Big Data: Final Project

Angel Aliseda, Javier Patino, Fernando Regalado, Felipe Salazar

5/12/2020

Contents

1 Retrieving the tweets from the accounts of Donald Trump and Joe Biden

1.1 Data transformation

```
# Import Twitter database with tweets from Donald Trump and Joe Biden

vignette("auth", package = "rtweet")

## working with the stream
vignette("stream", package = "rtweet")

candidates <- get_timelines(
  user = c("realDonaldTrump", "JoeBiden"),
  n = 10000
)

table(candidates$screen_name)

candidates %>%
  group_by(screen_name) %>%
  ts_plot(by = "week")

candidates <- apply(candidates,2,as.character)

#Export the final dataframe to a csv file
write.csv(candidates, "Candidates_tweets.csv")
```

```
# Load database from repository
data <- read.csv(file = "Candidates_tweets.csv") %>%
  mutate(text = as.character(text))

var_list <- names(data)
#View(var_list)
dim(data)
```

```
## [1] 6460   91
```

```
# Transformation -> one-word-per-row
data_tokens <- data %>%
  # tokenizing text from tweets
  unnest_tokens(output = text,
                input = text) %>%
  # remove numbers and string to lowercase
  filter(!str_detect(text, "^[0-9]*$")) %>%
  mutate(text = str_to_lower(text)) %>%
  # removing stop words
  anti_join(stop_words, by = c("text" = "word")) %>%
  # stemming words to avoid redundancy
  mutate(word_stem = wordStem(text)) %>%
  # merging with SENTIMENT DICTIONARY
  inner_join(get_sentiments("bing"), by = c("text" = "word"))

dim(data_tokens)
```

```
## [1] 12721    93
```

```
# Transformation -> one-tweet-per-row matrix
(data_matrix <- data_tokens %>%
  # get count of each token in each document
  count(X, word_stem) %>%
  # creating a document-term matrix with all features and tf weighting
  cast_dtm(document = X, term = word_stem, value = n))
```

```
## <<DocumentTermMatrix (documents: 4902, terms: 1176)>>
## Non-/sparse entries: 12293/5752459
## Sparsity           : 100%
## Maximal term length: 14
## Weighting          : term frequency (tf)
```

2 Topic Modelling

2.1 Topic Analysis

A good start for our exploratory analysis of tweets is trying to find out if there are some recognizable patterns in the way both personalities write. Thus, we can try to discover and identify some topics under which some words are more likely to be employed than others. This assessment not only allow us to cluster tweets under topics but also, we can test if these are useful to predict the impact that tweets could have in social media using measures such the number of retweets.

```
# Data matrix for Trump
trump_tm <- data_tokens %>%
  mutate(screen_name = as.character(screen_name)) %>%
  # filtering word "trump"
  filter(screen_name == "realDonaldTrump",
         word_stem != "trump") %>%
  count(X, word_stem) %>%
  cast_dtm(document = X,
           term = word_stem,
           value = n) %>%
  removeSparseTerms(sparse = .99)

trump_tm <- trump_tm[unique(trump_tm$i),] # removing tweets with no terms remaining

# Data matrix for Biden
biden_tm <- data_tokens %>%
  mutate(screen_name = as.character(screen_name)) %>%
  # filtering word "trump"
  filter(screen_name == "JoeBiden",
         word_stem != "trump") %>%
  count(X, word_stem) %>%
  cast_dtm(document = X,
           term = word_stem,
           value = n) %>%
  removeSparseTerms(sparse = .99)

biden_tm <- biden_tm[unique(biden_tm$i),] # removing tweets with no terms remaining
```

The analysis for Trump includes 1362 tweets containing 47 words after removing stop words and stemming. Likewise, data for Biden includes 1907 tweets and 56 words after the same data cleaning. Given this information, we can hypothesize different numbers of topics that underlie behind text and decide the final number based on the highest log Bayes Factor (BF), an information criterion. It is important to notice that by clustering for text, we assume that documents (tweets) are drawn from a multinomial mixture distribution in which each word has a different probability depending on the topic and each tweet itself is a mixture or combination of these topics.

2.1.1 Topic analysis for Trump's tweets:

```
# Topics for Trump
trump_topics <- topics(trump_tm, K=(2:8), verb=10)
```

```
##
## Estimating on a 1362 document collection.
## Fit and Bayes Factor Estimation for K = 2 ... 8
## log posterior increase: 22.2, 29.9, 18.9, 47.8, 46.1, 75.3, 70.2, 24.1, 3.7, 1.6, 13.4, 10.1, 6.8, 18
```

```
## log BF( 2 ) = 7576.17 [  22 steps, disp = 1.03  ]
## log posterior increase: 220.3, 38.4, 8.7, 2.3, 1.1, 0.2, 0.2, 0.1, done. (L = -8371.9)
## log BF( 3 ) = 11175.68 [  9 steps, disp = 0.9  ]
## log posterior increase: 117.8, 112.6, 45.9, 7.8, 2.6, 15.5, 11.6, 2.5, 0.5, 0, done. (L = -8549.9)
## log BF( 4 ) = 15273.3 [  11 steps, disp = 0.76  ]
## log posterior increase: 47.6, 143.5, 46.8, 9.9, 3, 1.1, 0.6, 0.5, 0.7, 10.6, 12, 3.3, 0.5, 0, done.
## log BF( 5 ) = 19924.43 [  15 steps, disp = 0.64  ]
## log posterior increase: 41.2, 82, 138.2, 46.1, 12.9, 3.1, 0.6, 0, done. (L = -8636.8)
## log BF( 6 ) = 25051.14 [  9 steps, disp = 0.63  ]
## log posterior increase: 67.3, 55, 37.7, 18.7, 3.2, 0.9, 0.1, 0, done. (L = -8718.5)
## log BF( 7 ) = 20512.12 [  9 steps, disp = 0.63  ]
## log posterior increase: 30.6, 157.7, 37.5, 6.8, 2.3, 0.7, 0.1, done. (L = -8707.5)
## log BF( 8 ) = 19306.06 [  8 steps, disp = 0.57  ]
```

```
summary(trump_topics, n=10)
```

```
##
## Top 10 phrases by topic-over-null term lift (and usage %):
##
## [1] 'readi', 'wow', 'issu', 'congratul', 'lead', 'approv', 'fast', 'crisi', 'happi', 'break' (21.9)
## [2] 'crime', 'love', 'incred', 'strong', 'support', 'endors', 'safe', 'hard', 'wow', 'wors' (19.8)
## [3] 'win', 'fake', 'corrupt', 'bad', 'fail', 'enemi', 'radic', 'wors', 'li', 'wow' (19)
## [4] 'kill', 'li', 'crazi', 'protect', 'wors', 'hard', 'radic', 'emerg', 'relief', 'issu' (14.2)
## [5] 'free', 'lost', 'critic', 'proud', 'sick', 'won', 'fraud', 'wors', 'readi', 'wow' (13)
## [6] 'collus', 'hoax', 'illeg', 'peac', 'help', 'wrong', 'fail', 'radic', 'li', 'happi' (12.1)
##
## Log Bayes factor and estimated dispersion, by number of topics:
##
##             2        3        4        5        6        7        8
## logBF 7576.17 11175.68 15273.30 19924.43 25051.14 20512.12 19306.06
## Disp     1.03     0.90     0.76     0.64     0.63     0.63     0.57
##
## Selected the K = 6 topic model
```

Taking into account up to 8 different topics for Trump, BF criteria selects only 6 topics. The following table contains the top ten words with higher probability in each topic:

```
t1 <- rownames(trump_topics$theta)[order(trump_topics$theta[,1], decreasing=TRUE)[1:10]]
t2 <- rownames(trump_topics$theta)[order(trump_topics$theta[,2], decreasing=TRUE)[1:10]]
t3 <- rownames(trump_topics$theta)[order(trump_topics$theta[,3], decreasing=TRUE)[1:10]]
t4 <- rownames(trump_topics$theta)[order(trump_topics$theta[,4], decreasing=TRUE)[1:10]]
t5 <- rownames(trump_topics$theta)[order(trump_topics$theta[,5], decreasing=TRUE)[1:10]]
t6 <- rownames(trump_topics$theta)[order(trump_topics$theta[,6], decreasing=TRUE)[1:10]]

kable(cbind(t1,t2, t3, t4, t5, t6),
      col.names = c("Topic 1", "Topic 2", "Topic 3", "Topic 4", "Topic 5", "Topic 6"),
      align = "c")
```

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
viru	support	fake	protect	critic	help
relief	endors	win	hard	proud	wrong
break	strong	bad	crazi	sick	illeg
honor	love	corrupt	kill	free	hoax
approv	incred	enemi	wors	won	peac
crisi	safe	radic	li	fraud	collus

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
fast	crime	fail	radic	lost	radic
congratul	hard	viru	relief	viru	fail
lead	honor	sick	endors	support	break
happi	proud	approv	emerg	love	fake

Even when these 6 topics contain a variety of words, we can clearly identify that topic 1 and 3 have a positive connotation whereas topic 2, 4, 5 and 6 have are the opposite and related to a more contentious position. This interpretation does not differ much from the president's polarized speech commonly used.

The following wordcloud depicts the two most frequent topics in Trump's speech:

```
par(mfrow=c(1,2))
wordcloud(row.names(trump_topics$theta),
          random.order = FALSE,
          freq=trump_topics$theta[,1],
          #min.freq=0.0001,
          col="maroon")
wordcloud(row.names(trump_topics$theta),
          random.order = FALSE,
          freq=trump_topics$theta[,2],
          #min.freq=0.0001,
          col="maroon")
```

2.1.2 Topic analysis for Biden's tweets:

```
# Topics for Biden
biden_topics <- topics(biden_tm, K=(2:8), verb=10)
```

```
##
## Estimating on a 1907 document collection.
## Fit and Bayes Factor Estimation for K = 2 ... 8
## log posterior increase: 352.9, 181.7, 81.8, 29, 9, 13.5, 21.6, 14.6, 5.1, 1.9, 1.4, 1.4, 9.9, 7.1, 3
## log BF( 2 ) = 10978.29 [  31 steps, disp = 1.04  ]
## log posterior increase: 225.7, 135.8, 92.6, 45.5, 29.1, 28.7, 17.7, 2.1, 0.6, 0.2, 0.2, 0.1, done. (1
## log BF( 3 ) = 15832.35 [  13 steps, disp = 0.88  ]
```

```
## log posterior increase: 138.9, 254, 133.4, 42.9, 13.3, 5.5, 3.6, 6.4, 2.2, 7.4, 42.6, 23.6, 2.9, 0.3
## log BF( 4 ) = 21704.36 [  16 steps, disp = 0.83  ]
## log posterior increase: 84.1, 117.1, 97.8, 58.2, 21.2, 15.5, 22.6, 6, 1.9, 0.2, 0.1, 0.1, 0.2, 0.3, (
## log BF( 5 ) = 28193.26 [  23 steps, disp = 0.75  ]
## log posterior increase: 74.3, 79.6, 164.7, 133.2, 42.7, 14.5, 5, 1.4, 0.5, 0.2, 0, done. (L = -13711
## log BF( 6 ) = 35371.84 [  12 steps, disp = 0.73  ]
## log posterior increase: 74.1, 272.7, 86, 27, 14.1, 2, 0.4, 0.1, done. (L = -13626.2)
## log BF( 7 ) = 30821.91 [  9 steps, disp = 0.69  ]
## log posterior increase: 51.8, 78.3, 76.9, 54.1, 19.8, 4.4, 0.6, 0.2, 0.1, 0.1, done. (L = -13676.7)
## log BF( 8 ) = 28345.15 [  11 steps, disp = 0.61  ]
```

```
summary(biden_topics, n=10)
```

```
##
## Top 10 phrases by topic-over-null term lift (and usage %):
##
## [1] 'readi', 'backbon', 'crisi', 'win', 'protect', 'defeat', 'afford', 'assault', 'fail', 'threat' (1
## [2] 'promis', 'fair', 'support', 'respect', 'proud', 'benefit', 'incred', 'digniti', 'backbon', 'urg
## [3] 'corrupt', 'issu', 'abus', 'tough', 'power', 'clean', 'celebr', 'happi', 'strong', 'love' (17.5)
## [4] 'epidem', 'reform', 'hate', 'safe', 'stronger', 'relief', 'wrong', 'courag', 'help', 'free' (16.3
## [5] 'super', 'wealthi', 'lost', 'lead', 'honor', 'hard', 'faith', 'urgent', 'relief', 'backbon' (13.7
## [6] 'danger', 'attack', 'endors', 'fear', 'deni', 'grate', 'peac', 'urgent', 'free', 'tough' (11.8)
##
## Log Bayes factor and estimated dispersion, by number of topics:
##
##              2        3        4        5        6        7        8
## logBF 10978.29 15832.35 21704.36 28193.26 35371.84 30821.91 28345.15
## Disp      1.04     0.88     0.83     0.75     0.73     0.69     0.61
##
## Selected the K = 6 topic model
```

Once again, we evaluate up to 8 topics for Biden's tweets and BF selects 6 topics again. The following table contains the top ten words with higher probability in each topic:

```
b1 <- rownames(biden_topics$theta)[order(biden_topics$theta[,1], decreasing=TRUE)[1:10]]
b2 <- rownames(biden_topics$theta)[order(biden_topics$theta[,2], decreasing=TRUE)[1:10]]
b3 <- rownames(biden_topics$theta)[order(biden_topics$theta[,3], decreasing=TRUE)[1:10]]
b4 <- rownames(biden_topics$theta)[order(biden_topics$theta[,4], decreasing=TRUE)[1:10]]
b5 <- rownames(biden_topics$theta)[order(biden_topics$theta[,5], decreasing=TRUE)[1:10]]
b6 <- rownames(biden_topics$theta)[order(biden_topics$theta[,6], decreasing=TRUE)[1:10]]

kable(cbind(b1,b2, b3, b4, b5, b6),
      col.names = c("Topic 1", "Topic 2", "Topic 3", "Topic 4", "Topic 5", "Topic 6"),
      align = "c")
```

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
crisi	support	love	safe	lead	attack
protect	promis	progress	epidem	hard	fear
afford	proud	won	hate	honor	grate
win	digniti	abus	reform	wealthi	danger
threat	fair	corrupt	stronger	super	peac
defeat	benefit	issu	courag	lost	endors
assault	respect	emerg	wrong	faith	deni
fail	incred	celebr	free	fail	threat
readi	free	clean	help	emerg	afford

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
backbon	backbon	happi	relief	relief	free

In contrast to Trump 'tweets, Biden's topics contain less aggressive words and the tone of speech seems to be more moderate. This time we don't observe strong words such "hoax", "fraud" or "collusion". Moreover, Biden tends to include more positive words in his tweets since these are more frequent across topics.

We can see this graphically in the following word cloud for the two most frequent topics.

```
par(mfrow=c(1,2))
wordcloud(row.names(biden_topics$theta),
          random.order = FALSE,
          freq=biden_topics$theta[,1],
          #min.freq=0.0001,
          col="navy")
wordcloud(row.names(biden_topics$theta),
          random.order = FALSE,
          freq=biden_topics$theta[,2],
          #min.freq=0.0001,
          col="navy")
```

2.2 Topic regression

Now we try to find out if clustering is useful for predicting the impact of tweets measured as the number or retweet or times that users shared the information. For this, we relate topics with variable `retweet_count` in a a second-stage low-D regression and compare results with a regression using all words (our original high-D matrix).

2.2.1 Trump: second-stage low-D regression vs word regression

```
# Regress retweets_counts on omega (matrix with tweets topic weights)
trump_rts <- slice(data, as.numeric(trump_tm$dimnames$Docs))
y <- trump_rts[,"retweet_count"]

trump_topics_reg.cv <- cv.gamlr(trump_topics$omega, y,
                        lambda.min.ratio=10^{-4})

coef(trump_topics_reg.cv) # number of retweets up or down for moving up 10\% weight in that topic
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                seg1
## intercept 19771.43
## 1              .
## 2              .
## 3              .
## 4              .
## 5              .
## 6              .
```

Results for Trump suggest that only topic 2 have a positive effect on number of retweets. Recall that under this topic, aggressive words are likely to appear such "fake", "enemy", "corrupt" or "crazy". Therefore, results are preliminary evidence that suggest that Trump's confrontational tone is effective at least in Twitter. In average, number of Trump's retweets is 1.9771×10^4, but it can increase by 0 if text corresponds to topic 2.

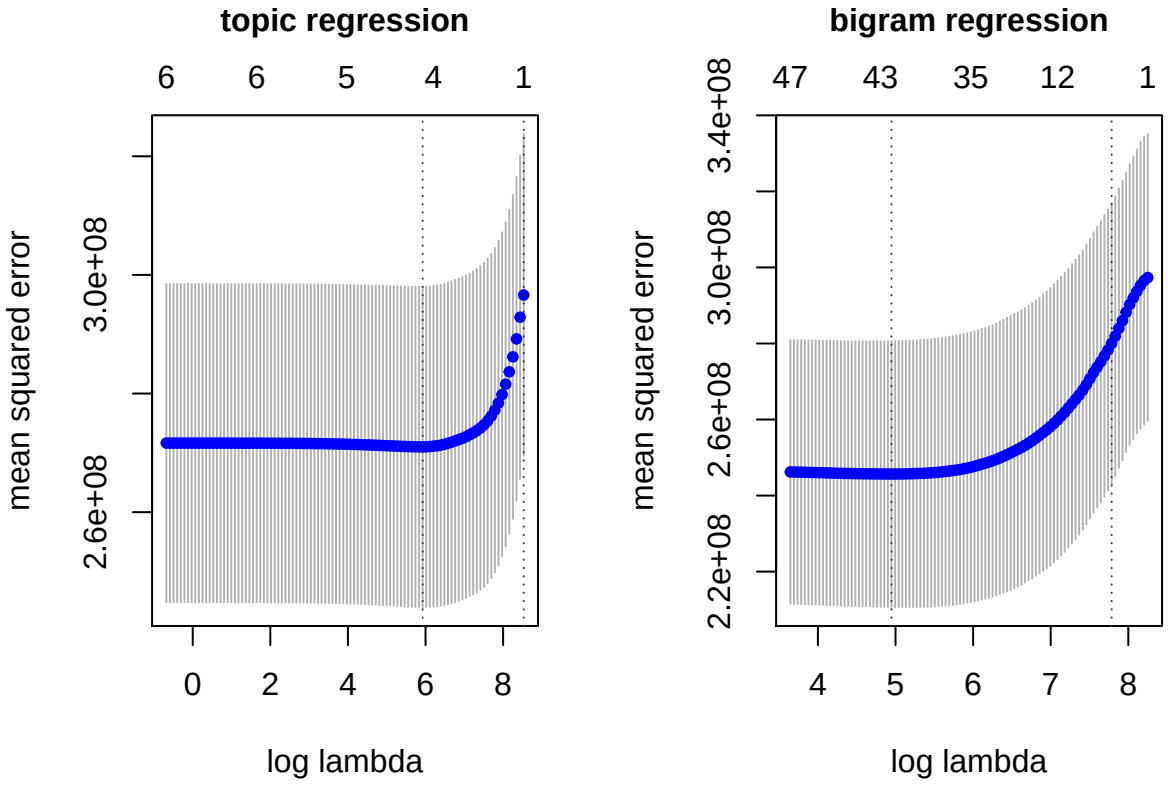
```
## give it the word %s as inputs
trump_words_reg.cv <- cv.gamlr(trump_tm, y)
coef(trump_words_reg.cv)
```

```
## 48 x 1 sparse Matrix of class "dgCMatrix"
##                seg11
## intercept 19236.693
## win              .
## fake       1668.244
## incred           .
## crime            .
## radic      5924.544
## fast             .
## approv           .
## wors             .
## crazi            .
## hard             .
## peac             .
## bad              .
## wrong            .
## hoax             .
## illeg            .
## wow              .
## proud            .
## critic           .
## endors           .
## love             .
## strong           .
## lead             .
## help             .
## safe             .
## corrupt          .
## emerg            .
## honor            .
## protect          .
## relief           .
## won              .
## support    -540.512
## readi            .
## break            .
```

```
## enemi      7739.454
## happi         .
## kill          .
## free          .
## fail          .
## congratul     .
## crisi         .
## issu          .
## li            .
## sick          .
## lost          .
## fraud         .
## collus        .
## viru          .
```

In addition, word regression suggests similar findings. Using high-dimensional matrix, word regression finds that the same words contained in topic 2 have a significant effect on the number of retweets which adds evidence to signal found in previous regression. Moreover, some additional words are added to the list such "relief", "support" or "crisis" but these have a negative effect on diffusion of tweets.

```
par(mfrow=c(1,2))
plot(trump_topics_reg.cv)
mtext("topic regression", font=2, line=2)
plot(trump_words_reg.cv)
mtext("bigram regression", font=2, line=2)
```



Finally, a comparison of the predictive power of these two regressions reveals that topic clustering is less effective than using words. Thus, the later one yields a lower out-of-sample mean squared error which entails a better performance in terms of prediction.

2.2.2 Biden: second-stage low-D regression vs word regression

```
# Regress retweets_counts on omega (matrix with tweets topic weights)
biden_rts <- slice(data, as.numeric(biden_tm$dimnames$Docs))
y <- biden_rts[,"retweet_count"]

# favorite/retweet

biden_topics_reg.cv <- cv.gamlr(biden_topics$omega, y,
                        lambda.min.ratio=10^{-4})

coef(biden_topics_reg.cv) # number of retweets up or down for moving up 10\% weight in that topic
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                seg1
## intercept 3675.601
## 1                .
## 2                .
## 3                .
## 4                .
## 5                .
## 6                .
```

In contrast to Trump's results, for Biden none of the topics are useful to predict number of retweets. This entails that none of the topics contain a signal and therefore, regardless of topic, a Biden's tweet is expected to be retweeted in average about 3676 times.

```
## give it the word %s as inputs
biden_words_reg.cv <- cv.gamlr(biden_tm, y)
coef(biden_words_reg.cv)
```

```
## 57 x 1 sparse Matrix of class "dgCMatrix"
##                 seg9
## intercept 3527.2284
## promis           .
## threat           .
## digniti          .
## incred           .
## love             .
## progress         .
## respect          .
## happi            .
## lead             .
## grate            .
## support          .
## reform           .
## won              .
## honor            .
## faith            .
## hard             .
## defeat           .
## fail      2094.1675
## win              .
## crisi      914.6573
## power            .
```

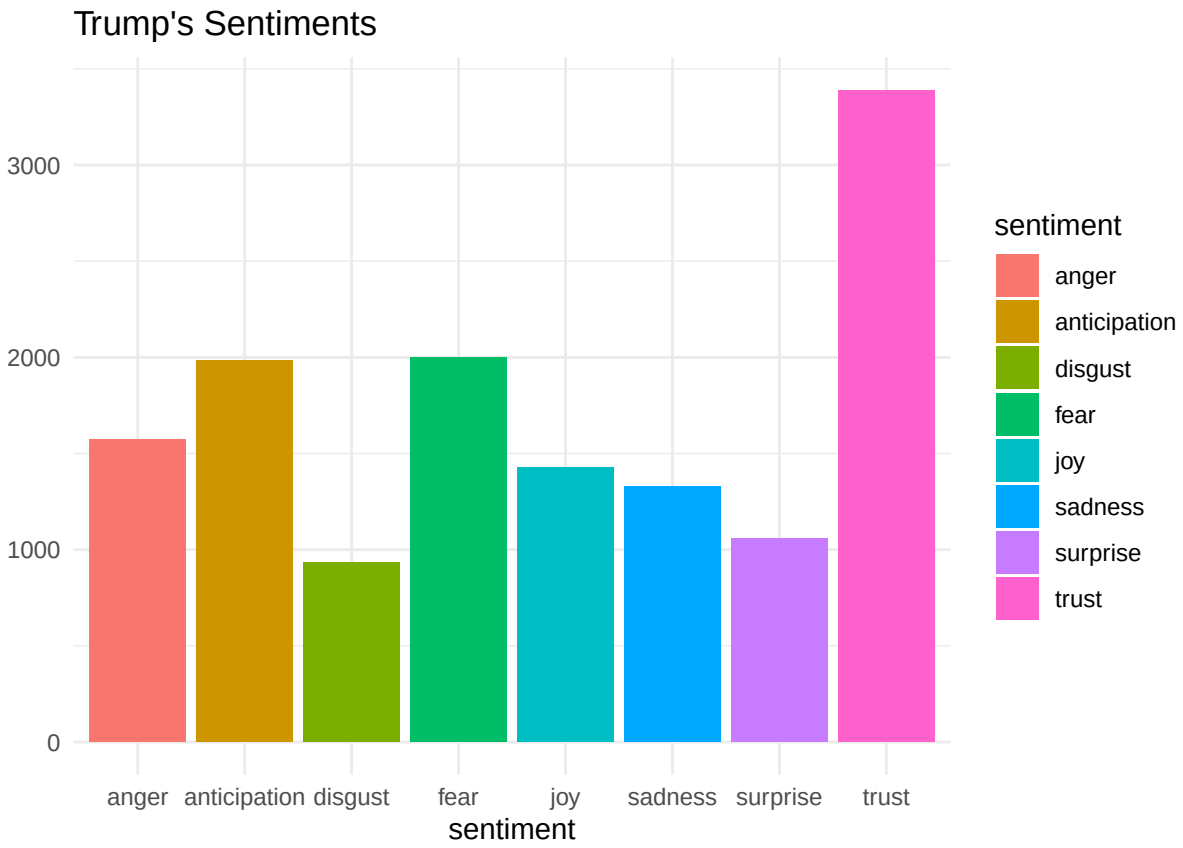
```
## hate         .
## fear         .
## stronger     .
## deni         .
## protect      .
## peac         .
## safe         .
## free         .
## lost         .
## benefit      .
## relief       .
## super        .
## wealthi      .
## tough        .
## readi        .
## celebr       .
## help         .
## fair         .
## epidem       .
## proud        .
## issu         .
## backbon      .
## strong       .
## courag       .
## corrupt      .
## urgent       .
## afford       .
## endors       .
## emerg        .
## attack       .
## danger       .
## wrong        .
## clean        .
## assault      .
## abus         .
```

These findings are consistent with results using words. Regression shows that none of the words have an effect on retweets. Thus, underlying topics do not contain any signal for prediction since words neither. Whereas for Trump the usage of some specific topics and words have an effect on retweets, Biden followers tend to be less sensible to the topic or tone of his speech what could potentially explain the findings.

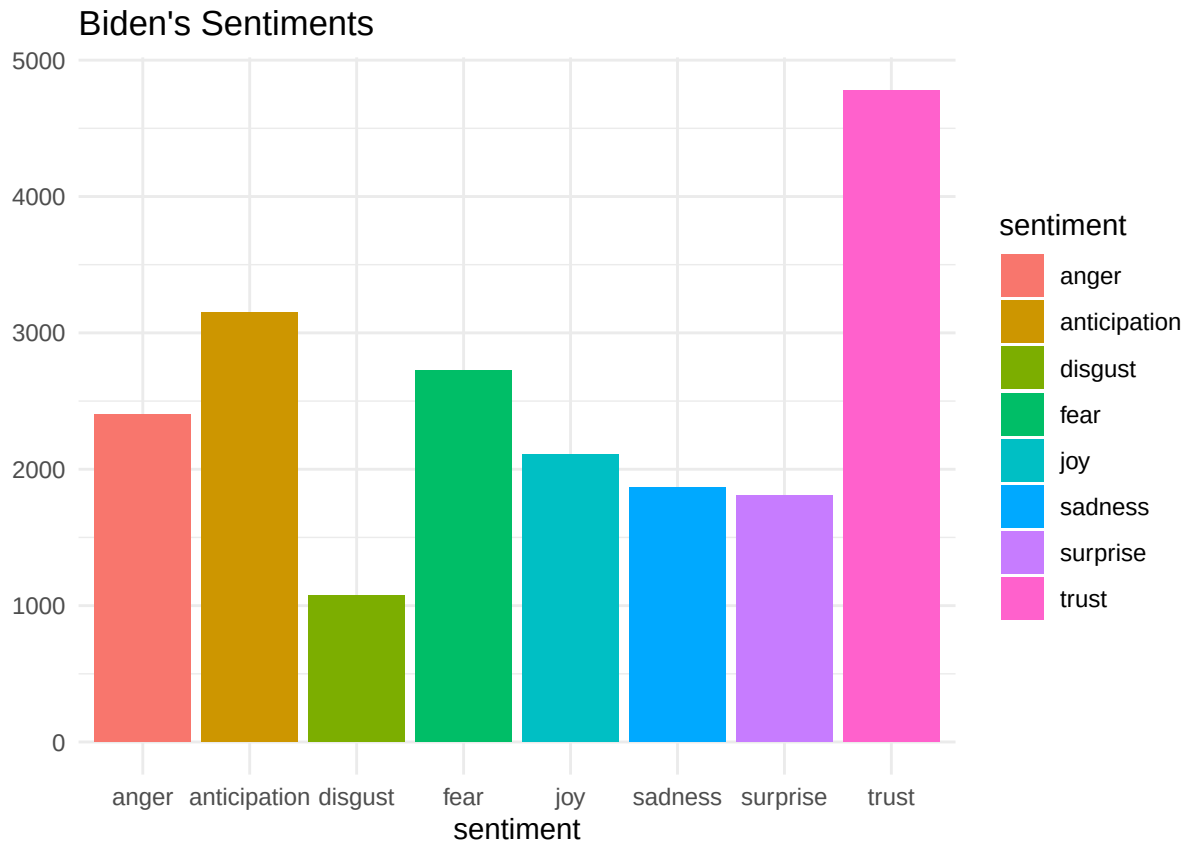
3 Sentiment Analysis

This section shows the sentiment analysis of the tweets of both candidates. For doing this task, we considered the NRC sentiment dictionary.

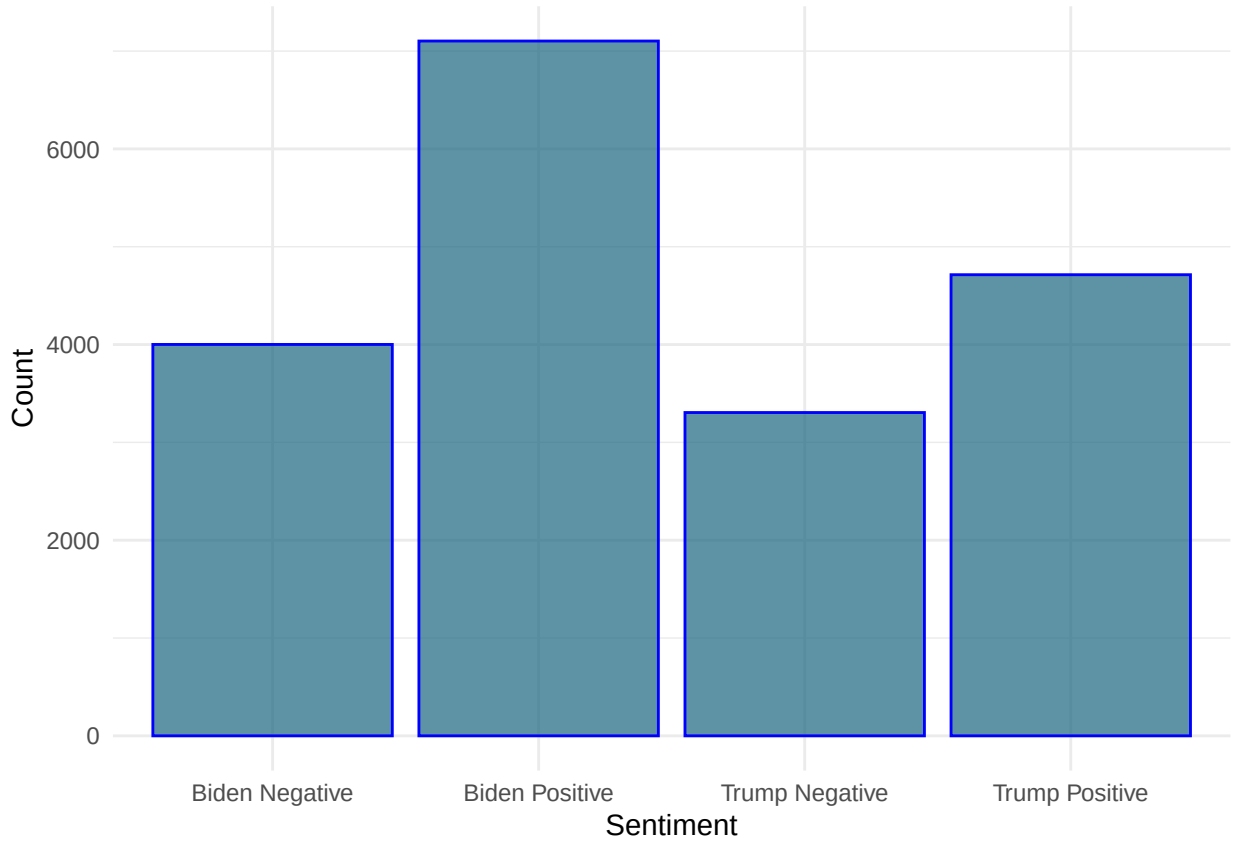
```
#Sentiment analysis for Trump
data_trump <- data[data$screen_name== "realDonaldTrump",]
data_trump$text_plain <- plain_tweets(data_trump$text)
sa_trump <- get_nrc_sentiment(data_trump$text_plain)
#Transposing the dataset
sa1_trump<-data.frame(t(sa_trump))
#Obtaining count
new_sa_trump <- data.frame(rowSums(sa1_trump))
names(new_sa_trump)[1] <- "count"
new_sa_trump <- cbind("sentiment" = rownames(new_sa_trump), new_sa_trump)
rownames(new_sa_trump) <- NULL


#Sentiment analysis for Biden
data_biden <- data[data$screen_name== "JoeBiden",]
data_biden$text_plain <- plain_tweets(data_biden$text)
sa_biden <- get_nrc_sentiment(data_biden$text_plain)
#Transposing the dataset
sa1_biden<-data.frame(t(sa_biden))
#Obtaining count
new_sa_biden <- data.frame(rowSums(sa1_biden))
names(new_sa_biden)[1] <- "count"
new_sa_biden <- cbind("sentiment" = rownames(new_sa_biden), new_sa_biden)
rownames(new_sa_biden) <- NULL

#Plotting the sentiments
qplot(sentiment, data=new_sa_trump[1:8,], weight=count, geom="bar",fill=sentiment)+ggtitle("Trump's Sen
```




```
qplot(sentiment, data=new_sa_biden[1:8,], weight=count, geom="bar",fill=sentiment)+ggtitle("Biden's Sen
```




```
#Ploting negative and positive sentiments
data_plot <- data.frame(
  name=c("Trump Negative", "Biden Negative", "Trump Positive", "Biden Positive"),
  value=c(new_sa_trump[9,]$count, new_sa_biden[9,]$count,
          new_sa_trump[10,]$count, new_sa_biden[10,]$count)
)

ggplot(data_plot, aes(x=name, y=value))+geom_bar(stat="identity", color='blue', fill=rgb(0.1,0.4,0.5,0.
  xlab("Sentiment") +
  ylab("Count")
```



As we can see in the graphs, the distribution of the sentiments are similar. Due to their political nature, the most frequent sentiment is "trust" for both candidates. Surprisingly, when we compare the positive and negative sentiments for both candidates, Joe Biden shows a higher count for the negative sentiment than Donald Trump.

4 LASSO

4.1 Effects of specific words on Favorites and Retweets for Trump and Biden

```
### Dependent variables for the LASSO analysis
## Favorites
# Total number of favorites for Trump for tweets included in term matrix
tweets <- as.numeric(data_matrix$dimnames$Docs)

favs_trump <- data %>%
  select(X, favorite_count) %>%
  filter(X<3215) %>%
  filter(X %in% tweets & favorite_count>0) #original tweets included in the term matrix

# Total number of favorites for Biden for tweets included in term matrix

favs_biden <- data %>%
  select(X, favorite_count) %>%
  filter(X>3214) %>%
  filter(X %in% tweets & favorite_count>0) #original tweets included in the term matrix

## Retweets
# Trump
rt_trump <- data %>%
  select(X, retweet_count) %>%
  filter(X<3215) %>%
  filter(X %in% tweets) #retweets included in the term matrix

# Biden
rt_biden <- data %>%
  select(X, retweet_count) %>%
  filter(X>3214) %>%
  filter(X %in% tweets) #retweets included in the term matrix
```

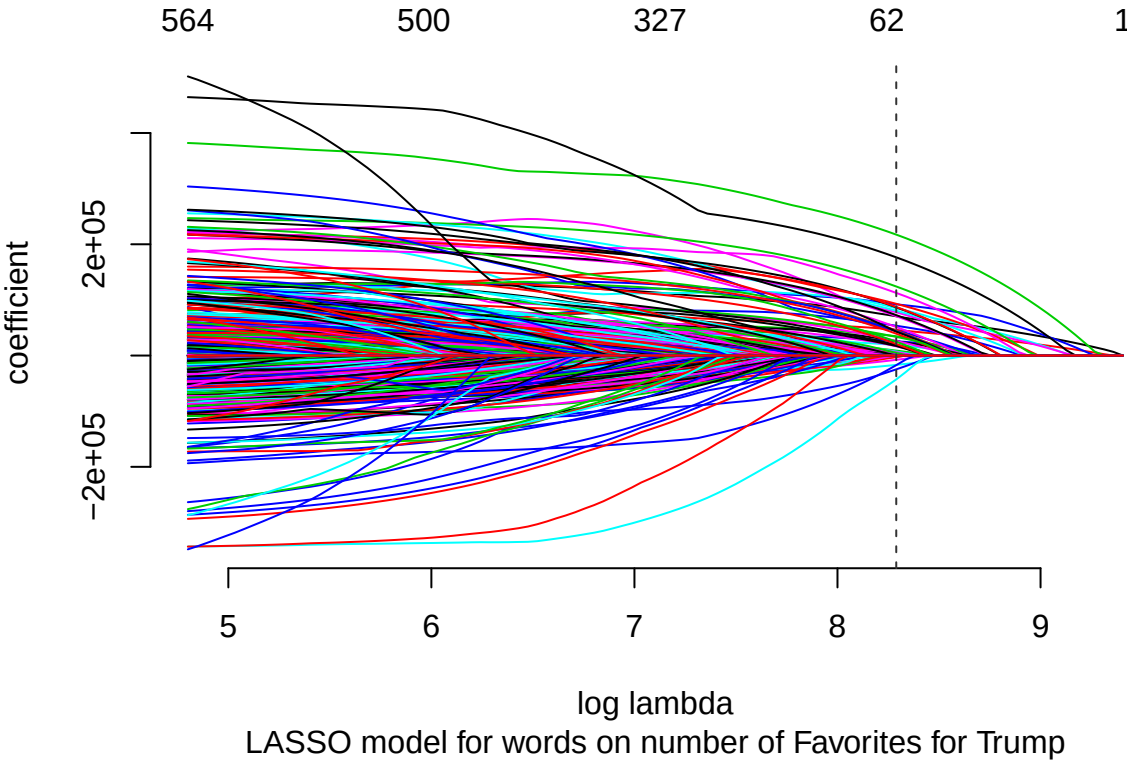
What makes a successful tweet? Are there common words for candidates that gets them more favorites or retweets? In order to answer these questions, we performed a predictive analysis where we looked at whether there are common words that predict if a tweet is succesful. To do this, we used two variables that we obtained from the Twitter database which are total number of Favorites and total number of Retweets.

We performed a LASSO analysis for both candidates and for Favorites and Retweets sepparately. Trump's tweets have a median of 1.003585×10^5 favs and a median of 1.3149×10^4 retweets. On the other hand, Biden's tweets have a median of 6188 favs and 1283.5 retweets. We can see here that, at least in Twitter, candidate Trump is more popular than candidate Biden which is reflection of the amount of time Trump invested in Twitter during his campaign and his presidency.

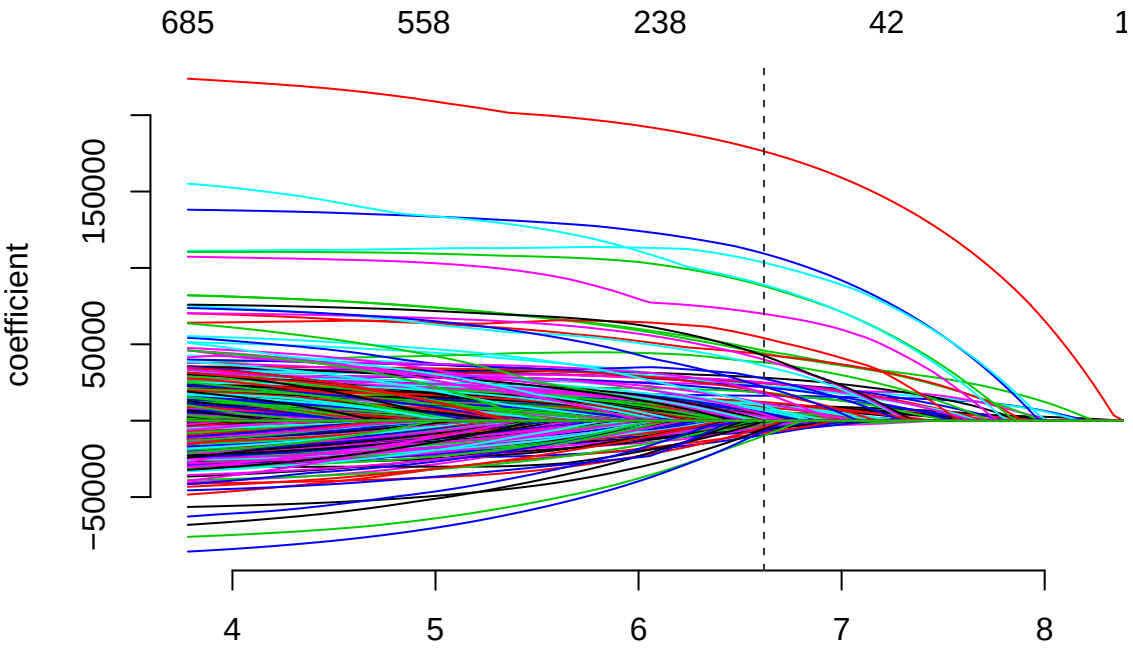
The following plots show the LASSO analysis of the total number of favs and retweets by each word using the matrix we computed for the topic analysis. From the models with the lowest AICc, we obtained the top 10 words with the most positive and most negative effect for retweets and favs for each candidate.

```
### LASSO for total count of Favorites as outcome variable for each candidate

lasso_favs_trump <- gamlr(data_matrix[favs_trump$X,],
                          favs_trump$favorite_count)

lasso_favs_biden <- gamlr(data_matrix[as.character(favs_biden$X),],
                          favs_biden$favorite_count)
```

```
### LASSO for total count of Retweets as outcome variable for each candidate
lasso_rt_trump <- gamlr(data_matrix[rt_trump$X,],
                        rt_trump$retweet_count)

lasso_rt_biden <- gamlr(data_matrix[as.character(rt_biden$X),],
                        rt_biden$retweet_count)

### LASSO plots for each model

plot(lasso_favs_trump,
     sub = "LASSO model for words on number of Favorites for Trump")
```

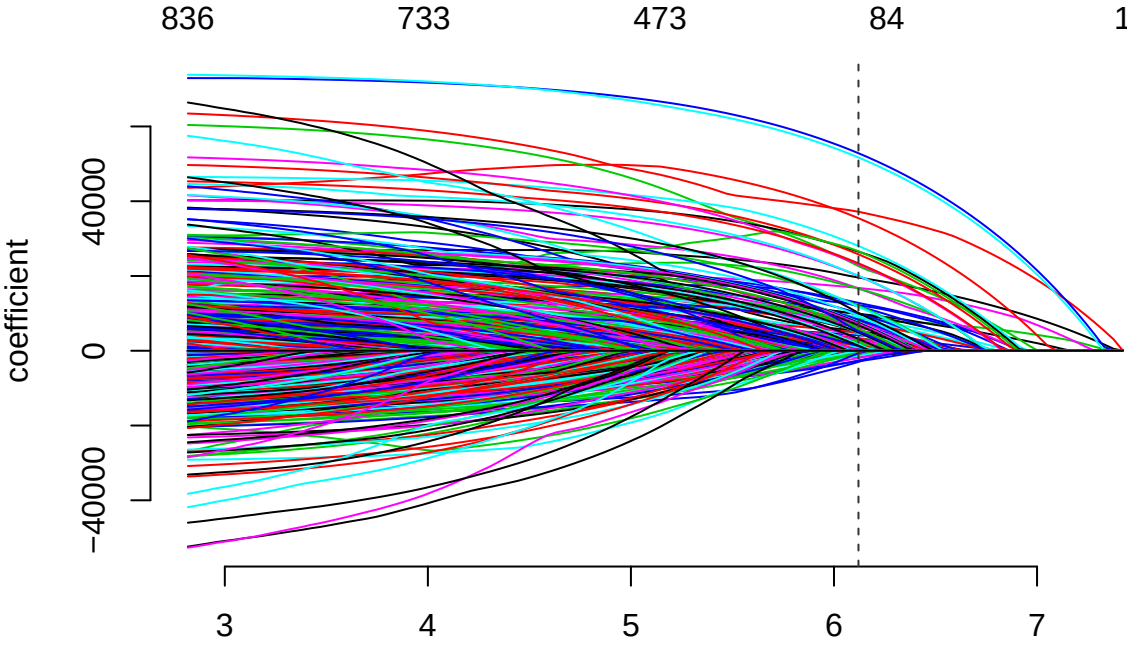



```
plot(lasso_favs_biden,
     sub = "LASSO model for words on number of Favorites for Biden")
```



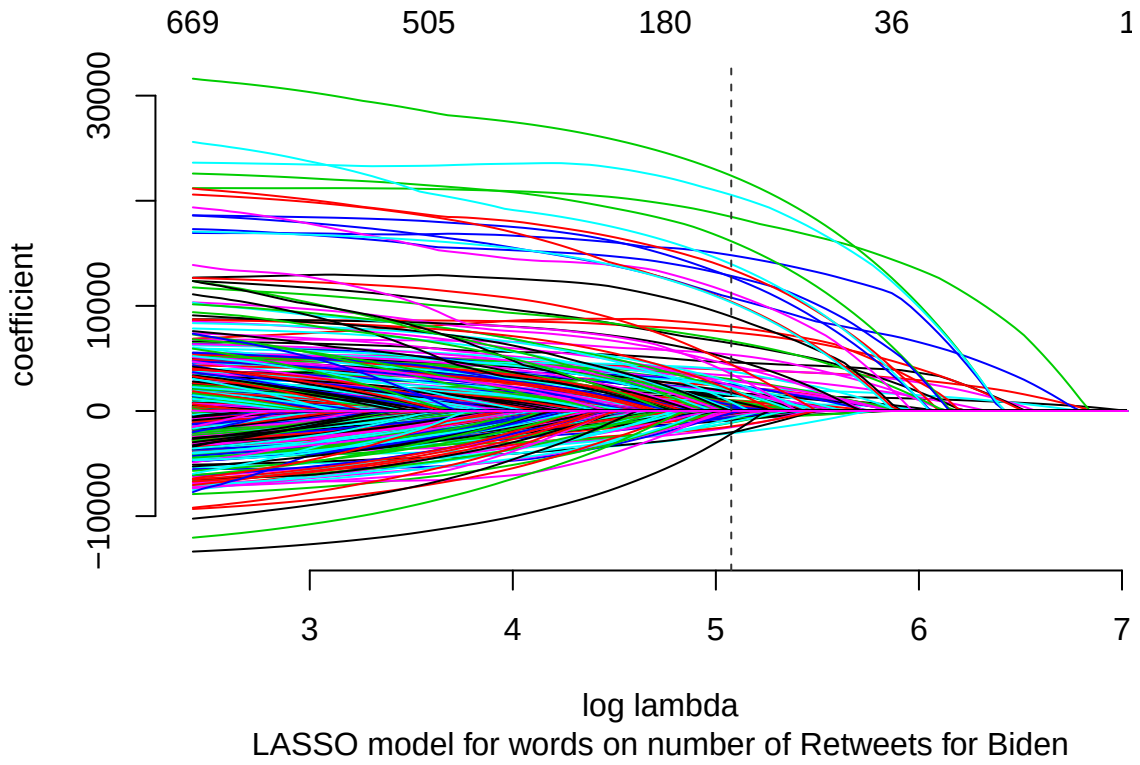
LASSO model for words on number of Favorites for Biden

```
plot(lasso_rt_trump,
     sub = "LASSO model for words on number of Retweets for Trump")
```



LASSO model for words on number of Retweets for Trump

```
plot(lasso_rt_biden,
     sub = "LASSO model for words on number of Retweets for Biden")
```



LASSO model for words on number of Retweets for Biden

```
### Top 5 words with highest effect on Favorites for both candidates
## Top 5 words with most positive effect on favs for Trump

lasso_favs_trump_coefs <- coef(lasso_favs_trump)[-1,]
kable(lasso_favs_trump_coefs[order(-lasso_favs_trump_coefs)][1:5],
      caption = "Trump's top 5 words with most positive effect on favorites")
```

Table 3: Trump's top 5 words with most positive effect on favorites

	x
perfect	217334.90
pretens	176502.21
bloodi	124132.30
unqualifi	112586.27
rich	92969.14

```
## Top 5 words with most negative effect on favs for Trump
kable(lasso_favs_trump_coefs[order(lasso_favs_trump_coefs)][1:5],
      caption = "Trump's top 5 words with most negative effect on favorites")
```

Table 4: Trump's top 5 words with most negative effect on favorites

	x
tire	-42915.933
plagu	-24786.564
cruelti	-17279.987
collus	-16779.342
infect	-7698.081

```
## Top 5 words with most positive effect on favs for Biden
lasso_favs_biden_coefs <- coef(lasso_favs_biden)[-1,]
kable(lasso_favs_biden_coefs[order(-lasso_favs_biden_coefs)][1:5],
      caption = "Biden's top 5 words with most positive effect on favorites")
```

Table 5: Biden's top 5 words with most positive effect on favorites

	x
incendiari	176302.57
incit	109371.07
inept	103447.46
complac	88937.46
furiou	88113.40

```
## Top 5 words with most negative effect on favs for Biden
kable(lasso_favs_biden_coefs[order(lasso_favs_biden_coefs)][1:5],
      caption = "Biden's top 5 words with most negative effect on favorites")
```

Table 6: Biden's top 5 words with most negative effect on favorites

	x
hysteria	-9840.775
aggress	-9302.851
dictat	-8024.756
reckless	-6936.710
cheaper	-6846.887

```
## Top 5 words with most positive effect on rt for Trump
lasso_rt_trump_coefs <- coef(lasso_rt_trump)[-1,]
kable(lasso_rt_trump_coefs[order(-lasso_rt_trump_coefs)][1:5],
      caption = "Trump's top 5 words with most positive effect on retweets")
```

Table 7: Trump's top 5 words with most positive effect on retweets

	x
bloodi	52838.47
intens	51698.46
wreck	37181.16
subvers	35451.49
hack	28029.48

```
## Top 5 words with most negative effect on rt for Trump
kable(lasso_rt_trump_coefs[order(lasso_rt_trump_coefs)][1:5],
      caption = "Trump's top 5 words with most negative effect on retweets")
```

Table 8: Trump's top 5 words with most negative effect on retweets

	x
relentless	-2978.602
racist	-2802.306
hatr	-2605.694
clearer	-1957.618
peac	-1537.048

```
## Top 5 words with most positive effect on rt for Biden
lasso_rt_biden_coefs <- coef(lasso_rt_biden)[-1,]
kable(lasso_rt_biden_coefs[order(-lasso_rt_biden_coefs)][1:5],
      caption = "Biden's top 5 words with most positive effect on retweets")
```

Table 9: Biden's top 5 words with most positive effect on retweets

	x
incendiari	22385.24
inept	20554.49
unprepar	18471.83
incit	16152.51
grim	14819.26

```
## Top 5 words with most negative effect on rt for Biden
kable(lasso_rt_biden_coefs[order(lasso_rt_biden_coefs)][1:5],
      caption = "Biden's top 5 words with most negative effect on retweets")
```

Table 10: Biden's top 5 words with most negative effect on retweets

	x
disregard	-2176.075
reckless	-2115.435
aggress	-2022.093
crush	-1546.197
errat	-1471.139

In the case of Trump's tweets, we can see that words such as 'obstacle', 'astoundishing', 'uncertain', or 'afraid' are the words with the most positive effects. The largest effect is the word 'bitter', because having that word in a tweet increased 2.173349×10^5 the number of favorites a tweet receives. Moreover, the 10 words that have most negative effects in Trump's tweets are 'collusion', 'fast', 'strong', 'hater', 'approval'. This words relate to the Russia investigation but also they could relate to his continuous tweets about his allegedly high approval rate.

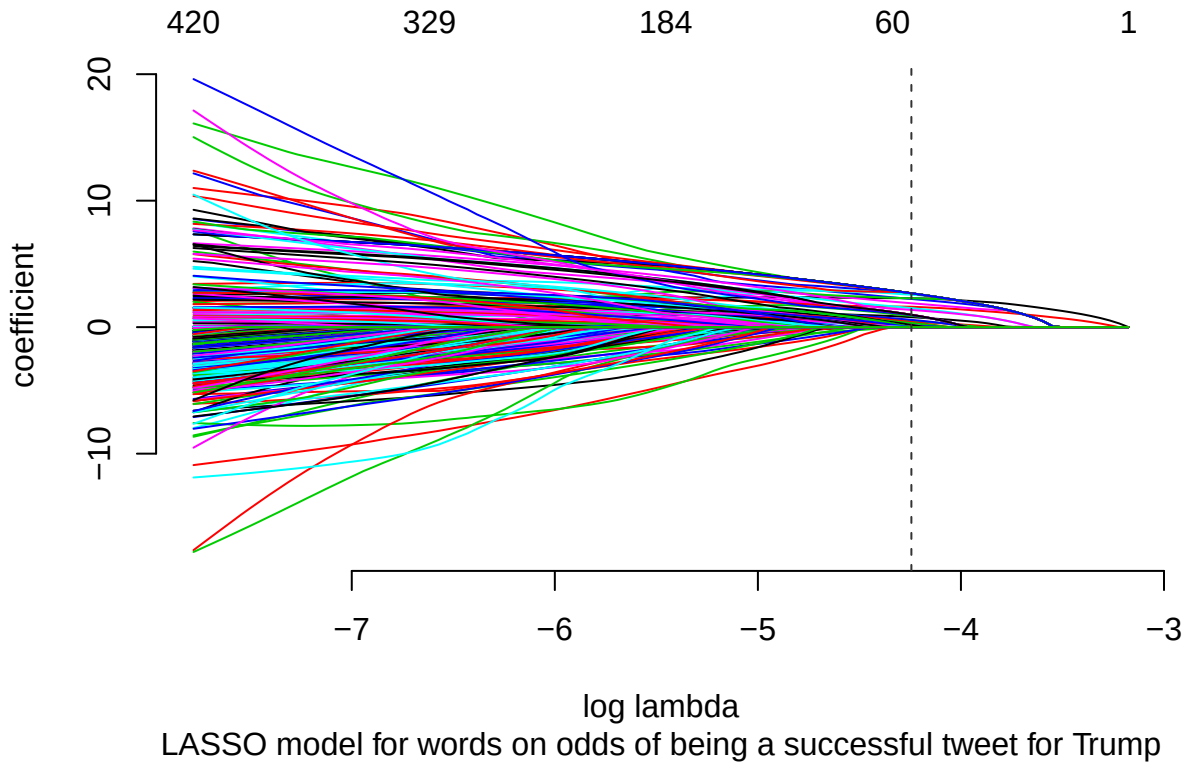
On the other hand, Biden's tweets have different words that have a positive effect on the number of favorites. For instance, words like 'incendiary', 'inept', 'unqualified', or 'incompetent' have strong positive effects. These words seem to be related to tweets talking about President Trump and his actions during the administration. Including one of this word in a tweet, is associated with an increase in the number of favorites by up to 1.7630257×10^5. Further, words such as 'dictator' or 'agressive' have the most negative effect on the number of favorites for Biden. Which could signla that, even though Biden's followers like when he criticize President Trump, they do not fall for aggressive comments against him.

The last four tables present the words with the most positive and negative effects on the number of retweets. As we can see for Trump's tweets, the words are similar than the ones we found above. However, for negative effects, we found other words that relate to the impeachment process such as 'subpoena', 'crisis', or 'crime'. This reflects that, even among Trump followers, his tweets about the impeachment process and the Russia Investigation, harmed his popularity.

Moreover, for Biden's tweets a similar pattern arises. Words that describe President Trump as inept or unqualified gave him a wide support among his followers, but aggresive or harsher words against him lowered the popularity of his tweets.

4.2 Words included in Trump's and Biden's "successful" tweets

```
### We defined a "successful" tweet if it is on the 90th percentile or more
success_fav_trump <- favs_trump$favorite_count>quantile(favs_trump$favorite_count,.9)
success_fav_biden <- favs_biden$favorite_count>quantile(favs_biden$favorite_count,.9)

## Analysis for Trump's successful tweets
lasso_favs_trump_success <- gamlr(data_matrix[favs_trump$X,],
                                  success_fav_trump,
                                  family = "binomial")
plot(lasso_favs_trump_success,
     sub = "LASSO model for words on odds of being a successful tweet for Trump")
```

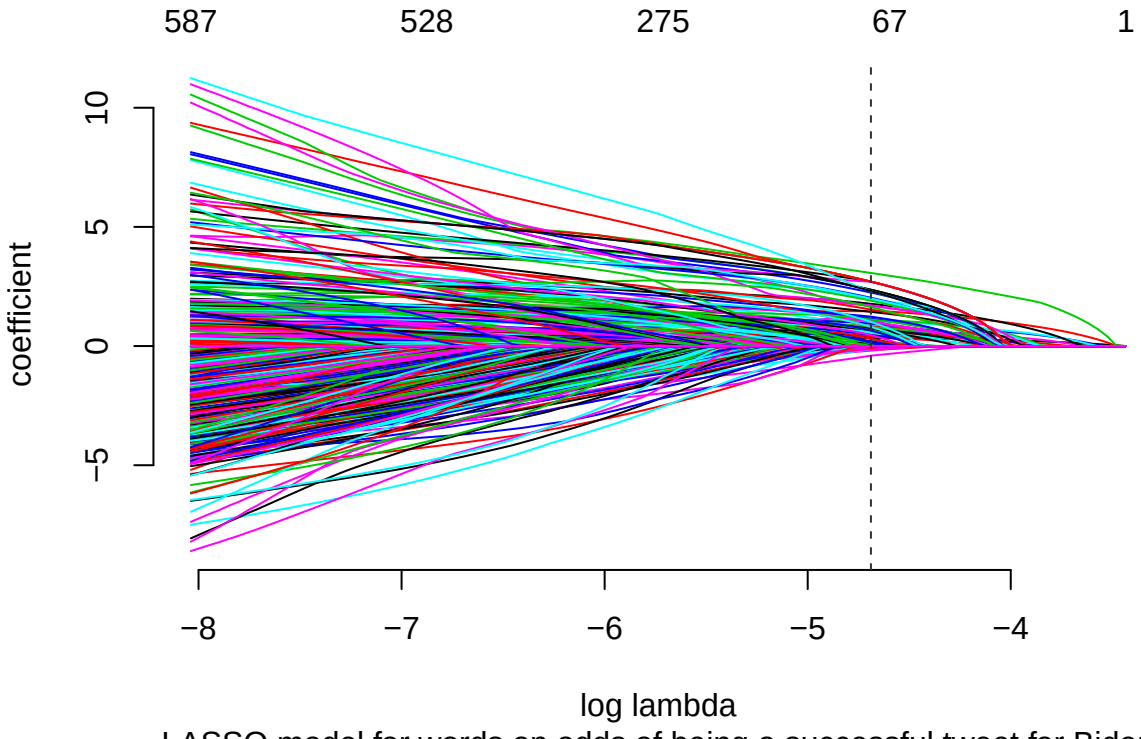



```
lasso_favs_trump_success_coefs <- coef(lasso_favs_trump_success)[-1,]
kable(lasso_favs_trump_success_coefs[order(-lasso_favs_trump_success_coefs)][1:10],
      caption = "Trump's top 10 words with highest effect on odds of being a successful tweet")
```

Table 11: Trump's top 10 words with highest effect on odds of being a successful tweet

	x
sincer	2.677979
emphat	2.677976
perfect	2.677974
horrif	2.677924
vain	2.677855
pretens	2.677844
blow	2.677839
risk	2.677838
faint	2.677837
bloodi	2.677825

```
## Analysis for Bidens's successful tweets
lasso_favs_biden_success <- gamlr(data_matrix[as.character(favs_biden$X),],
                                  success_fav_biden,
                                  family = "binomial")
plot(lasso_favs_biden_success,
     sub = "LASSO model for words on odds of being a successful tweet for Biden")
```




```
lasso_favs_biden_success_coefs <- coef(lasso_favs_biden_success)[-1,]
kable(lasso_favs_biden_success_coefs[order(-lasso_favs_biden_success_coefs)][1:10],
      caption = "Biden's top 10 words with highest effect on odds of being a successful tweet")
```

Table 12: Biden's top 10 words with highest effect on odds of being a successful tweet

	x
wound	3.067009
belittl	2.702877
complac	2.702877
extort	2.702867
fervid	2.702866
toughest	2.702863
vaniti	2.702863
incit	2.364283
law	2.364160
upset	2.354162

Additional to the analysis on the number of favorites and retweets, we performed an analysis on the words that increased the probability of being in a successful tweet. We defined 'successful' as a binary variable that indicates whether a tweet is on the 90th percentile with most favorites for each candidates. We defined this variable this way because of the large difference between the number of favorites each candidate receives.

We used a logistic LASSO with the 'successful' tweet variable as the outcome variable, and all the words we identified above as covariates. For Trump's tweets, the inclusion of words with the root 'horrif', such as 'horrific' or 'horrifying', increases the odds of a tweet to be on the 90th percentil of most successful tweets by 14.5556433 times, holding everything else constant.

Furthermore, for Biden's tweets words such as 'wound', 'belittle' or 'complacient' are the words that increases the odds for a tweet to be successful. For instance, the word 'wound' increases 21.4775699the odds of being a successful tweet ceteris paribus.

In conclusion, from the LASSO model with the lowest AICc we found that there are some specific words that increase or decrease the number of favorites and retweets for each candidate. Moreover, we found that some of those words also are part of the most successful tweets for each candidate. Finally, it is important to notice that even though we found important signals for individual words, the context in which those words were used also matters. Same words could have different meanings depending on the context and this is lost when we look at them sepparately from the whole tweet.

5 Network Analysis

```
# Organize data for network analysis
network <- select(data, user_id, screen_name, mentions_screen_name)
network <- network %>%
  filter(is.na(mentions_screen_name) == FALSE)
network_final <- network %>%
  mutate(has_multiple_names = ifelse(str_detect(mentions_screen_name, "[(]"), 1, 0))
multiple_names <- network_final %>%
  filter(has_multiple_names == 1)
multiple_names <- multiple_names %>%
  mutate(number_of_words = str_count(mentions_screen_name, "[,]") + 1)
max_num_words = max(multiple_names$number_of_words)
multiple_names <- multiple_names %>%
  separate(
    mentions_screen_name,
    c('word1', 'word2', 'word3', 'word4', 'word5', 'word6', 'word7', 'word8', 'word9', 'word10', 'word1
    sep = "[^[:alnum:]]+",
    remove = TRUE,
    convert = FALSE,
    extra = "warn",
    fill = "warn"
  )
multiple_names <- multiple_names %>%
  pivot_longer(starts_with('word'), names_to = 'word_num')
multiple_names <- multiple_names %>%
  filter(value != 'c') %>%
  rename(mentions_screen_name = value) %>%
  select(user_id, screen_name, mentions_screen_name) %>%
  filter(!is.na(mentions_screen_name))
initial_network <- network_final %>%
  filter(has_multiple_names == 0) %>%
  select(-has_multiple_names)

network_final <- rbind(initial_network, multiple_names)
```

For the network analysis, we considered the accounts Trump and Biden mentioned in their tweets. We did not analyse the accounts that were mentioned without the handle ("@"), just the ones that were directly mentioned. The network analysis plots illustrate the mentions from Trump and Biden's accounts. Trump is displayed in a light red circle and Biden in a light blue circle. All mentioned accounts are shown in a light green circle each. The red lines show who Trump mentioned while the blue lines show who Biden mentioned. The shorter the line, and therefore closer the circle, the more mentiones are done by the candidate. We analyzed only selected accounts mentioned and divided them into three groups: (1) News: major news accounts trying to include different sources and perspectives. (2) Other candidates: other candidates int the 2020 presidential election. We only included candidates who were mentioned and were still running by November 1, 2019. (3) Accounts of interest: some accounts who have significant relevance. President Obama; Hillary Clinton, ex Secretary of State and former candidate against President Trump in 2016; current vice president Pence; the two highest secretaries in the order of precedence, Pompeo and Mnuchin, State and Treasury respectively; both of the candidate's parties; and Speaker of the House and Senate Majority Leader who interestingly are from the two candidate's parties, the Speaker, Pelosi is Democrat, and the Senate Majority Leader, McConnell, is Republican.

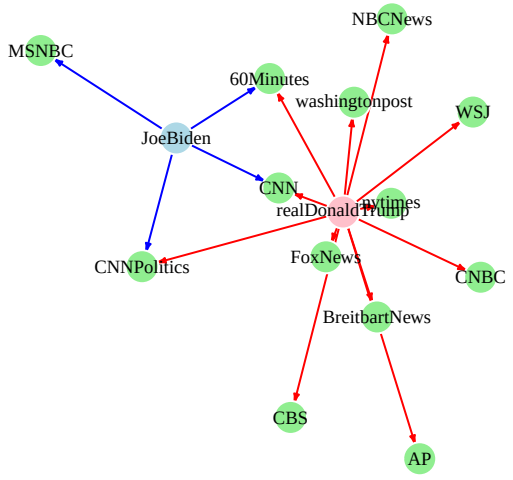
5.1 News

The network analysis of news accounts shows Biden is less active than Trump when it comes to mentioning news accounts. Trump is almost equally likely to mention news sources that critize him like NY Times as those that favor him like Fox News. Meanwhile, Biden mentions few accounts which are in general terms favorable to him. Both candidates mention CNN but Trump does it at a larger scale.

```
# List of News Accounts
newslist <- c("60Minutes", "AP", "BreitbartNews", "CBS", "CNBC", "CNN", "CNNPolitics", "FoxNews", "MSNB
network_news <- filter(network_final, mentions_screen_name %in% newslist)
```

```
# Create Edges and Nodes for News
candidate_news <- network_news %>%
  distinct(screen_name) %>%
  rename(label = screen_name)

mention_news <- network_news %>%
  distinct(mentions_screen_name) %>%
  rename(label = mentions_screen_name)

nodes_news <- full_join(candidate_news, mention_news, by = "label")

nodes_news <- nodes_news %>% rowid_to_column("id")

route_news <- network_news %>%
  group_by(screen_name, mentions_screen_name) %>%
  summarise(weight = n()) %>%
  ungroup()

edges_news <- route_news %>%
  left_join(nodes_news, by = c("screen_name" = "label")) %>%
  rename(from = id)

edges_news <- edges_news %>%
  left_join(nodes_news, by = c("mentions_screen_name" = "label")) %>%
  rename(to = id)

edges_news <- select(edges_news, from, to, weight)

news_igraph <- graph_from_data_frame(d = edges_news, vertices = nodes_news, directed = TRUE)
```

```
# News Network Plot
edges_Trump_news <- incident(news_igraph, V(news_igraph)[label=="realDonaldTrump"], mode="all")
ecol_news <- rep("blue", ecount(news_igraph))
ecol_news[edges_Trump_news] <- "red"

vcol_news <- rep("lightgreen", vcount(news_igraph))
vcol_news[V(news_igraph)$label=="JoeBiden"] <- "lightblue"
vcol_news[V(news_igraph)$label=="realDonaldTrump"] <- "pink"

plot(news_igraph, vertex.color=vcol_news, vertex.frame.color="white", vertex.label.color="black", edge.
```



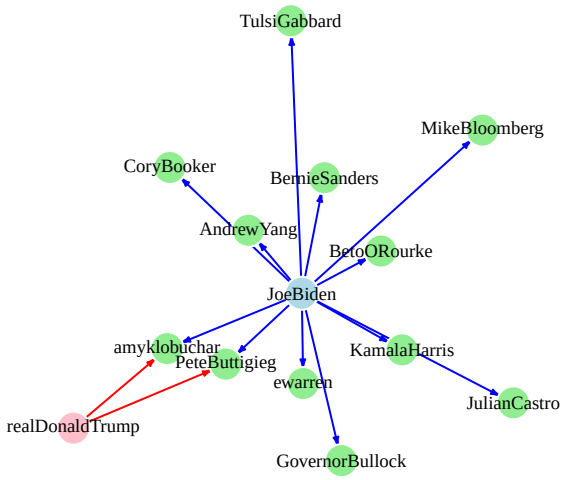
5.2 Other Candidates

All the candidates that meet the criteria to be classified in this list (being mentioned at least once and still running by November 1, 2019) are democrats are are also mentioned by Biden. Trump only mentions Buttigieg and Klobuchar who is also mentioned more by Trump than by Biden.

```
# List of Other Candidates Accounts
candidateslist <- c("amyklobuchar", "AndrewYang", "BernieSanders", "BetoORourke", "CoryBooker", "ewarren
network_candidates <- filter(network_final, mentions_screen_name %in% candidateslist)
```

```
# Create Edges and Nodes for Other Candidates
candidate_candidates <- network_candidates %>%
  distinct(screen_name) %>%
  rename(label = screen_name)

mention_candidates <- network_candidates %>%
  distinct(mentions_screen_name) %>%
  rename(label = mentions_screen_name)

nodes_candidates <- full_join(candidate_candidates, mention_candidates, by = "label")

nodes_candidates <- nodes_candidates %>% rowid_to_column("id")

route_candidates <- network_candidates %>%
  group_by(screen_name, mentions_screen_name) %>%
  summarise(weight = n()) %>%
  ungroup()

edges_candidates <- route_candidates %>%
  left_join(nodes_candidates, by = c("screen_name" = "label")) %>%
  rename(from = id)

edges_candidates <- edges_candidates %>%
  left_join(nodes_candidates, by = c("mentions_screen_name" = "label")) %>%
  rename(to = id)

edges_candidates <- select(edges_candidates, from, to, weight)

candidates_igraph <- graph_from_data_frame(d = edges_candidates, vertices = nodes_candidates, directed =
```

```
# Other Candidates Network Plot
edges_Trump_candidates <- incident(candidates_igraph, V(candidates_igraph)[label=="realDonaldTrump"], mo
ecol_candidates <- rep("blue", ecount(candidates_igraph))
ecol_candidates[edges_Trump_candidates] <- "red"

vcol_candidates <- rep("lightgreen", vcount(candidates_igraph))
vcol_candidates[V(candidates_igraph)$label=="JoeBiden"] <- "lightblue"
vcol_candidates[V(candidates_igraph)$label=="realDonaldTrump"] <- "pink"

plot(candidates_igraph, vertex.color=vcol_candidates, vertex.frame.color="white", vertex.label.color="bl
```



5.3 Accounts of Interest

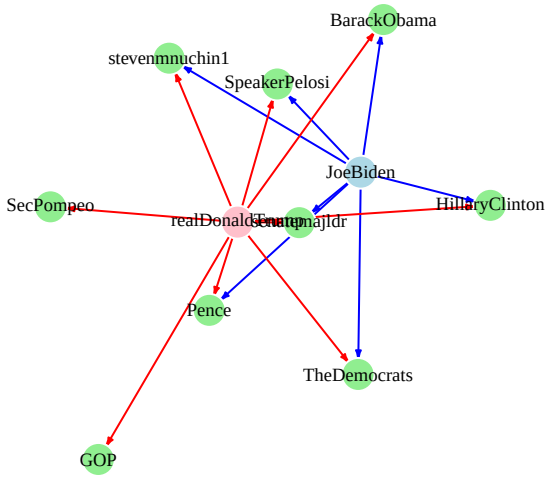
The las group covers some accounts that have significant impact during the election process. This explains why Trump mentions all of them and Biden all except the Republican Party and the Secretary of State. The Senate Majority Leader, a Republican, is the most mentioned by both candidates. While Biden doesn't mention the Republican Party, Trump mentions the Democratic Party as much as Biden, while Obama and Clinton are mentioned slighly more by Biden than by Trump.

```
# List of Accounts of Interest
interestlist <- c("BarackObama", "GOP", "HillaryClinton", "Pence", "SecPompeo", "senatemajldr", "Speaker
network_interest <- filter(network_final, mentions_screen_name %in% interestlist)
```

```
# Create Edges and Nodes for Accounts of Interst
candidate_interest <- network_interest %>%
  distinct(screen_name) %>%
  rename(label = screen_name)

mention_interest <- network_interest %>%
  distinct(mentions_screen_name) %>%
  rename(label = mentions_screen_name)

nodes_interest <- full_join(candidate_interest, mention_interest, by = "label")

nodes_interest <- nodes_interest %>% rowid_to_column("id")

route_interest <- network_interest %>%
  group_by(screen_name, mentions_screen_name) %>%
  summarise(weight = n()) %>%
```

```
  ungroup()

edges_interest <- route_interest %>%
  left_join(nodes_interest, by = c("screen_name" = "label")) %>%
  rename(from = id)

edges_interest <- edges_interest %>%
  left_join(nodes_interest, by = c("mentions_screen_name" = "label")) %>%
  rename(to = id)

edges_interest <- select(edges_interest, from, to, weight)

interest_igraph <- graph_from_data_frame(d = edges_interest, vertices = nodes_interest, directed = TRUE)
```

```
# Accounts of Interest Network Plot
edges_Trump_interest <- incident(interest_igraph, V(interest_igraph)[label=="realDonaldTrump"], mode="al
ecol_interest <- rep("blue", ecount(interest_igraph))
ecol_interest[edges_Trump_interest] <- "red"

vcol_interest <- rep("lightgreen", vcount(interest_igraph))
vcol_interest[V(interest_igraph)$label=="JoeBiden"] <- "lightblue"
vcol_interest[V(interest_igraph)$label=="realDonaldTrump"] <- "pink"

plot(interest_igraph, vertex.color=vcol_interest, vertex.frame.color="white", vertex.label.color="black"
```



6 Session info

```
devtools::session_info()
```

```
## - Session info -------------------------------------------------------------
##  setting  value
##  version  R version 3.6.3 (2020-02-29)
##  os       macOS Catalina 10.15.5
##  system   x86_64, darwin15.6.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       America/Bogota
##  date     2020-06-12
##
## - Packages -----------------------------------------------------------------
##  package      * version date       lib source
##  assertthat     0.2.1   2019-03-21 [1] CRAN (R 3.6.0)
##  backports      1.1.6   2020-04-05 [1] CRAN (R 3.6.2)
##  broom          0.5.5   2020-02-29 [1] CRAN (R 3.6.0)
##  callr          3.4.3   2020-03-28 [1] CRAN (R 3.6.2)
##  cellranger     1.1.0   2016-07-27 [1] CRAN (R 3.6.0)
##  cli            2.0.2   2020-02-28 [1] CRAN (R 3.6.0)
##  codetools      0.2-16  2018-12-24 [1] CRAN (R 3.6.3)
##  colorspace     1.4-1   2019-03-18 [1] CRAN (R 3.6.0)
##  crayon         1.3.4   2017-09-16 [1] CRAN (R 3.6.0)
##  DBI            1.1.0   2019-12-15 [1] CRAN (R 3.6.0)
##  dbplyr         1.4.2   2019-06-17 [1] CRAN (R 3.6.0)
##  desc           1.2.0   2018-05-01 [1] CRAN (R 3.6.0)
##  devtools       2.3.0   2020-04-10 [1] CRAN (R 3.6.2)
##  digest         0.6.25  2020-02-23 [1] CRAN (R 3.6.0)
##  dplyr        * 0.8.5   2020-03-07 [1] CRAN (R 3.6.0)
##  ellipsis       0.3.0   2019-09-20 [1] CRAN (R 3.6.0)
##  evaluate       0.14    2019-05-28 [1] CRAN (R 3.6.0)
##  fansi          0.4.1   2020-01-08 [1] CRAN (R 3.6.0)
##  farver         2.0.3   2020-01-16 [1] CRAN (R 3.6.0)
##  forcats      * 0.5.0   2020-03-01 [1] CRAN (R 3.6.0)
##  fs             1.4.1   2020-04-04 [1] CRAN (R 3.6.2)
##  gamlr        * 1.13-5  2018-05-13 [1] CRAN (R 3.6.0)
##  generics       0.0.2   2018-11-29 [1] CRAN (R 3.6.0)
##  ggplot2      * 3.3.0   2020-03-05 [1] CRAN (R 3.6.0)
##  glue           1.4.0   2020-04-03 [1] CRAN (R 3.6.2)
##  gtable         0.3.0   2019-03-25 [1] CRAN (R 3.6.0)
##  haven          2.2.0   2019-11-08 [1] CRAN (R 3.6.0)
##  highr          0.8     2019-03-20 [1] CRAN (R 3.6.0)
##  hms            0.5.3   2020-01-08 [1] CRAN (R 3.6.0)
##  htmltools      0.4.0   2019-10-04 [1] CRAN (R 3.6.0)
##  httr           1.4.1   2019-08-05 [1] CRAN (R 3.6.0)
##  igraph       * 1.2.5   2020-03-19 [1] CRAN (R 3.6.0)
##  janeaustenr    0.1.5   2017-06-10 [1] CRAN (R 3.6.0)
##  jsonlite       1.6.1   2020-02-02 [1] CRAN (R 3.6.0)
##  knitr        * 1.28    2020-02-06 [1] CRAN (R 3.6.0)
##  labeling       0.3     2014-08-23 [1] CRAN (R 3.6.0)
```

```
##  lattice        0.20-41 2020-04-02 [1] CRAN (R 3.6.2)
##  lifecycle      0.2.0   2020-03-06 [1] CRAN (R 3.6.0)
##  lubridate      1.7.8   2020-04-06 [1] CRAN (R 3.6.2)
##  magrittr       1.5     2014-11-22 [1] CRAN (R 3.6.0)
##  maptpx       * 1.9-7   2020-05-28 [1] CRAN (R 3.6.2)
##  Matrix       * 1.2-18  2019-11-27 [1] CRAN (R 3.6.0)
##  memoise        1.1.0   2017-04-21 [1] CRAN (R 3.6.0)
##  modelr         0.1.6   2020-02-22 [1] CRAN (R 3.6.0)
##  modeltools     0.2-23  2020-03-05 [1] CRAN (R 3.6.0)
##  munsell        0.5.0   2018-06-12 [1] CRAN (R 3.6.0)
##  nlme           3.1-147 2020-04-13 [1] CRAN (R 3.6.2)
##  NLP          * 0.2-0   2018-10-18 [1] CRAN (R 3.6.0)
##  pillar         1.4.3   2019-12-20 [1] CRAN (R 3.6.0)
##  pkgbuild       1.0.6   2019-10-09 [1] CRAN (R 3.6.0)
##  pkgconfig      2.0.3   2019-09-22 [1] CRAN (R 3.6.0)
##  pkgload        1.0.2   2018-10-29 [1] CRAN (R 3.6.0)
##  prettyunits    1.1.1   2020-01-24 [1] CRAN (R 3.6.0)
##  processx       3.4.2   2020-02-09 [1] CRAN (R 3.6.0)
##  ps             1.3.2   2020-02-13 [1] CRAN (R 3.6.0)
##  purrr        * 0.3.3   2019-10-18 [1] CRAN (R 3.6.0)
##  R6             2.4.1   2019-11-12 [1] CRAN (R 3.6.0)
##  RColorBrewer * 1.1-2   2014-12-07 [1] CRAN (R 3.6.0)
##  Rcpp           1.0.4   2020-03-17 [1] CRAN (R 3.6.0)
##  readr        * 1.3.1   2018-12-21 [1] CRAN (R 3.6.0)
##  readxl       * 1.3.1   2019-03-13 [1] CRAN (R 3.6.0)
##  remotes        2.1.1   2020-02-15 [1] CRAN (R 3.6.0)
##  reprex         0.3.0   2019-05-16 [1] CRAN (R 3.6.0)
##  rlang          0.4.5   2020-03-01 [1] CRAN (R 3.6.0)
##  rmarkdown      2.1     2020-01-20 [1] CRAN (R 3.6.0)
##  rprojroot      1.3-2   2018-01-03 [1] CRAN (R 3.6.0)
##  rstudioapi     0.11    2020-02-07 [1] CRAN (R 3.6.0)
##  rtweet       * 0.7.0   2020-01-08 [1] CRAN (R 3.6.0)
##  rvest          0.3.5   2019-11-08 [1] CRAN (R 3.6.0)
##  scales         1.1.0   2019-11-18 [1] CRAN (R 3.6.0)
##  sessioninfo    1.1.1   2018-11-05 [1] CRAN (R 3.6.0)
##  slam         * 0.1-47  2019-12-21 [1] CRAN (R 3.6.0)
##  SnowballC    * 0.7.0   2020-04-01 [1] CRAN (R 3.6.2)
##  stringi        1.4.6   2020-02-17 [1] CRAN (R 3.6.0)
##  stringr      * 1.4.0   2019-02-10 [1] CRAN (R 3.6.0)
##  syuzhet      * 1.0.4   2017-12-14 [1] CRAN (R 3.6.0)
##  testthat       2.3.2   2020-03-02 [1] CRAN (R 3.6.0)
##  tibble       * 3.0.0   2020-03-30 [1] CRAN (R 3.6.2)
##  tidyr        * 1.0.2   2020-01-24 [1] CRAN (R 3.6.0)
##  tidyselect     1.0.0   2020-01-27 [1] CRAN (R 3.6.0)
##  tidytext     * 0.2.4   2020-04-17 [1] CRAN (R 3.6.2)
##  tidyverse    * 1.3.0   2019-11-21 [1] CRAN (R 3.6.0)
##  tm           * 0.7-7   2019-12-12 [1] CRAN (R 3.6.0)
##  tokenizers     0.2.1   2018-03-29 [1] CRAN (R 3.6.0)
##  topicmodels  * 0.2-11  2020-04-19 [1] CRAN (R 3.6.2)
##  usethis        1.6.1   2020-04-29 [1] CRAN (R 3.6.2)
##  vctrs          0.2.4   2020-03-10 [1] CRAN (R 3.6.0)
##  withr          2.1.2   2018-03-15 [1] CRAN (R 3.6.0)
##  wordcloud    * 2.6     2018-08-24 [1] CRAN (R 3.6.0)
##  xfun           0.13    2020-04-13 [1] CRAN (R 3.6.2)
```

```
##  xml2           1.3.1   2020-04-09 [1] CRAN (R 3.6.2)
##  yaml           2.2.1   2020-02-01 [1] CRAN (R 3.6.0)
##
## [1] /Library/Frameworks/R.framework/Versions/3.6/Resources/library
```