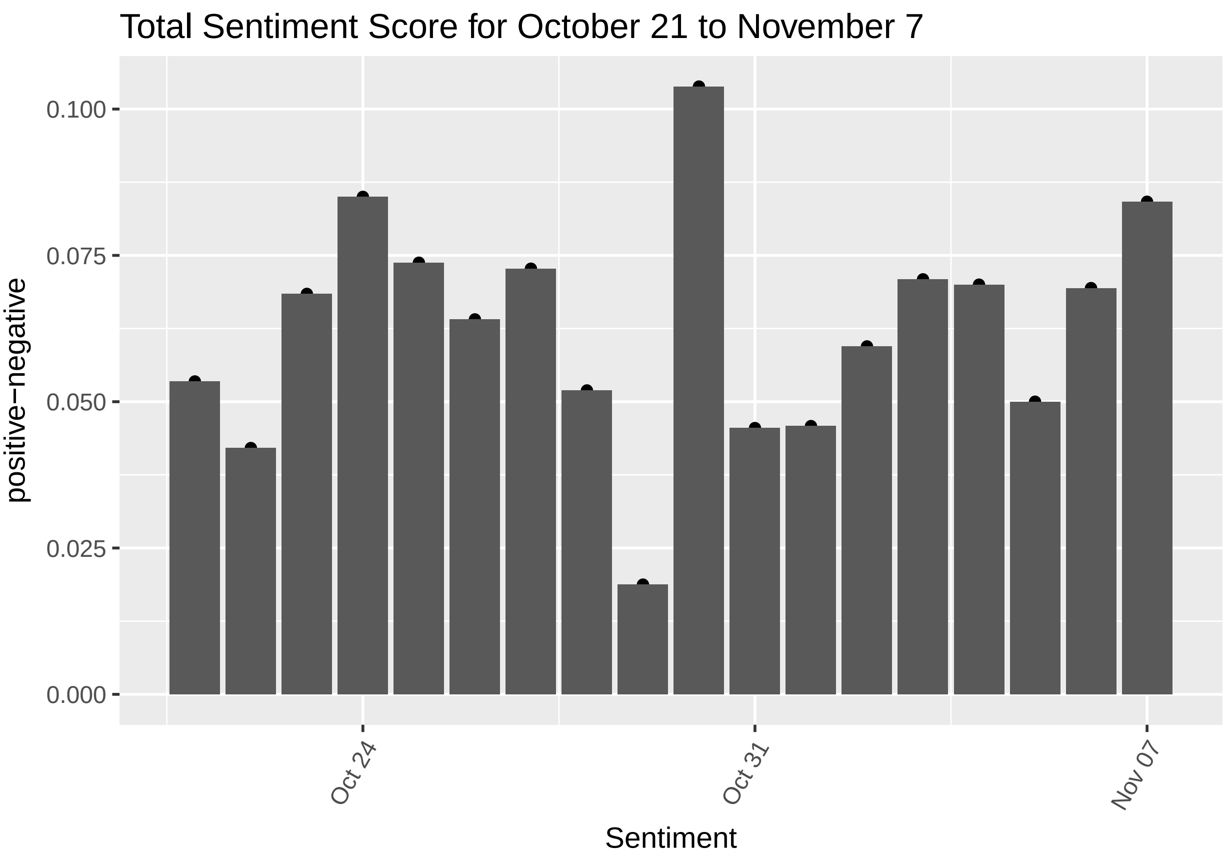
I repeat all these steps for the last month before Trumps election:

```
full_nov1 <- as.data.frame(read.csv("November_articles1.csv"))
full_nov2 <- as.data.frame(read.csv("November_articles2.csv"))
nov_dates1 <- as.data.frame(read.csv("November_date1a.csv"))
nov_dates2 <- as.data.frame(read.csv("November_date2.csv"))

nov_dates1a <- as.Date(nov_dates1$X2.of, format = "%m/%d")
nov_dates2a <- as.data.frame(as.Date(nov_dates2$X1.of, format = "%m/%d"))

# I now wish to combine the dataframes into one dataframe:
colnames(nov_dates1) <- c('date')
colnames(nov_dates2) <- c('date')
colnames(full_nov1) <- c('articles')
colnames(full_nov2) <- c('articles')
November_1 <- cbind(full_nov1, nov_dates1a)
November_2 <- cbind(full_nov2, nov_dates2a)
colnames(November_1) <- c('articles', 'date')
colnames(November_2) <- c('articles', 'date')
```

```
Nov_all <- rbind(November_1, November_2)

# This dataframe is too bit to work with. I therefore create a randon sample of 200 different articles
nov_sam <- November_2[sample(nrow(November_2),size=200,replace=FALSE),]

nov2_sentiment_raw <- lapply(as.vector(nov_sam[,1]), get_nrc_sentiment)
nov2_sentiment <- as.data.frame(t(nov2_sentiment_raw))
nov2_sentiment_df <- data.frame(matrix(unlist(nov2_sentiment), nrow=200, byrow=T))
nov2_sentiment_df <- nov2_sentiment_df/200
df_nov2 <- cbind(nov_sam, nov2_sentiment_df)
# df_nov2 <- data.frame(matrix(unlist(df_nov2), nrow=200, byrow=F))

colnames(df_nov2) <- c('articles', 'date', 'anger', 'anticipation', 'disgust', 'fear', 'joy', 'sadness'

# sentiment analysis
df_nov2$sentiment <- df_nov2$positive - df_nov2$negative

# There are multiple articles pr day, so I get the mean sentiment pr day using dplyr:
sentiment_day_nov <- df_nov2 %>% # I put in my dataset
  group_by(date)  %>% # grouping by date
  summarise(date.sentiment=mean(sentiment, na.rm=T)) # name new variable and specify function

as.data.frame(sentiment_day_nov)
```

```
##          date date.sentiment
## 1  2016-10-21     0.05346154
## 2  2016-10-22     0.04208333
## 3  2016-10-23     0.06842105
## 4  2016-10-24     0.08500000
## 5  2016-10-25     0.07375000
## 6  2016-10-26     0.06406250
## 7  2016-10-27     0.07272727
## 8  2016-10-28     0.05192308
## 9  2016-10-29     0.01875000
## 10 2016-10-30     0.10384615
## 11 2016-10-31     0.04550000
## 12 2016-11-01     0.04583333
## 13 2016-11-02     0.05944444
## 14 2016-11-03     0.07090909
## 15 2016-11-04     0.07000000
## 16 2016-11-05     0.05000000
## 17 2016-11-06     0.06941176
## 18 2016-11-07     0.08416667
```

```
# I now graph the average sentiment:
ggplot(data =sentiment_day_nov, aes(x = date, y = date.sentiment)) +
  geom_point()+
  geom_bar(stat = "identity")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  xlab("Sentiment") + ylab("positive-negative") + ggtitle("Total Sentiment Score for October 21 to Nover
```



This graph indicates that the coverage of Trump was overall positive in the weeks leading up to the election. I find this result very surprising, to say the least. Not only did coverage on average get positive from October 30 and until the election, positive sentiments also outweighed negative sentiments on average for the entire period.

These results bring either my application of the syuzhet-package or the package itself into question.

WORD FREQUENCIES IN FULL LENGTH ARTICLES

I initially load data and turn it into a corpus:

```r
# JUNE
setwd("C:/Users/bergliotc/Dropbox/Berkeley/PS239T/17_project")
june_articles <- read.csv("June_articles2.csv", header = TRUE)
june_articles <- Corpus(VectorSource(june_articles$articles))
```

Next step is to proces my data before conducting analysis:

```r
june_dat <- Corpus(VectorSource(june_articles)) # loading the data as corpus
june_dat <- tm_map(june_dat, removeWords, c("Donald", "Trump", "column", "words", "said", "highlight",
full_june_dtm <- DocumentTermMatrix(june_dat,
           control = list(tolower = TRUE,
                          removePunctuation = TRUE,
                          removeNumbers = TRUE,
                          stopwords = TRUE,
                          stemming=TRUE))
```

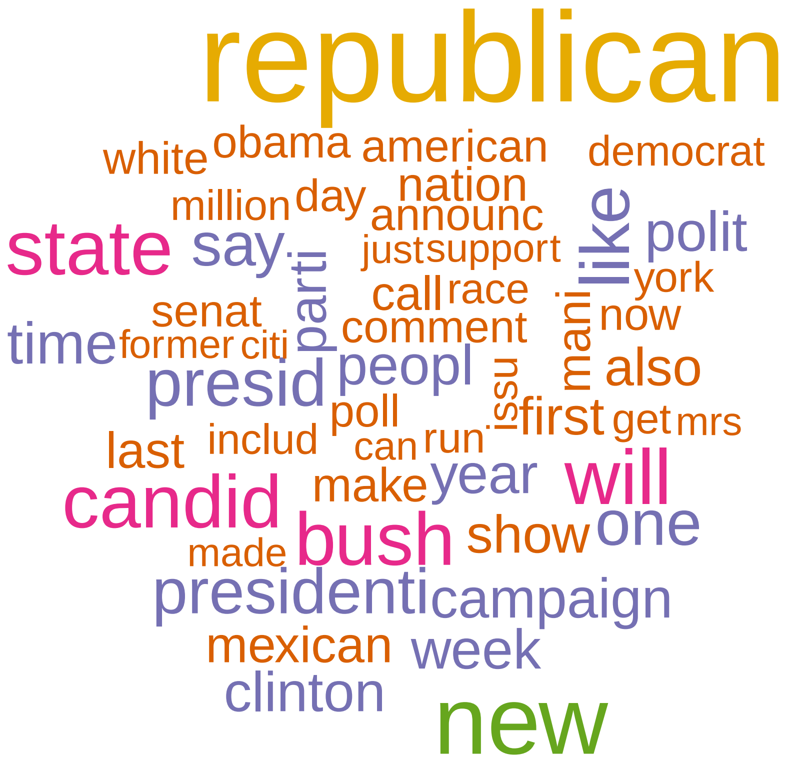
I now turn to analyzing my data:

```
# wordcloud!
freq_full_june <- colSums(as.matrix(full_june_dtm))
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 3.3.2
```

```
set.seed(1)
wordcloud(names(freq_full_june), freq_full_june, min.freq = 70, colors=brewer.pal(6,"Dark2"))
```

```
## Warning in wordcloud(names(freq_full_june), freq_full_june, min.freq =
## 70, : immigr could not be fit on page. It will not be plotted.
```



```
?wordcloud
```

```
## starting httpd help server ...
```

```
##  done
```

```
freq_full_june[1:30]
```

```
##    â bruce     â oper      âadio    abandon       abat      abbey
##          1          1          1          8          1          1
##        abc     abhorr       abil        abl      abort      abras
##          3          1          8          6          2          1
##     absenc    absolut     absorb   abstract     absurd       abus
##          1          6          1          1          2          2
##       abut     academ    acceler     accent     accept     access
##          1          1          1          1          4          3
##      accid     acclim   accommod  accompani accomplish     accord
##          1          1          1          2          2         23
```

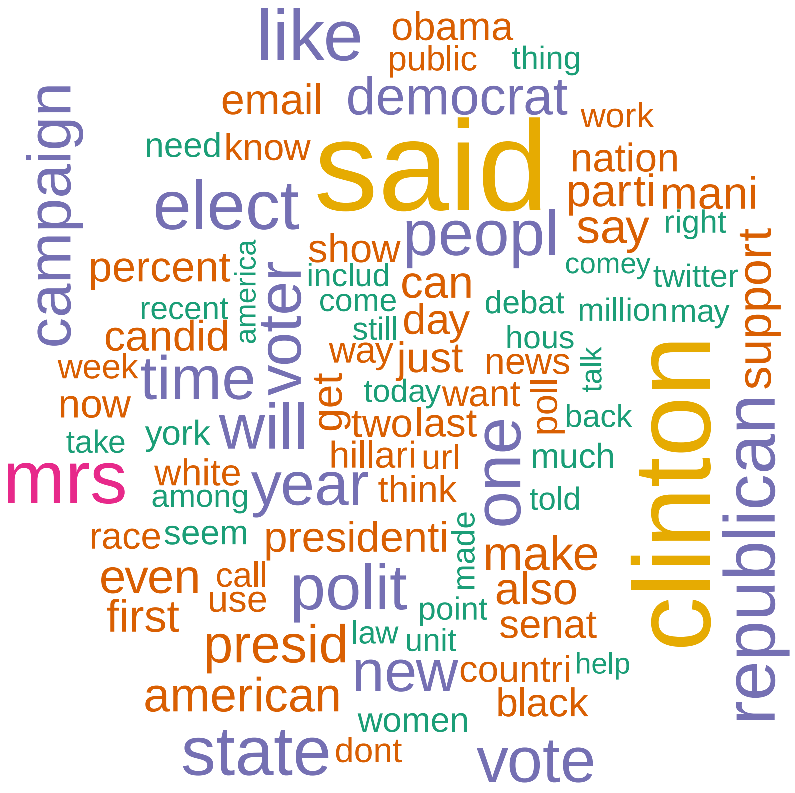
```
# I also get the frequency of certain critical words of interest so that I can compare with November:
y <- inspect( full_june_dtm[dimnames(full_june_dtm)$Docs, c("lie", "accus", "putin", "assault", "women"
```

```
## <<DocumentTermMatrix (documents: 104, terms: 7)>>
## Non-/sparse entries: 50/678
## Sparsity           : 93%
## Maximal term length: 7
## Weighting          : term frequency (tf)
##
##      Terms
## Docs  lie accus putin assault women danger racist
##   1     0     0     0       0     0      0      1
##   2     1     0     0       0     0      0      3
##   3     0     0     0       0     0      0      0
##   4     0     0     0       0     0      0      0
##   5     0     0     0       0     0      0      0
##   6     0     0     0       0     0      0      0
##   7     0     0     0       0     0      0      0
##   8     0     0     0       0     0      0      0
##   9     0     0     0       0     0      0      0
##   10    1     1     1       0     0      0      0
##   11    0     0     0       0     0      0      0
##   12    0     0     0       0     0      0      0
##   13    0     0     0       0     0      0      0
##   14    0     0     0       0     0      0      0
##   15    0     1     0       0     0      0      0
##   16    0     0     0       0     0      0      0
##   17    0     0     0       0     0      0      0
##   18    0     0     0       0     1      0      0
##   19    1     0     0       0     2      0      0
##   20    0     1     0       0     1      0      0
##   21    0     0     0       0     0      0      0
##   22    0     0     0       0     0      0      0
##   23    0     0     0       0     0      0      0
##   24    0     0     0       1     0      0      0
##   25    0     0     0       0     0      0      0
##   26    0     0     0       0     0      0      0
##   27    0     1     0       0     0      0      0
##   28    0     0     0       0     0      0      0
##   29    0     1     0       0     0      0      0
##   30    0     0     0       0     1      0      0
##   31    0     0     0       0     0      0      0
##   32    0     0     0       0     0      0      0
##   33    1     0     0       0     1      0      1
##   34    0     0     0       0     0      0      0
##   35    0     0     0       0     4      0      0
##   36    0     0     0       0     0      0      0
##   37    0     0     0       0     0      0      0
##   38    0     0     0       0     0      0      0
##   39    0     0     0       0     0      0      0
##   40    0     0     0       0     0      0      1
##   41    0     0     0       0     0      0      0
##   42    0     0     0       0     0      0      0
##   43    0     0     0       0     0      0      0
##   44    0     0     0       0     0      0      0
##   45    0     1     0       1     0      0      0
##   46    0     0     0       0     0      0      0
##   47    0     0     0       0     0      0      0
##   48    0     0     0       0     0      0      0
##   49    0     0     0       0     0      0      0
##   50    0     0     0       0     0      0      0
##   51    0     0     0       0     0      0      0
##   52    0     0     0       0     0      0      0
##   53    0     0     0       0     1      0      0
##   54    1     1     0       0     0      0      5
##   55    0     0     0       0     0      0      0
##   56    1     0     0       0     0      0      0
##   57    0     0     0       0     0      0      0
##   58    0     0     0       0     0      0      0
##   59    0     0     0       0     1      0      0
##   60    1     0     0       0     0      0      0
##   61    0     0     0       0     0      0      0
##   62    0     1     0       0     0      1      0
##   63    0     0     0       0     2      0      0
##   64    0     0     0       0     0      0      0
##   65    0     0     0       0     0      0      0
##   66    0     0     0       0     0      0      0
##   67    0     0     0       0     0      0      0
##   68    0     0     0       0     0      0      0
##   69    0     1     0       0     0      1      1
##   70    0     0     0       0     0      0      0
##   71    0     0     0       0     0      0      0
##   72    0     0     0       0     0      0      0
##   73    0     0     0       0     0      0      0
##   74    0     0     0       0     0      0      0
##   75    0     0     0       0     0      0      0
##   76    0     0     0       0     0      0      0
##   77    0     0     0       0     0      0      0
##   78    0     0     0       0     0      0      0
##   79    0     0     0       0     0      1      0
##   80    0     0     0       0     0      0      0
##   81    0     2     0       2     0      0      0
##   82    0     0     0       0     0      0      0
##   83    0     1     0       0     0      0      0
##   84    0     0     0       0     0      0      0
##   85    0     0     0       0     2      0      0
##   86    0     0     0       0     1      0      0
##   87    1     0     0       0     2      0      0
##   88    0     0     0       0     0      0      0
##   89    0     0     0       0     0      0      0
##   90    0     0     0       0     0      0      0
##   91    0     0     0       0     1      0      0
##   92    0     1     0       0     0      0      0
##   93    0     0     0       0     0      0      0
##   94    0     0     0       0     0      0      0
##   95    0     0     0       0     0      0      2
## 96   0    0    0     0    3     0     0
## 97   0    0    0     0    0     0     0
## 98   0    0    0     0    0     0     0
## 99   0    0    0     0    1     0     0
## 100  0    0    0     0    0     0     0
## 101  0    1    0     0    0     0     0
## 102  0    0    0     0    0     0     0
## 103  0    0    0     0    0     0     0
## 104  0    0    0     0    0     0     0
```

```
#colSums(y)
critical_words <- matrix(data = NA, ncol = 7, nrow = 2)
colnames(critical_words) <- c('lie',   'accus',   'putin', 'assault',   'women',  'danger',  'racist')
rownames(critical_words) <- c('June', 'November')
critical_words[1,] <- colSums(y)
```

I now apply the same analysis and methods to november:

```
# Due to insufficient memory on my computer, I only use the last 2.5 weeks before the election as my ba

novem_dat <- Corpus(VectorSource(nov_sam[,1])) # loading the data as corpus
novem_dat <- tm_map(novem_dat, removeWords, c("Donald", "Trump", "words", "highlight", "video")) # Remo
novem_dtm <- DocumentTermMatrix(novem_dat,
           control = list(tolower = TRUE,
                          removePunctuation = TRUE,
                          removeNumbers = TRUE,
                          stopwords = TRUE,
                          stemming=TRUE))




freq_full_nov <- colSums(as.matrix(novem_dtm))
set.seed(1)
wordcloud(names(freq_full_nov), freq_full_nov, min.freq = 150, colors=brewer.pal(6,"Dark2"))
```



```
?wordcloud
freq_full_june[1:30]
```

```
##    â bruce      â oper      âadio    abandon       abat      abbey 
##          1           1          1          8          1          1 
##        abc      abhorr       abil        abl      abort     abras 
##          3           1          8          6          2          1 
##     absenc     absolut     absorb   abstract     absurd       abus 
##          1           6          1          1          2          2 
##       abut      academ    acceler     accent     accept     access 
##          1           1          1          1          4          3 
##      accid      acclim   accommod  accompani accomplish     accord 
##          1           1          1          2          2         23
```

Eyeballing the wordcloud shows that coverage was surprisingly normal - 'said',

```
findFreqTerms(novem_dtm, lowfreq=200) # I want the terms that occur at least 200 times.
```

```
##  [1] "also"       "american"   "black"      "call"       "campaign"  
##  [6] "can"        "candid"     "clinton"    "countri"    "day"       
## [11] "democrat"   "elect"      "email"      "even"       "first"     
## [16] "get"        "hillari"    "just"       "know"       "last"      
## [21] "like"       "make"       "mani"       "mrs"        "nation"    
## [26] "new"        "news"       "now"        "obama"      "one"       
## [31] "parti"      "peopl"      "percent"    "polit"      "poll"      
## [36] "presid"     "presidenti" "public"     "race"       "republican"
## [41] "said"       "say"        "senat"      "show"       "state"     
## [46] "support"    "think"      "time"       "two"        "use"
## [51] "vote"       "voter"      "want"       "way"        "week"
## [56] "white"      "will"       "work"       "year"
```

Eyeballing these results again indicates that the coverage is somewhat neitral. Words that might be associated negatively with Trump, i.e. ‘women’, ‘white’ and ‘black’ are present, but those are perhaps the only ones.

Surprisinly, comparing November to June does not yield that different rests, which indicates that perhaps the articles were not that critical. Another, and probably more likely, interpretation is that the articles were more critical but that the increasing levels on criticism cannot be accurately grasped by using a freqcuency-based approach.

Even though the most frequent words are not overly critical, I am still interested in seeing the occurence of specific words. The following words are an initial subset of words that might be of interest:

```
length(novem_dtm)
```

```
## [1] 6
```

```
z <- inspect( novem_dtm[dimnames(novem_dtm)$Docs, c("lie", "accus", "putin", "assault", "women", "dange
```

```
## <<DocumentTermMatrix (documents: 200, terms: 7)>>
## Non-/sparse entries: 212/1188
## Sparsity           : 85%
## Maximal term length: 7
## Weighting          : term frequency (tf)
##
##      Terms
## Docs  lie accus putin assault women danger racist
##   1     0     0     0       0     1      0      0
##   2     0     0     0       0     0      0      0
##   3     0     0     0       0     0      0      0
##   4     0     0     0       0     0      1      1
##   5     0     0     0       1     0      0      0
##   6     1     0     0       0     0      0      0
##   7     3     0     0       0     1      0      1
##   8     0     3     0       5    12      0      0
##   9     0     0     0       0     2      0      0
##   10    0     1     0       0     1      1      1
##   11    0     1     0       0     0      0      0
##   12    0     0     0       0     1      0      1
##   13    0     0     0       0     0      0      0
##   14    0     2     2       0     0      3      0
##   15    0     0     0       0     0      0      0
##   16    0     1     0       1     1      0      0
##   17    0     1     2       0     0      0      0
##   18    0     0     0       0     0      0      0
##   19    0     0     0       0     1      0      0
##   20    0     0     0       0     0      0      0
##   21    0     0     0       0     0      0      0
##   22    0     0     0       0     1      0      0
##   23    0     0     0       0     0      0      0
##   24    0     0     0       0     0      0      0
##   25    0     2     0       1     0      0      0
##   26    0     0     0       0     1      0      0
##   27    0     0     0       0     1      0      1
##   28    0     0     0       0     1      0      0
##   29    0     1     0       1     2      0      0
##   30    0     0     0       0     0      0      0
##   31    0     0     0       0     0      0      0
##   32    0     0     0       1     1      0      0
##   33    0     0     0       0     0      0      0
##   34    0     2     0       1     0      0      2
##   35    0     0     0       0     0      0      0
##   36    0     0     0       1     1      0      0
##   37    0     0     0       0     1      0      0
##   38    5     0     0       0     0      0      0
##   39    0     0     0       0     0      0      0
##   40    0     0     0       0     0      0      0
##   41    0     0     0       0     1      0      0
##   42    0     0     0       1     1      1      0
##   43    0     0     0       2     0      0      0
##   44    0     1     0       0     0      0      0
##   45    0     0     0       0     0      0      0
##   46    0     1     0       1     2      0      0
##   47    1     1     0       0     0      0      0
##   48    0     0     0       0    11      0      0
##   49    0     0     0       0     4      0      0
##   50    0     0     0       0     1      0      0
##   51    0     0     0       0     0      0      0
##   52    0     0     0       0     0      0      0
##   53    0     0     1       0     0      0      0
##   54    2     2     0       2     1      0      0
##   55    0     0     0       0     0      0      0
##   56    0     0     0       0     0      0      0
##   57    0     0     0       0     0      0      0
##   58    0     0     0       1     3      0      0
##   59    0     0     0       0     1      0      0
##   60    1     0     0       0     0      0      0
##   61    2     0     0       0     0      0      5
##   62    0     0     0       0     0      0      0
##   63    0     0     0       0     0      0      0
##   64    0     0     0       0     2      0      0
##   65    0     0     0       0     0      0      0
##   66    2     0     0       0     0      0      0
##   67    0     0     0       0     2      1      0
##   68    0     1     0       0     6      0      0
##   69    0     0     0       0     0      0      0
##   70    0     0     0       0     0      0      0
##   71    1     0     0       0     0      0      0
##   72    0     0     0       0     0      3      0
##   73    0     0     0       0     0      0      0
##   74    0     0     0       1     0      0      0
##   75    0     0     0       0     0      0      0
##   76    0     0     0       1     5      0      0
##   77    1     2     0       1     3      0      1
##   78    1     0     0       0     1      0      0
##   79    1     0     0       0     0      1      0
##   80    0     0     0       0     0      0      0
##   81    0     0     0       0     0      0      0
##   82    0     0     0       0     4      0      0
##   83    0     2     0       1     0      0      0
##   84    0     2     0       0     0      0      0
##   85    0     1     0       0     1      0      0
##   86    0     0     0       0     3      0      0
##   87    0     0     0       0     0      0      0
##   88    0     0     0       0     0      0      0
##   89    0     0     0       0     3      0      0
##   90    0     0     0       0     0      0      0
##   91    0     0     0       0     1      0      1
##   92    0     0     0       0     0      0      0
##   93    0     0     0       0     0      0      0
##   94    0     0     0       0     0      0      0
##   95    0     0     0       0     0      0      0
##   96    0     0     0       1     0      0      0
##   97    0     0     0       0     0      0      0
##   98    0     1     0       0     0      0      0
##   99    0     0     0       0     0      1      0
##   100   0     0     0       0     4      0      0
##   101   0     0     0       0     0      1      0
##   102   0     0     0       0     0      0      0
##   103   0     0     0       0     0      0      0
##   104   0     0     0       1     1      0      0
##   105   0     1     0      20     0      0      0
##   106   0     0     0       2     2      0      0
##   107   0     0     0       0     0      0      0
##   108   0     1     0       1     0      0      0
##   109   0     1     0       0     0      0      0
##   110   0     0     0       0     6      0      0
##   111   0     0     0       0     1      0      0
##   112   0     0     0       0     0      0      0
##   113   0     0     0       0     0      1      0
##   114   0     0     0       0     0      1      0
##   115   1     0     0       1     1      0      0
##   116   0     0     0       0     0      0      0
##   117   0     0     0       0     0      1      0
##   118   0     0     1       0     0      0      0
##   119   0     0     0       0     0      0      0
##   120   0     0     0       0     5      0      0
##   121   0     1     0       0     0      0      0
##   122   0     1     0       0     0      0      0
##   123   0     0     0       0     6      0      0
##   124   0     0     0       0     0      0      0
##   125   0     0     1       0     0      0      1
##   126   5     0     0       0     1      0      0
##   127   0     0     0       0     0      0      0
##   128   0     1     0       0     1      0      0
##   129   0     0     0       0     6      0      0
##   130   1     0     0       0     0      0      0
##   131   0     1     0       0     1      0      0
##   132   0     0     1       0     0      0      0
##   133   0     4     0       0     0      1      0
##   134   1     3     0       0     5      0      0
##   135   0     0     0       0     4      0      0
##   136   0     2     0       0     0      0      0
##   137   0     3     0       0     0      0      0
##   138   0     0     0       0     0      0      0
##   139   0     0     0       0     0      0      0
##   140   0     0     0       1     0      0      0
##   141   1     1     0       1     2      0      0
##   142   0     0     0       0     0      0      0
##   143   0     2     0       0     0      0      0
##   144   0     2     0       0     0      1      0
##   145   1     0     0       0     0      0      0
##   146   0     0     0       0     1      0      0
##   147   0     0     0       1     1      0      0
##   148   0     0     0       0     0      0      0
##   149   0     1     0       0     0      0      0
##   150   0     0     0       0     0      2      0
##   151   0     1     0       0     0      1      0
##   152   0     0     0       0     0      2      0
##   153   0     0     0       0     1      0      0
##   154   0     0     0       0     0      1      0
##   155   0     0     0       0     0      0      0
##   156   0     0     0       0     5      0      0
##   157   0     0     0       0     1      0      0
##   158   0     1     0       0     0      0      0
##   159   1     0     0       0     0      0      0
##   160   0     0     0       0     0      0      0
##   161   0     3     0       0     7      0      0
##   162   0     0     0       0     0      0      0
##   163   0     1     0       1     1      0      0
##   164   0     0     0       0     0      0      0
##   165   0     0     0       0     0      1      1
##   166   0     0     0       0     0      0      0
##   167   0     1     0       0     0      0      0
##   168   0     0     0       0     0      0      0
##   169   0     0     0       0    16      0      0
##   170   0     1     0       0     0      0      0
##   171   1     0     0       1     2      0      0
##   172   0     0     0       0     0      0      0
##   173   0     0     0       0     0      0      0
##   174   0     0     0       0     0      0      0
##   175   0     0     0       1     0      0      0
##   176   0     2     0       0     3      0      0
##   177   0     1     1       0     1      0      0
##   178   0     0     0       0     1      0      0
##   179   0     0     0       0     0      0      0
##   180   0     1     0       1     3      1      0
##   181   0     0     0       0     0      0      0
##   182   0     0     0       0     0      0      0
##   183   0     0     0       0     0      0      0
##   184   0     0     0       0     0      0      0
##   185   0     1     0       0     0      0      0
##   186   0     0     0       0     0      0      0
##   187   0     0     0       0     4      0      0
##   188   0     0     0       1     0      0      0
##   189   0     0     0       0     0      0      0
##   190   0     0     0       0     0      0      1
##   191   0     1     0       0     0      0      2
##   192 0     0     0       0     3      0      0
##   193 0     0     0       0     0      0      0
##   194 0     0     0       0     0      0      0
##   195 0     0     0       0     1      0      0
##   196 1     6     2       0     1      2      0
##   197 0     1     0       0     9      0      1
##   198 0     0     0       0     0      0      0
##   199 0     0     0       0     0      0      0
##   200 0     0     0       1     0      0      0
```

```
# colSums(z) # blanked out due to space issues
critical_words[2,] <- colSums(z)
critical_words
```

```
##          lie accus putin assault women danger racist
## June       8    14     1       4    24      3     14
## November  34    74    11      58   192     28     20
```

The table shows that some words, especially assault, women, and dager became more frequent over time.

CONCLUSION

In conclusion, I find on basis of my analysis of snippet sentiment that coverage does indeed get more negative over time. I also find that coverage increases in sentiment broadly speaking, meaning that higher intensity of sentiments across categories (fear, anger, surprise etc) increases.

Looking at the specific frequencies does not enlighten which changes of words might drive this development. Across months, snippets seem to be characterized by fairly similar terms.

Looking at full months yields contradictory findings, at least in terms of sentiments, since the last month appears to be more positive in the coverage of Trump. Looking at words frequencies confirms the picture from the snippets, since June and November appear to be characterized by fairly similar terms also in the full-length articles.

I cannot on basis of my findings conclude anything final since the graphs of total sentiment scores seem to show different results for the snippets and the full articles. Moreover, the table of critical words shows that these words exibit a large increase in frequency. Further analysis is clearly needed if I wish to make final conclusions.

One reason for the seemingly inconsistent results might be a matter of signal being lost in the noise - I have very, very large files with many words to analyze. For this reason, normal words like 'elect', 'campaign' etc simply drown out the more interesting critical words, that my table shows do actually increase in frequency.

To get at this, I would need to get full-length articles and analyze them after removing the 'noise'. But determining whats noise and what should rather be counted as part of the normal election coverage is tricky and would require careful analysis.

Another reason for the inconsistent results is the fact that I cast a very broad net in my article search. I find all articles involving Trump, and therefore also articles that do not specifically cover him but instead mention him in passing. These articles obviously bias my results, and would need to be removed.