Analysis of Tweets

Santosh Saranyan

12/2/2021

Loading in the dataset

```
dir<-"twitter"
path<-file.path(dir,"realDonaldTrump-20201106.csv")
#Preserving id by reading it in as a character
df<-read_csv(path, col_types=cols(id=col_character()))
#Getting the year from the date column
df$Year<-format(df$date,format="%Y")
head(df,10)
```

```
## # A tibble: 10 × 9
##    id                  text  isRetweet isDeleted device favorites retweets
date
##    <chr>               <chr> <lgl>     <lgl>     <chr>      <dbl>    <dbl>
<dttm>
##  1 98454970654916608   Repu… FALSE     FALSE     Tweet…        49      255
2011-08-02 18:07:48
##  2 1234653427789070336 I wa… FALSE     FALSE     Twitt…     73748    17404
2020-03-03 01:34:50
##  3 1218010753434820614 RT @… TRUE      FALSE     Twitt…         0     7396
2020-01-17 03:22:47
##  4 1304875170860015617 The … FALSE     FALSE     Twitt…     80527    23502
2020-09-12 20:10:58
##  5 1218159531554897920 RT @… TRUE      FALSE     Twitt…         0     9081
2020-01-17 13:13:59
##  6 1217962723234983937 RT @… TRUE      FALSE     Twitt…         0    25048
2020-01-17 00:11:56
##  7 1315779944002199552 “I’m… FALSE     FALSE     Twitt…    149007    34897
2020-10-12 22:22:39
##  8 1223640662689689602 Gett… FALSE     FALSE     Twitt…    285863    30209
2020-02-01 16:14:02
##  9 1319501865625784320 http… FALSE     FALSE     Twitt…    130822    19127
2020-10-23 04:52:14
## 10 1319500520126664705 http… FALSE     FALSE     Twitt…    153446    20275
2020-10-23 04:46:53
## # … with 1 more variable: Year <chr>
```

Removing retweets, tweets without spaces and replacing @ with quotes to just @ to remove usernames later

```
df<-df[which(df$isRetweet=="FALSE"),]
df<-df[grepl(" ", df$text),]
```

```r
df$text<-str_replace(df$text, '"""@', "@")
head(df,10)
```

```
## # A tibble: 10 × 9
##    id                  text  isRetweet isDeleted device favorites retweets
date
##    <chr>               <chr> <lgl>     <lgl>     <chr>      <dbl>    <dbl>
<dttm>
##  1 98454970654916608   Repu… FALSE     FALSE     Tweet…        49      255
2011-08-02 18:07:48
##  2 1234653427789070336 I wa… FALSE     FALSE     Twitt…     73748    17404
2020-03-03 01:34:50
##  3 1304875170860015617 The … FALSE     FALSE     Twitt…     80527    23502
2020-09-12 20:10:58
##  4 1315779944002199552 “I’m… FALSE     FALSE     Twitt…    149007    34897
2020-10-12 22:22:39
##  5 1223640662689689602 Gett… FALSE     FALSE     Twitt…    285863    30209
2020-02-01 16:14:02
##  6 1215247978966986752 Than… FALSE     FALSE     Twitt…     48510    11608
2020-01-09 12:24:31
##  7 1319491234042269696 As p… FALSE     FALSE     Twitt…    253761    79855
2020-10-23 04:09:59
##  8 1319683876046934016 HUGE… FALSE     FALSE     Twitt…    215994    51830
2020-10-23 16:55:29
##  9 1319655865083940865 Than… FALSE     FALSE     Twitt…    178163    24864
2020-10-23 15:04:10
## 10 1319510534098735106 11 D… FALSE     FALSE     Twitt…    197604    49800
2020-10-23 05:26:41
## # … with 1 more variable: Year <chr>
```

Tokenizing the tweets with token=“tweets”

```r
df_tidy<-unnest_tokens(df, output="word", input=text, token="tweets")
```

```
## Using `to_lower = TRUE` with `token = 'tweets'` may not preserve URLs.
```

```r
df_tidy
```

```
## # A tibble: 900,183 × 9
##    id    isRetweet isDeleted device favorites retweets date
Year
##    <chr> <lgl>     <lgl>     <chr>      <dbl>    <dbl> <dttm>
<chr>
##  1 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  2 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  3 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  4 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
```

```
##   5 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   6 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   7 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   8 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   9 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
## 10 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
## # … with 900,173 more rows, and 1 more variable: word <chr>
```

Removing urls and usernames

```
df_tidy2<-df_tidy[!grepl("http", df_tidy$word),]
df_tidy2<-df_tidy2[!grepl("@", df_tidy2$word),]
df_tidy2

## # A tibble: 848,524 × 9
##    id    isRetweet isDeleted device favorites retweets date
Year
##    <chr> <lgl>     <lgl>     <chr>      <dbl>    <dbl> <dttm>
<chr>
##   1 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   2 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   3 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   4 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   5 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   6 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   7 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   8 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##   9 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
## 10 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
## # … with 848,514 more rows, and 1 more variable: word <chr>
```

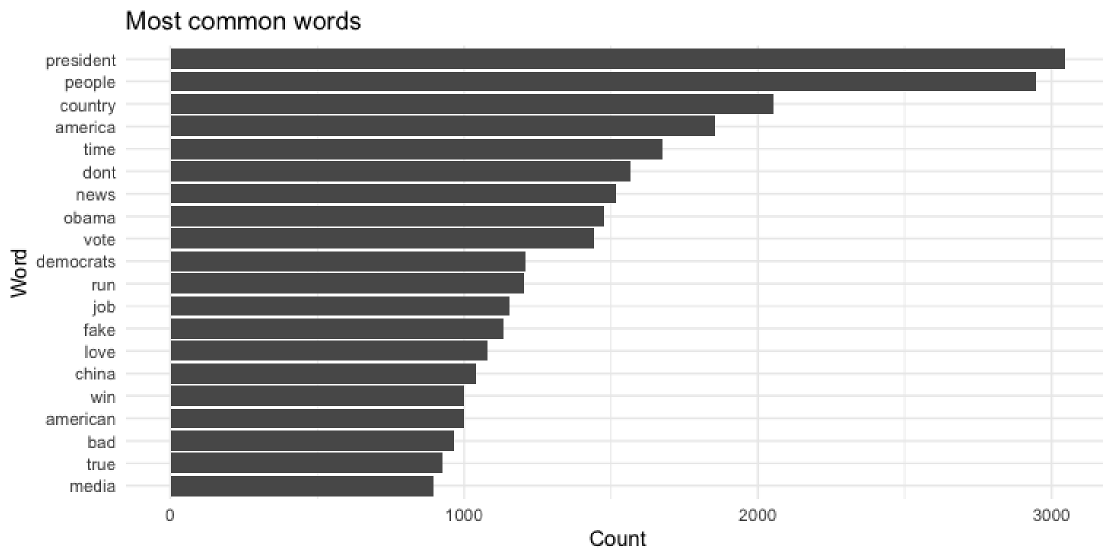
Removing &amp, stop words and variations of donald trump.

```
df_tidy3<-anti_join(df_tidy2, stop_words, by="word")
df_tidy3<-df_tidy3[!grepl("amp", df_tidy3$word),]
```

```
df_tidy3<-df_tidy3[!grepl("&amp", df_tidy3$word),]
df_tidy3<-df_tidy3[!grepl("donald", df_tidy3$word),]
df_tidy3<-df_tidy3[!grepl("trump", df_tidy3$word),]
head(df_tidy3,10)

## # A tibble: 10 × 9
##    id    isRetweet isDeleted device favorites retweets date
Year
##    <chr> <lgl>     <lgl>     <chr>      <dbl>    <dbl> <dttm>
<chr>
##  1 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  2 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  3 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  4 9845… FALSE     FALSE     Tweet…        49      255 2011-08-02 18:07:48
2011
##  5 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  6 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  7 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  8 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  9 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
## 10 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
## # … with 1 more variable: word <chr>
```

Visualizing the top 20 common words in all the tweets

```
df_tidy3 %>%
  count(word, sort=TRUE) %>%
  top_n(20) %>%
  ggplot(aes(x=reorder(word, n), y=n)) +
  geom_col() +
  coord_flip() +
  labs(x="Word", y="Count",
       title="Most common words") +
  theme_minimal()

## Selecting by n
```



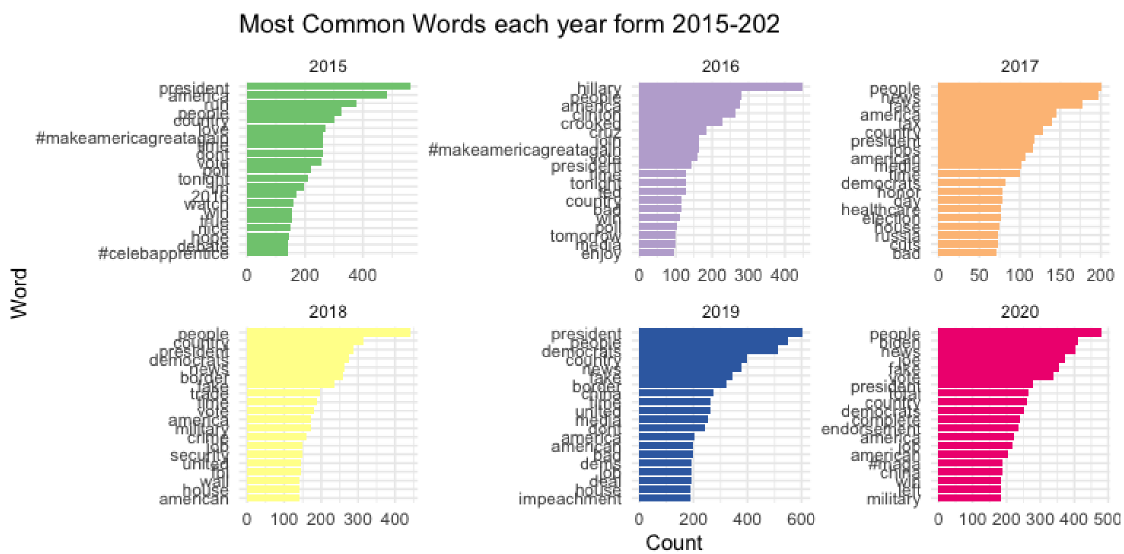
President seems to be the most common word followed by people, with country being the third most used.

Getting tweets sent between 2015 and 2020

```
df_tidy4<-filter(df_tidy3, date>="2015-01-01" & date<="2020-12-31")
head(df_tidy4,10)

## # A tibble: 10 × 9
##    id    isRetweet isDeleted device favorites retweets date
Year
##    <chr> <lgl>     <lgl>     <chr>      <dbl>    <dbl> <dttm>
<chr>
##  1 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  2 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  3 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  4 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  5 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  6 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  7 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  8 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
##  9 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
## 10 1234… FALSE     FALSE     Twitt…     73748    17404 2020-03-03 01:34:50
2020
## # … with 1 more variable: word <chr>
```

Grouping and faceting by year and visualizing the most common words for each year from 2015-2020

```
df_tidy4 %>%
  count(word, Year, sort=TRUE) %>%
  group_by(Year) %>%
  top_n(20) %>%
  ggplot(aes(x=reorder_within(word, n, Year), y=n, fill=Year)) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~Year, scales="free") +
  coord_flip() +
  labs(x="Word", y="Count",
       title="Most Common Words each year form 2015-202",
       fill="Year") +
  scale_fill_brewer(palette="Accent") +
  scale_x_reordered() +
  theme_minimal()

## Selecting by n
```



People seems to be one of the most common words across all years with it being the most used in 2017,2018 and 2020 and being the second most used in 2016 and 2019 and the fourth most used in 2015. President is the most used in 2015 and 2019, with 2016 having a unique top word of hillary. America is one of the top few words in 2015,2016 and 2017, but falls lower on the list in the later years, with democrats being more used in 2018 and 2019.

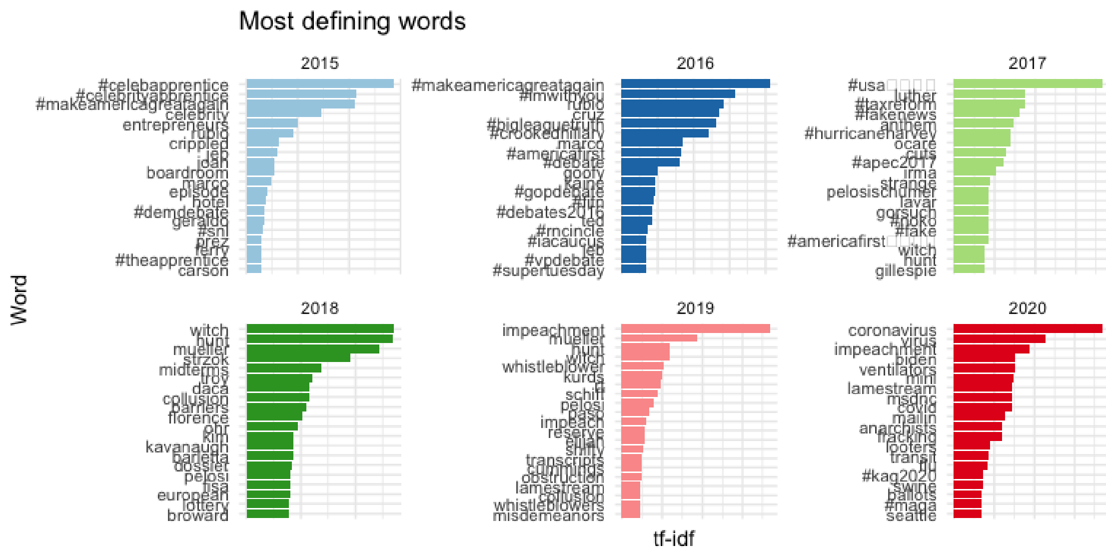
Calculating the tf-idf with year as the document.

```
df_tf_idf <- df_tidy4 %>%
  count(Year, word, sort=TRUE) %>%
  bind_tf_idf(term=word, document=Year, n=n)

arrange(df_tf_idf, desc(tf_idf))
```

```
## # A tibble: 43,648 × 6
##    Year  word                          n      tf   idf  tf_idf
##    <chr> <chr>                     <int>   <dbl> <dbl>   <dbl>
##  1 2015  #celebapprentice            138 0.00320 1.79  0.00574
##  2 2015  #celebrityapprentice        102 0.00237 1.79  0.00424
##  3 2015  #makeamericagreatagain      262 0.00608 0.693 0.00422
##  4 2016  #makeamericagreatagain      163 0.00594 0.693 0.00412
##  5 2019  impeachment                 188 0.00335 1.10  0.00368
##  6 2020  coronavirus                 103 0.00205 1.79  0.00368
##  7 2016  #imwithyou                   48 0.00175 1.79  0.00313
##  8 2015  celebrity                    70 0.00163 1.79  0.00291
##  9 2016  rubio                        71 0.00259 1.10  0.00284
## 10 2016  cruz                        184 0.00670 0.405 0.00272
## # … with 43,638 more rows
```

Plotting the document defining words for each year.

```r
library(stringr)

df_tf_idf %>%
  filter(str_detect(word, "[:alpha:]")) %>%
  group_by(Year) %>%
  top_n(20, wt=tf_idf) %>%
  ggplot(aes(x=reorder_within(word, tf_idf, Year),
             y=tf_idf,fill=factor(Year))) +
  geom_col(position="dodge",show.legend=FALSE) +
  coord_flip() +
  facet_wrap(~Year, scales="free") +
  labs(x="Word", y="tf-idf",
       title="Most defining words",
       fill="Year") +
  scale_fill_brewer(palette="Paired") +
  scale_x_reordered() +
  scale_y_continuous(labels=NULL) +
  theme_minimal()
```



Each year has a different set of document defining words, with 2015 having celebrity appearance as the top, with make america great again being the top in 2016 and one of the top few ones in 2015. 2018 and 2019 seems to have witch hunt as one of the tops while 2020 has the coronavirus as the most document defining word. Each year has almost a unique set of document defining words.

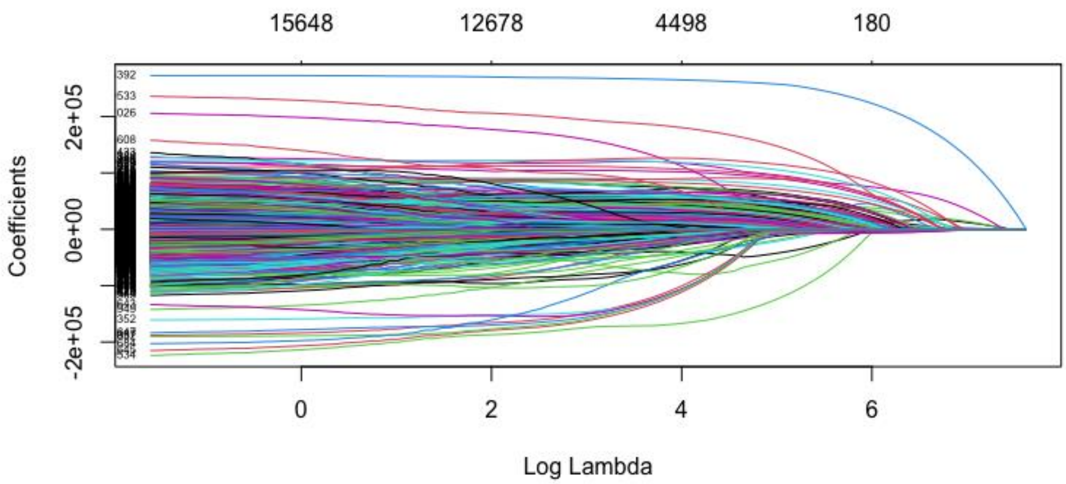
Creating the sparse matrix for tweets between 2016 and 2020.

```
#Creating the sparse matrix
df_tidy5<-filter(df_tidy3, date>="2016-01-01" & date<="2020-12-31")
df_dtm <- df_tidy5 %>%
  count(id, word) %>%
  cast_sparse(row=id, column=word, value=n)
#Getting the ids to join later to get the retweets column
df_dtm_ids<-tibble(id=rownames(df_dtm))
df_joined<-left_join(df_dtm_ids,df)

## Joining, by = "id"

#Matrix::print(df_dtm, col.names=TRUE)
```

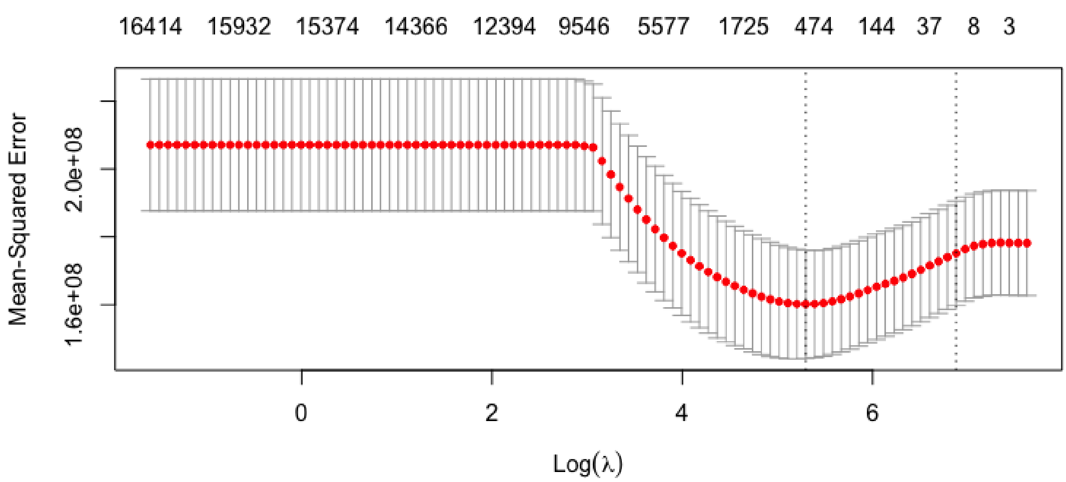
Fitting the model with cross-validation

```
library(glmnet)

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-3
```

```
set.seed(2)
x<-df_dtm
y<-df_joined$retweets
fit1 <- glmnet(x, y)
plot(fit1, xvar="lambda", label=TRUE)
```



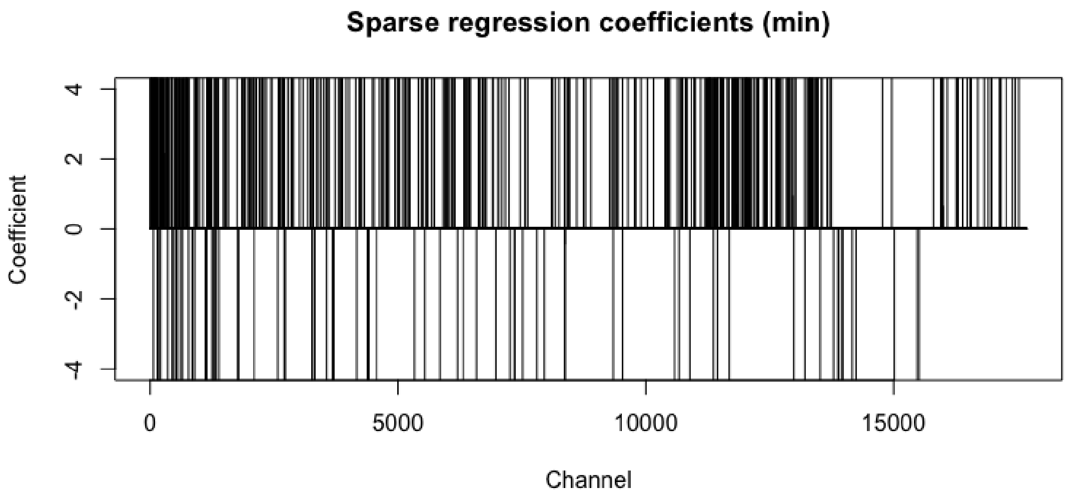

```
cvfit <- cv.glmnet(x, y)
plot(cvfit)
```




```
cvfit$lambda.1se

## [1] 970.4637

cvfit$lambda.min

## [1] 199.5771
```

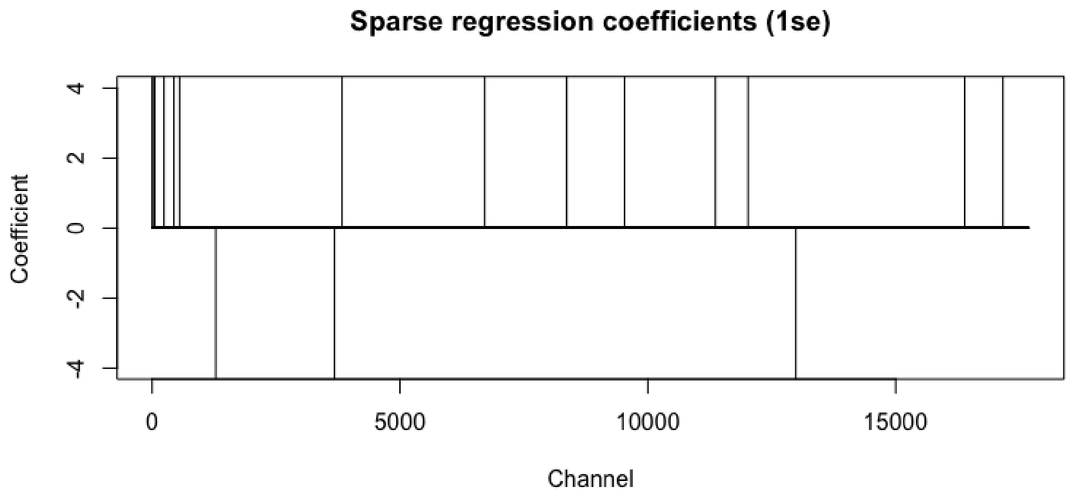
Calculating lambda for best model

```
c1 <- coef(cvfit, s="lambda.min")


sum(c1 != 0)

## [1] 568

plot(c1, type='h', ylim=c(-4, 4),
     xlab="Channel", ylab="Coefficient",
     main="Sparse regression coefficients (min)")
```



Calculating 1 standard error lambda

```
c2 <- coef(cvfit, s="lambda.1se")

sum(c2 != 0)

## [1] 18

plot(c2, type='h', ylim=c(-4, 4),
     xlab="Channel", ylab="Coefficient",
     main="Sparse regression coefficients (1se)")
```



Taking the lambda that is within 1 standard error (most sparse model), we get 884.25 with there being 23 non-zero coefficients

Displaying the most words with the strongest relationship with retweets

```
sparse_coeffs<-as.data.frame(as.matrix(c2))
head(arrange(sparse_coeffs, desc(sparse_coeffs)),10)

##                      s1
## #fnn        145621.612
## quarantine   44010.012
## rocky        17082.118
## (Intercept)  15872.324
## a$ap         14661.804
## insult        8592.303
## starved       6008.612
## draining      3050.026
## biden         2583.929
## fake          1031.944
```

fnn, quarantine and rocky have the strongest relationships with retweets