Twitter Data Mining

Tianwen Huan

12/10/2016

```
# Install the Sentiment Package
# if (!require('pacman')) install.packages('pacman&')
# pacman::p_load(devtools, installr)
# install_url('http://cran.r-project.org/src/contrib/Archive/Rstem/Rstem_0.4-1.tar.gz')
# install_url('http://cran.r-project.org/src/contrib/Archive/sentiment/sentiment_0.2.tar.gz')
```

```
# Install the Graph Package
# source("https://bioconductor.org/biocLite.R")
# biocLite("BiocInstaller")
# biocLite("graph")
# biocLite("Rgraphviz")
```

Getting Data

```
Trump <- read.csv("~/Documents/BU/MA 615 R/12.12final project/Trump.csv", comment.char="#")
#save(Trump, file = "Trump.RData")
```

Map

Clean Data for Maps

```
points <- data.frame(x=as.numeric(Trump$place_lon),
                     y=as.numeric(Trump$place_lat))

points <- points[points$y<50,]
points <- points[points$y>20,]
points <- points[points$x< -60,]
points <- points[points$x>-125,]

#save(points, file = "map.RData")
```

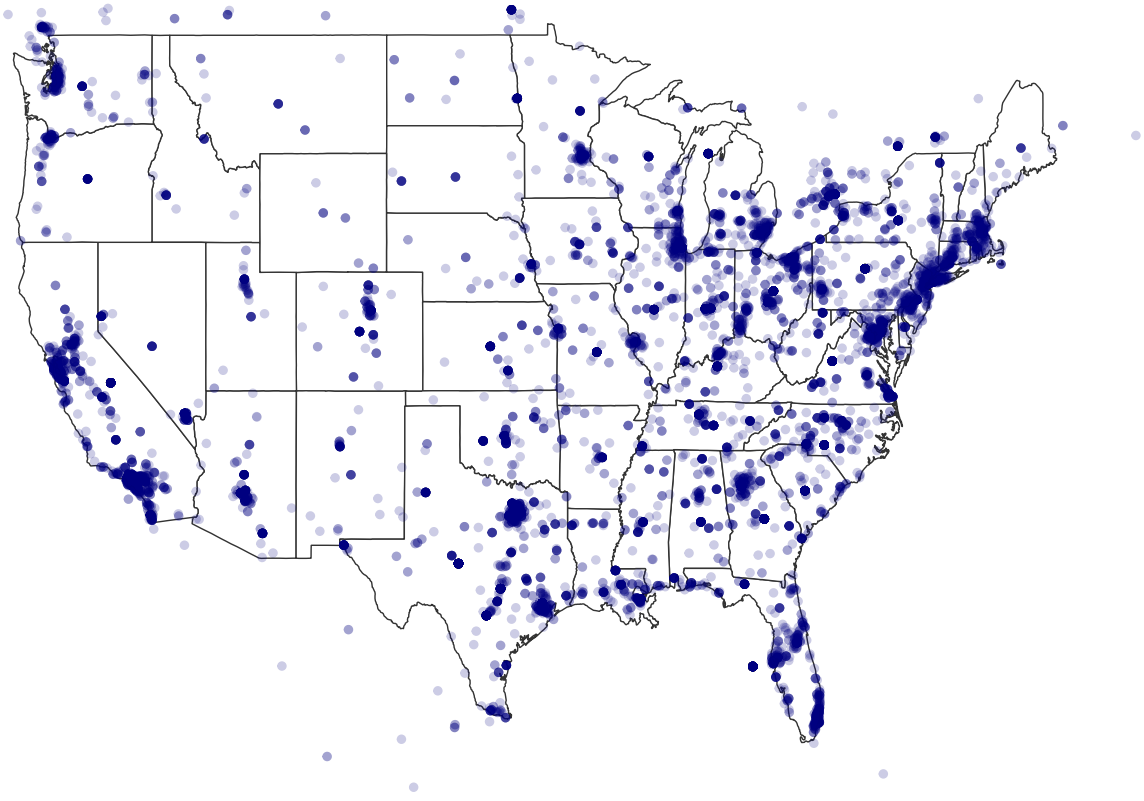
Dot Plot-ggplot

```
map.data <- map_data("state")
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:plyr':
##
##     ozone
```

```
ggplot(map.data)+
  geom_map(aes(map_id=region),
           map=map.data,
           fill="white",
           color="grey20", size=0.25)+
  expand_limits(x=map.data$place_long,y=map.data$place_lat)+
  theme(axis.line=element_blank(),
        axis.text=element_blank(),
        axis.ticks=element_blank(),
        axis.title=element_blank(),
        panel.background=element_blank(),
        panel.border=element_blank(),
        panel.grid.major=element_blank(),
        plot.background=element_blank(),
        plot.margin=unit(0*c(-1.5,-1.5,-1.5,-1.5),"lines"))+
  geom_point(data=points,
             aes(x=x,y=y),size=1,
             alpha=1/5,color="navy")
```



Data Mining

Clean Data for Data Mining

```
############## Create corpus
# build a corpus, and specify the source to be character vectors
Trump$text <- gsub("[^[:alnum:]///' ]", "", Trump$text)
```

```
Trump$created <-  as.POSIXct(strptime(Trump$user_created_at, "%a %b %d %H:%M:%S %z %Y"))
Trump <- Trump %>% arrange(created)
corpus <- Corpus(VectorSource(Trump$text))

# convert to lower case
corpus <- tm_map(corpus, content_transformer(tolower))

# remove URLs
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
corpus <- tm_map(corpus, content_transformer(removeURL))

# remove anything other than English letters or space
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
corpus <- tm_map(corpus, content_transformer(removeNumPunct))

# remove stopwords
myStopwords <- c(setdiff(stopwords('english'), c("Trump")),
                 "just", "will", "im", "like", "dont", "one", "can", "get", "now")
corpus <- tm_map(corpus, removeWords, myStopwords)

# remove extra whitespace
corpus <- tm_map(corpus, stripWhitespace)

# convert corpus to a Plain Text Document
corpus <- tm_map(corpus,PlainTextDocument)

# replace oldword with newword
replaceWord <- function(corpus, oldword, newword) {
  tm_map(corpus, content_transformer(gsub),
         pattern=oldword, replacement=newword)}

corpus <- replaceWord(corpus, "trumps", "trump")
corpus <- replaceWord(corpus, "donald", "trump")
corpus <- replaceWord(corpus, "russian", "russia")
corpus <- replaceWord(corpus, "american", "america")
corpus <- replaceWord(corpus, "jobs", "job")
```

Build Term Doc Matrix

```
# count word frequence
tdm <- TermDocumentMatrix(corpus, control = list(wordLengths = c(1, Inf)))
term.freq <-sort(rowSums(as.matrix(tdm)), decreasing=TRUE)
term.freq <- subset(term.freq, term.freq >= 50)

df <- data.frame(term = names(term.freq), freq = term.freq)
head(df, 10)
```

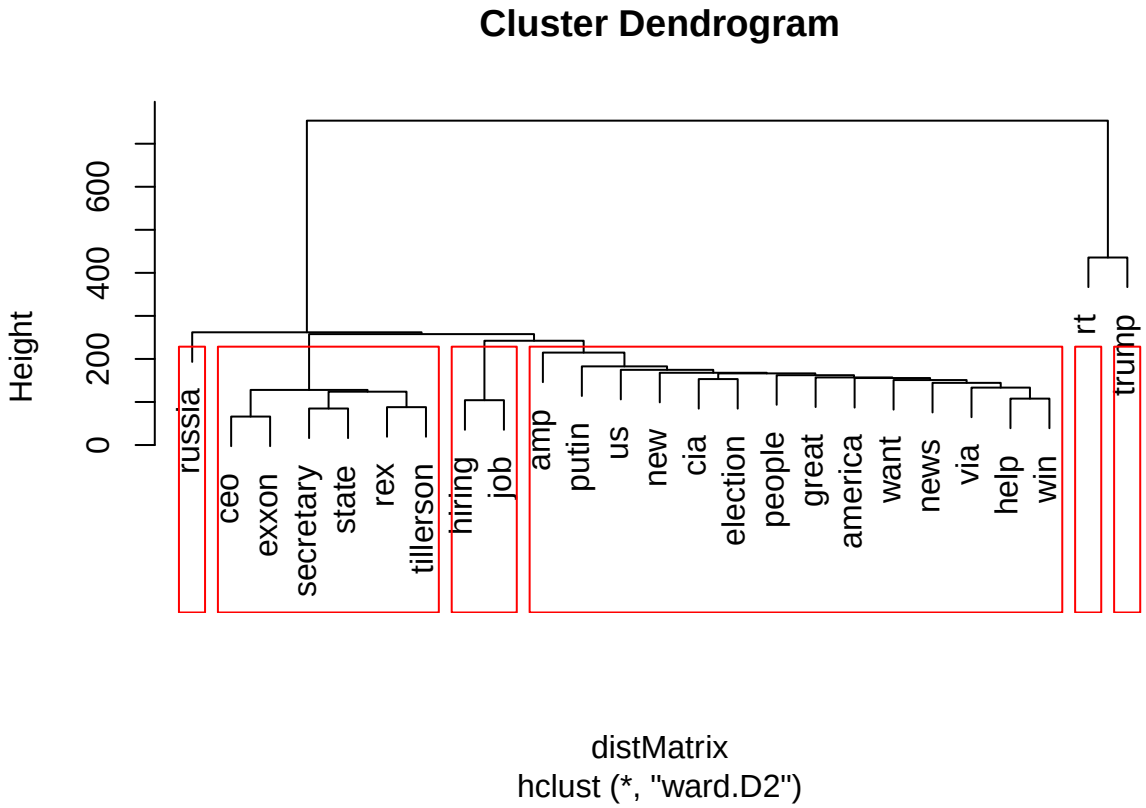
```
##                term  freq
## trump         trump 14721
## rt               rt 11717
## russia       russia  3198
## state         state  1816
```

```
## secretary secretary  1632
## tillerson tillerson  1618
## putin         putin  1468
## cia             cia  1408
## amp             amp  1314
## job             job  1281
```

```
#save(tdm,file="tdm.RData")
#save(df,file="worldcloud.RData")
```

Cluster Dendrogram

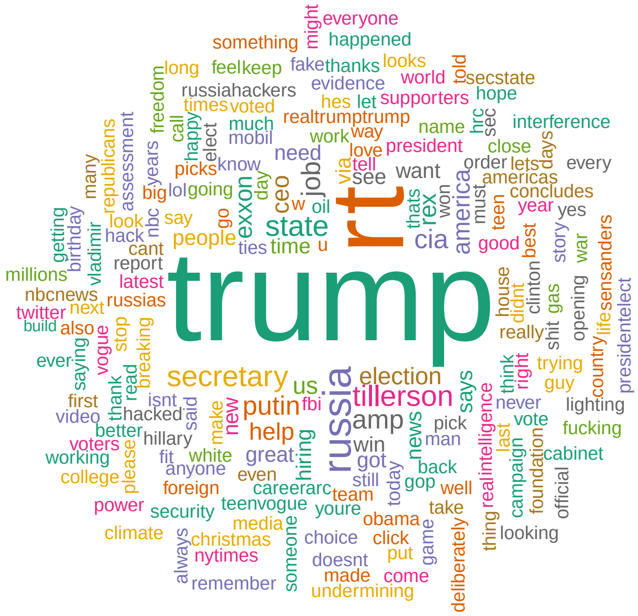
```
#remove sparse terms
tdm2 <- removeSparseTerms(tdm, sparse = 0.98)
m2 <- as.matrix(tdm2)

#cluster terms
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "ward.D2")
plot(fit)
rect.hclust(fit, k=6) #cut tree into 6 clusters
```



Word Cloud

```
# plot world cloud
set.seed(1)
```

```r
wordcloud(words = df$term, freq = df$freq, min.freq = 60,
          random.color = TRUE, random.order=FALSE,
          max.words=200, rot.per=0.35, colors=brewer.pal(8, "Dark2"))
```



Find Word Association

```r
findAssocs(tdm, "rt", 0.3)
```

```
## $rt
## trump
##  0.47
```

```r
findAssocs(tdm, "russia", 0.3)
```

```
## $russia
##      cia election
##     0.34     0.32
```

```r
findAssocs(tdm, "secretary", 0.3)
```

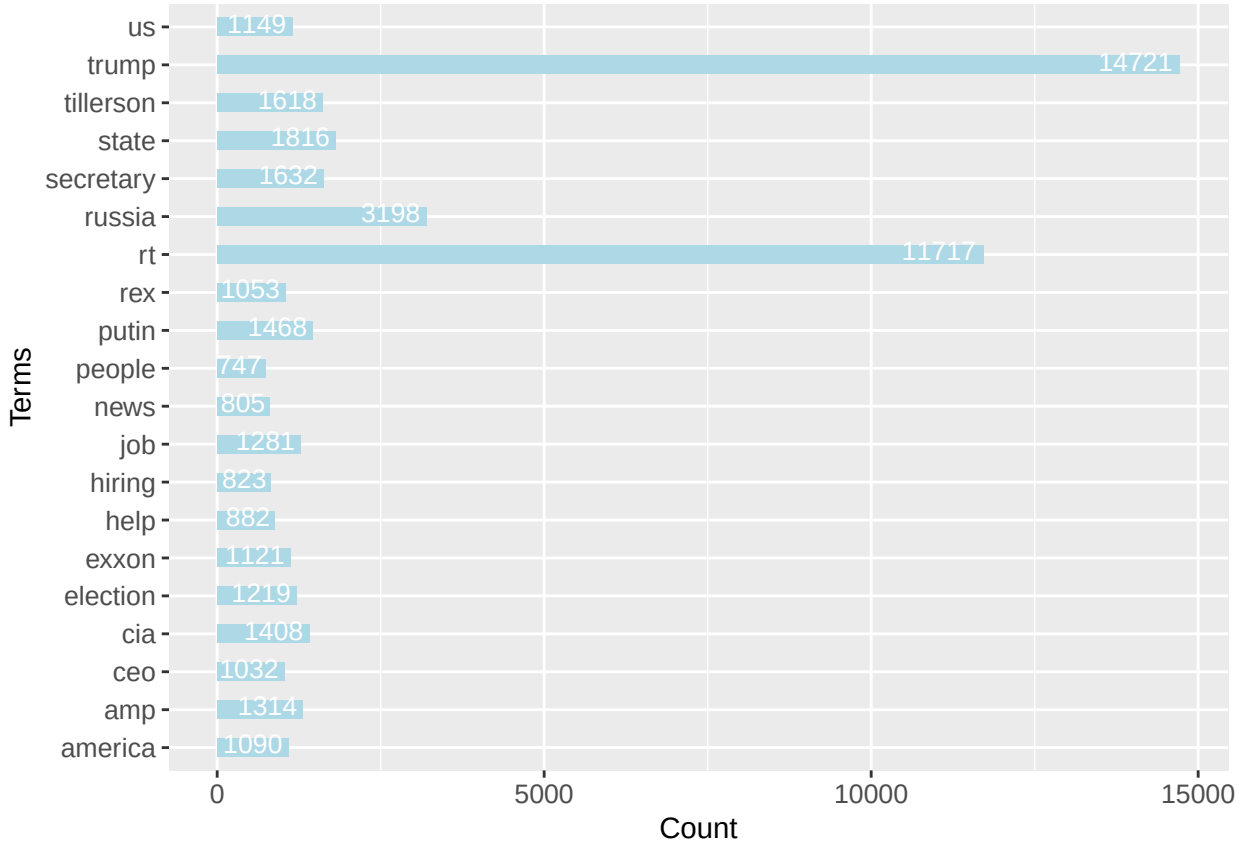
```
## $secretary
##     state tillerson       ceo     exxon       rex     mobil      name
##      0.82      0.67      0.62      0.60      0.54      0.41      0.31
##     picks
##      0.30
```

```r
findAssocs(tdm, "job", 0.3)
```

```
## $job
##       hiring    careerarc       latest      opening        click        apply
##         0.80         0.63         0.52         0.50         0.48         0.40
##         view         work hospitality    recommend          fit
##         0.39         0.38         0.34         0.31         0.30
```

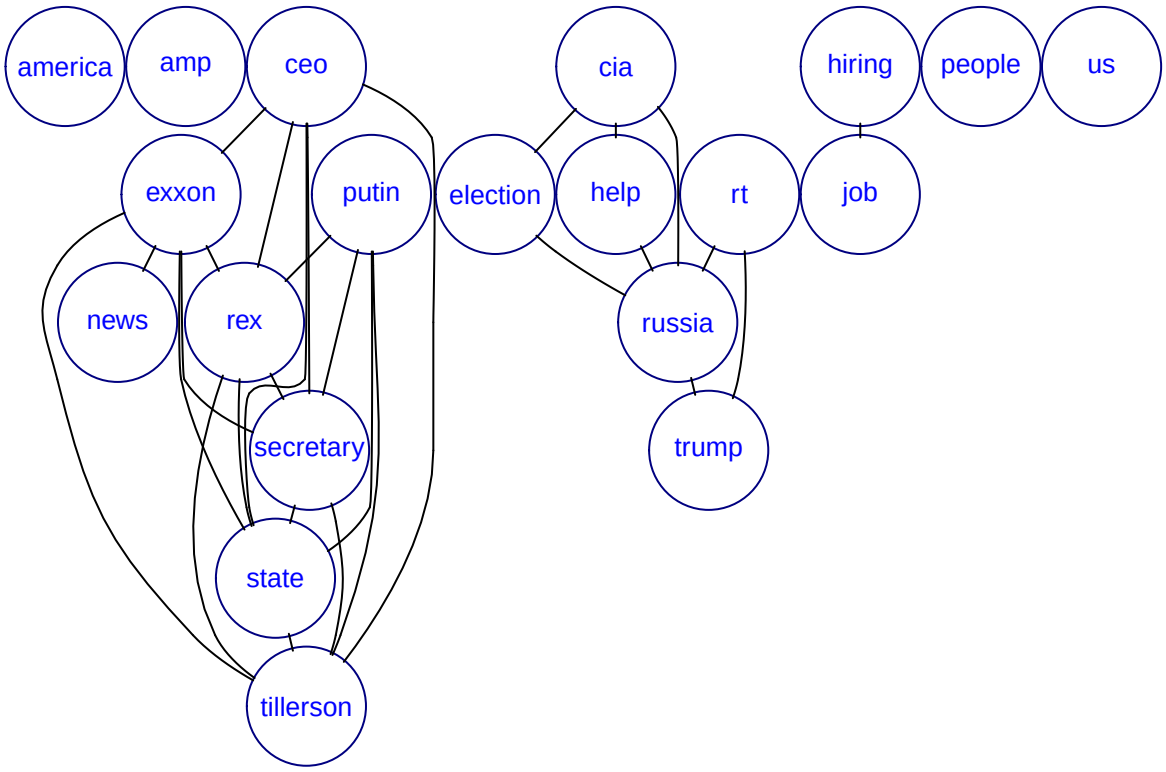
Barplot for Top Frenquency Words

```
df20 <- df %>% filter(freq>700)

ggplot(df20, aes(x=term, y=freq)) + geom_bar(stat="identity", width=0.5, fill="lightblue") +
  xlab("Terms") + ylab("Count") + coord_flip() +
  theme(axis.text=element_text(size=10)) +
  geom_text(aes(label=freq), vjust=0.3, hjust=1.1, color="white", size=3.5)
```



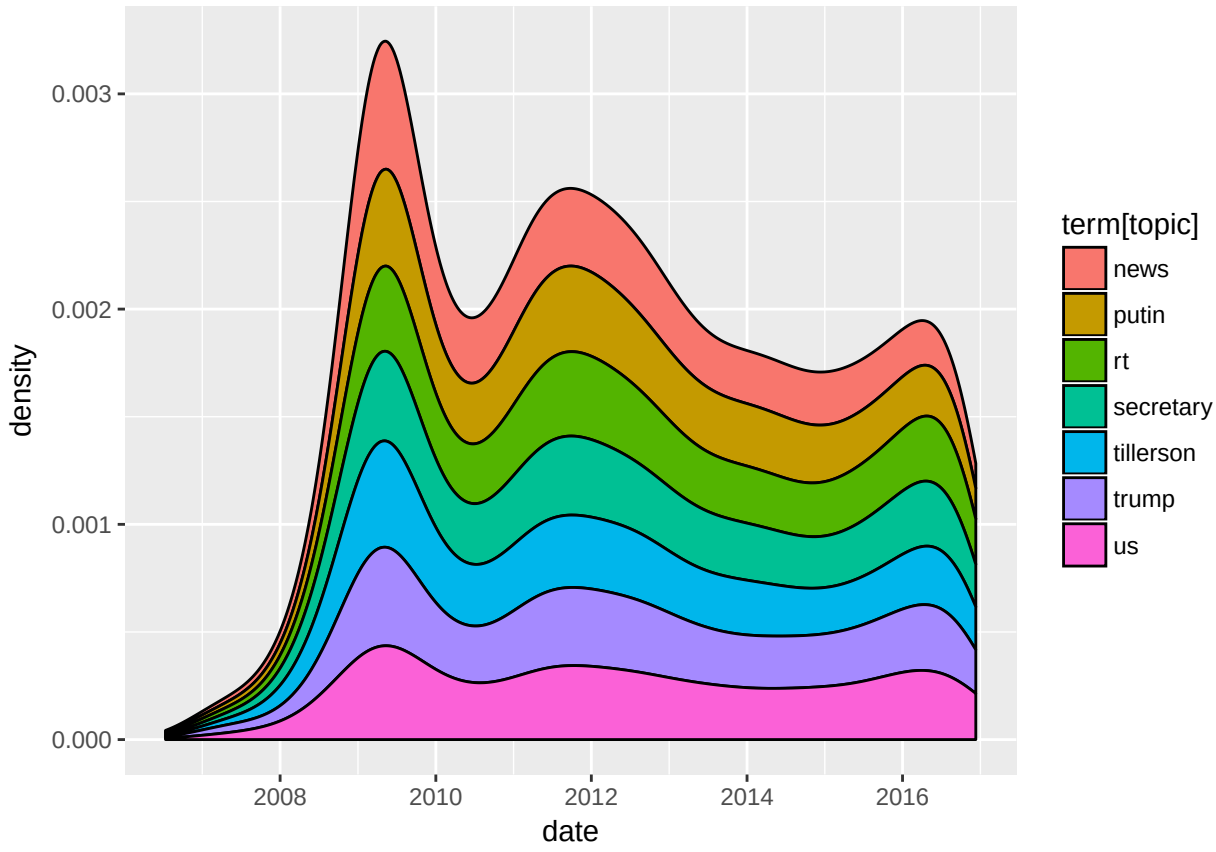
Word Association Graph

```
plot(tdm, terms=findFreqTerms(tdm, lowfreq=700), corThreshold = 0.2,
     attrs=list(node=list(width=20, fontsize=14, fontcolor="blue", color="navy")))
```



Topic Modelling

```
dtm <- as.DocumentTermMatrix(tdm)
rowTotals <- apply(dtm , 1, sum) # find the sum of words in each Document
dtm.new   <- dtm[rowTotals> 0, ] # remove all docs without words
#save(dtm.new,file="dtm.new.RData")
lda <- LDA(dtm.new, k = 8) # find 8 topics
term <- terms(lda, 7) # first 7 terms of every topic
topics <- topics(lda) # 1st topic identified for every document (tweet)

Tr <- Trump[rowTotals> 0, ]
#save(Tr,file="Tr.RData")
topics <- data.frame(date=as.IDate(Tr$created), topic=topics)

ggplot(topics, aes(date, fill = term[topic])) +
  geom_density(position = "stack")
```



Sentiment Analysis

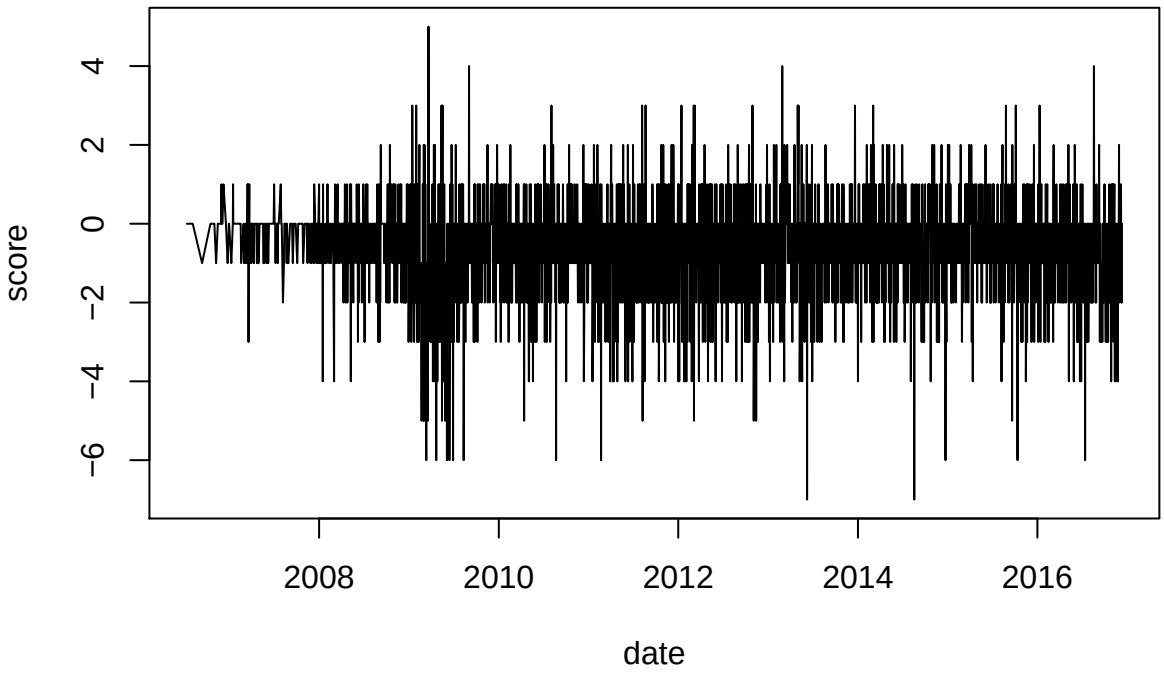
```
# install package sentiment140
require(devtools)
```

```
## Loading required package: devtools
```

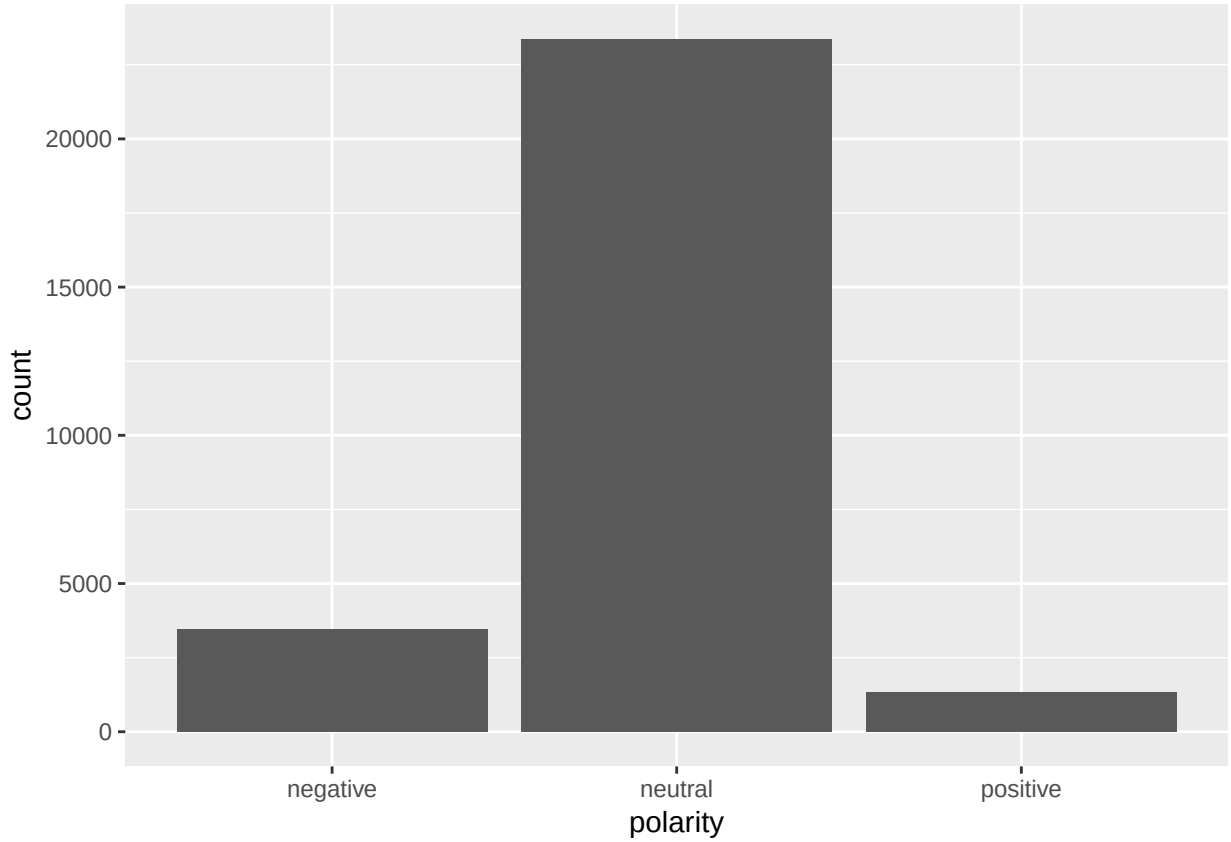
```
install_github("sentiment140", "okugami79")
```

```
## Skipping install of 'sentiment' from a github remote, the SHA1 (75be56d6) has not changed since last
##   Use `force = TRUE` to force installation
```

```
# sentiment analysis
library(sentiment)
sentiments <- sentiment(Trump$text)

## sentiment plot
sentiments$score <- 0
sentiments$score[sentiments$polarity == "positive"] <- 1
sentiments$score[sentiments$polarity == "negative"] <- -1
sentiments$date <- as.IDate(Trump$created)
result <- aggregate(score ~ date, data = sentiments, sum)
plot(result, type = "l")
```




```
qplot(polarity, data=sentiments) # Polarity table
```



Top Retweeted Tweetsop

```
# select top retweeted tweets
selected <- which(Trump$retweet_count >= 27000)

# plot them
dates <- strptime(Trump$created, format="%Y-%m-%d")
plot(x=dates, y=Trump$retweet_count, type="l", col="grey",
     xlab="Date", ylab="Times retweeted")

# plot points and text
colors <- rainbow(length(selected))[1:length(selected)]
points(dates[selected], Trump$retweet_count[selected],
       pch=19, col=colors)

text(dates[selected], Trump$retweet_count[selected],
     Trump$text[selected], col=colors, cex=.9)
```