Sentiment Analysis - Text Mining

Group 9 | Sai, Hari, Raj, Dinesh and Santhosh

September 17, 2017

** Bit Coin **

** What is a Bit coin and How social community looks at it. Does they have positive sentiment or Negative? All these questions can be answered using text Mining with the use of Twitter Data. It required lots of Text cleaning, building world cloud, finding most frequent words tweeted, which is the hot trending topic etc.**

Setting up the Environment and Initiating Required Libraries

```
#Library Needed for Text Mining
library(tm)
```

```
## Loading required package: NLP
```

```
#for string operations
library(stringi)

#for finding word Freequency
library(SnowballC)

#Other Required Libraries
library(RColorBrewer)
library(wordcloud)
library(qdap)
```

```
## Loading required package: qdapDictionaries
```

```
## Loading required package: qdapRegex
```

```
## Loading required package: qdapTools
```

```
##
## Attaching package: 'qdap'
```

```
## The following objects are masked from 'package:tm':
##
##     as.DocumentTermMatrix, as.TermDocumentMatrix
```

```
## The following object is masked from 'package:NLP':
##
##     ngrams
```

```
## The following object is masked from 'package:base':
##
##     Filter
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:qdapRegex':
##
##     %+%
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(topicmodels)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:qdapTools':
##
##     shift
```

```
library(sentiment)
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
## Loading required package: rjson
```

```
## Loading required package: plyr
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:qdapTools':
##
##     id
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following object is masked from 'package:qdap':
##
##     %>%
```

```
## The following object is masked from 'package:qdapTools':
##
##     id
```

```
## The following object is masked from 'package:qdapRegex':
##
##     explain
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(streamgraph)
```

```
##
## Attaching package: 'streamgraph'
```

```
## The following object is masked from 'package:qdap':
##
##     %>%
```

Functions and Methods for carrying out repeated tasks

Function for cleaning the data

```
cleanData = function(AnyText){
  some_txt = AnyText
  some_txt = genX(some_txt, " <", ">")
  some_txt = gsub(" {0,}http([[:punct:]]|[[:digit:]]|[a-z|A-Z])+", " ", some_txt)
  #Removing Single Characters
  some_txt = gsub("[[:punct:]]", " ", some_txt)
  some_txt = gsub("[[:digit:]]", " ", some_txt)
  some_txt = gsub("( ){1,}. "," ", some_txt)
  some_txt = gsub(" .( ){1,}"," ", some_txt)
  some_txt = gsub("^. {1,}"," ", some_txt)
  some_txt = gsub("{1,}.$"," ", some_txt)

  #Removing Numbers
  some_txt = gsub("[0-9]+","", some_txt)

  some_txt = gsub("(RT|via)((?:\b\\W*@\\w+)+)", " ", some_txt,perl=TRUE)
  some_txt = gsub("@\\w+", " ", some_txt)
  #some_txt = gsub("[ t]{2,}", " ", some_txt)
  #some_txt = gsub("\\t", "", some_txt)
  some_txt = gsub("^\\s+|\\s+$", " ", some_txt)
  some_txt = gsub("amp", "", some_txt)
  some_txt = gsub("^RT[ ]+","", some_txt)
  some_txt = gsub("[^[:alpha:][:space:]]*", "", some_txt)
  some_txt = gsub("[ ]{2,}"," ", some_txt)
  some_txt = gsub("(\b|^)[a-zA-Z0-9]{2,2}( |\n|$)+","", some_txt)
  #some_txt = gsub("(^ )|( $)","", some_txt)
  return(some_txt)
}
```

Dictionary Set-Up for completing the mis

```
stemCompletion2 <- function(x,dictionary) {
  x <- unlist(strsplit(as.character(x)," "))
  x <- x[x !=""]
  x <- stemCompletion(x, dictionary = dictionary)
  x <- paste(x, sep="", collapse=" ")
  PlainTextDocument(stripWhitespace(x))
}
```

Replacing the incorrect words

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

Importing the Twitter Data : Data about “Bitcoin”

```
#Twitter Data
TweetDF = read.csv("C:/Home/Work/GreatLakes/Web and Social Media Analytics/Assignment/bit
coin.csv", header = T, sep=",")
```

Cleaning the Data and ensuring minial duplicate tweets

```
CleandedTWeetData = sapply(TweetDF$text, function(t) cleanData(t))
CleandedTWeetData[150]
```

```
## [1] "China Shmyna Bitcoin Trading Is Way More Distributed Now Anyway via CoinDes "
```

```
RepeatTweets = data.frame(table(CleandedTWeetData))
RepeatTweets = RepeatTweets[order(RepeatTweets$Freq, decreasing = FALSE),]
nrow(RepeatTweets)
```

```
## [1] 663
```

```
nrow(unique(RepeatTweets))
```

```
## [1] 663
```

```
CleandedTWeetData = unique(CleandedTWeetData)
```

As both has 663 records, We can say that there is no duplicate Tweets

Creating Data Corpus and Preparing and Cleaning the data

```
#Corpus
TweetDataCorpus = VCorpus(VectorSource(CleandedTWeetData))
TweetDataCorpus = tm_map(TweetDataCorpus, content_transformer(stri_trans_tolower))
TweetDataCorpus = tm_map(TweetDataCorpus, content_transformer(cleanData))
# Removing Stop Words
enStopWords<- c((stopwords('en')),c("rt", "use", "used", "via", "amp", "http","https","ch
aracter","hour","year","id","min","datetimestamp","description","isdst","heading","langua
ge","meta","mday","mon","wday","yday","listcontent","day","re", "itt"))
TweetDataCorpus = tm_map(TweetDataCorpus,removeWords , enStopWords)
```

Creating a copy of the corpus to perform dictionary operations

```
TweetDataCorpusDict = TweetDataCorpus
TweetDataCorpusDict = unique(unlist(strsplit(as.character(TweetDataCorpusDict)," ")))
TweetDataCorpusWords = unlist(strsplit(as.character(TweetDataCorpus)," "))
TweetDataCorpusDict = Corpus(VectorSource(TweetDataCorpusDict))
```

Stemming the data

```
TweetDataCorpus = tm_map(TweetDataCorpus, stemDocument)
writeLines(strwrap(TweetDataCorpus[[150]]$content,60))
```

```
## gregahorvatfx deep pullback cryptocurr bitcoin litecoin
## updat near term weak lik
```

```
#####Stem Complete and Display the same tweet above with the completed and corrected tex
t.
TweetDataCorpus <- lapply(TweetDataCorpus, stemCompletion2, dictionary=TweetDataCorpusDic
t)
TweetDataCorpus <- VCorpus(VectorSource(TweetDataCorpus))
writeLines(strwrap(TweetDataCorpus[[150]]$content, 60))
```

```
## deep pullbacks cryptocurre bitcoin litecoin update near
## term weakness lik
```

All the data have been stemmed. if we notice there are still some words which are incomplete. That we will treat it by using other cord correction mechanics.

Identifying and correcting misspelt words if any and that too if it is impacting in squeezing data

```
TokenWords = sapply(TweetDataCorpusWords, function(x) gsub("([[:punct:]]| {1,}|[0-9])",
"", x))
TokenWords = data.frame(TokenWords)
TokenWords = data.frame(table(TokenWords))
TokenWords = TokenWords[order(TokenWords$Freq, decreasing = T),]
head(as.matrix(TokenWords), 40)
```

```
##      TokenWords       Freq
## 1    ""               "24784"
## 418  "character"      " 2652"
## 762  "en"             "  665"
## 625  "datetimestamp"  "  663"
## 656  "description"    "  663"
## 1050 "heading"        "  663"
## 1087 "hour"           "  663"
## 1119 "id"             "  663"
## 1190 "isdst"          "  663"
## 1264 "language"       "  663"
## 1399 "mday"           "  663"
## 1416 "meta"           "  663"
## 1429 "min"            "  663"
## 1457 "mon"            "  663"
## 1584 "origin"         "  663"
## 2341 "wday"           "  663"
## 2416 "yday"           "  663"
## 2418 "year"           "  663"
## 1308 "listcontent"    "  657"
## 1307 "listauthor"     "  653"
## 1311 "listsec"        "  643"
## 240  "bitcoin"        "  566"
## 2    "\n"             "  438"
## 334  "btc"            "  112"
## 281  "blockchain"     "   85"
## 430  "china"          "   64"
## 584  "cryptocurrency" "   63"
## 239  "bitcoi"         "   57"
## 1113 "ico"            "   57"
## 574  "crypto"         "   54"
## 1695 "price"          "   46"
## 796  "ethereum"       "   45"
## 814  "exchange"       "   42"
## 2370 "will"           "   41"
## 2290 "utrust"         "   39"
## 1306 "list\n"         "   35"
## 238  "bitco"          "   31"
## 1541 "now"            "   31"
## 2214 "trading"        "   31"
## 879  "fintech"        "   30"
```

```
TweetDataCorpus =  replaceWord(TweetDataCorpus, "bitcoi ", "bitcoin")
TweetDataCorpus =  replaceWord(TweetDataCorpus, "cryptocurren ", "cryptocurrency")
```

There are lots of Repeated and very common words in those corpus which will be removed as part of corpus cleaning and stop words treatment. Having those words will skew the results. Also, this list gives top 40 tweeted words related to Bitcoins

Building Term Document Matrix

```
TwitterTDM = TermDocumentMatrix(TweetDataCorpus, control= list(wordLengths= c(1, Inf)))
TwitterTDM
```

```
## <<TermDocumentMatrix (terms: 1787, documents: 663)>>
## Non-/sparse entries: 5866/1178915
## Sparsity           : 100%
## Maximal term length: 22
## Weighting          : term frequency (tf)
```

```
TwitterIdx <- which(dimnames(TwitterTDM)$Terms %in% c("bitcoin", "blockchain"))
as.matrix(TwitterTDM[TwitterIdx,21:30])
```

```
##             Docs
## Terms        21 22 23 24 25 26 27 28 29 30
##   bitcoin     1  1  1  1  0  0  1  2  2  1
##   blockchain  1  1  0  0  0  0  0  1  0  0
```

To identify the terms that are most used freequently

```
freq_terms = findFreqTerms(TwitterTDM, lowfreq = 20)
term.freq <- rowSums(as.matrix(TwitterTDM))
term.freq <- subset(term.freq, term.freq > 20)
term.freq
```

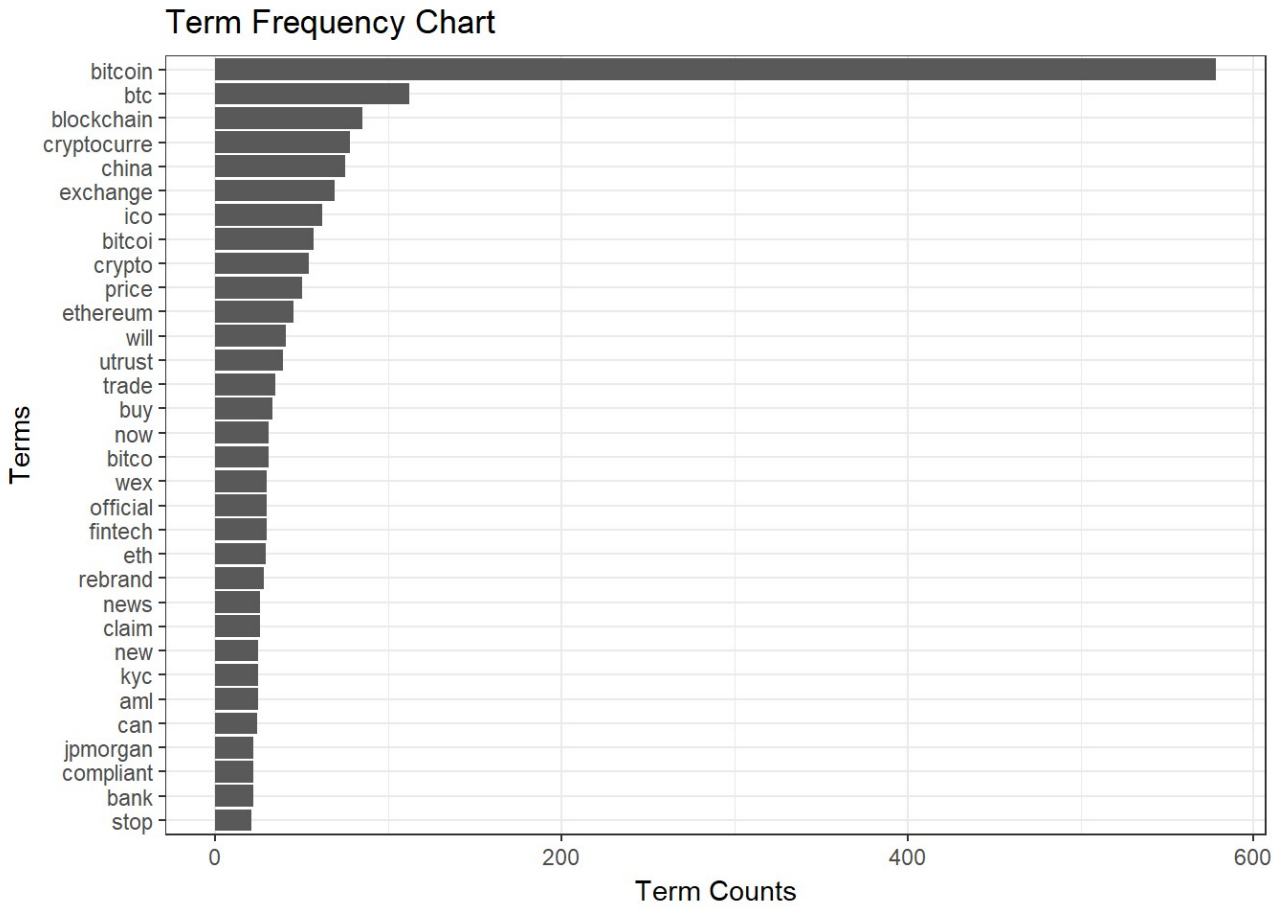
```
##         aml        bank       bitco      bitcoi     bitcoin  blockchain
##          25          22          31          57         578          85
##         btc         buy         can       china       claim   compliant
##         112          33          24          75          26          22
##      crypto cryptocurre         eth    ethereum    exchange     fintech
##          54          78          29          45          69          30
##         ico    jpmorgan         kyc         new        news         now
##          62          22          25          25          26          31
##    official       price     rebrand        stop       trade      utrust
##          30          50          28          21          35          39
##         wex        will
##          30          41
```

```
TermDataFrame <- data.frame(term = names(term.freq), freq= term.freq)
TermDataFrame
```

```
##                    term freq
## aml                 aml   25
## bank               bank   22
## bitco             bitco   31
## bitcoi           bitcoi   57
## bitcoin         bitcoin  578
## blockchain   blockchain   85
## btc                 btc  112
## buy                 buy   33
## can                 can   24
## china             china   75
## claim             claim   26
## compliant     compliant   22
## crypto           crypto   54
## cryptocurre cryptocurre   78
## eth                 eth   29
## ethereum       ethereum   45
## exchange       exchange   69
## fintech         fintech   30
## ico                 ico   62
## jpmorgan       jpmorgan   22
## kyc                 kyc   25
## new                 new   25
## news               news   26
## now                 now   31
## official       official   30
## price             price   50
## rebrand         rebrand   28
## stop               stop   21
## trade             trade   35
## utrust           utrust   39
## wex                 wex   30
## will               will   41
```

Plotting the Term Freequency as a graph

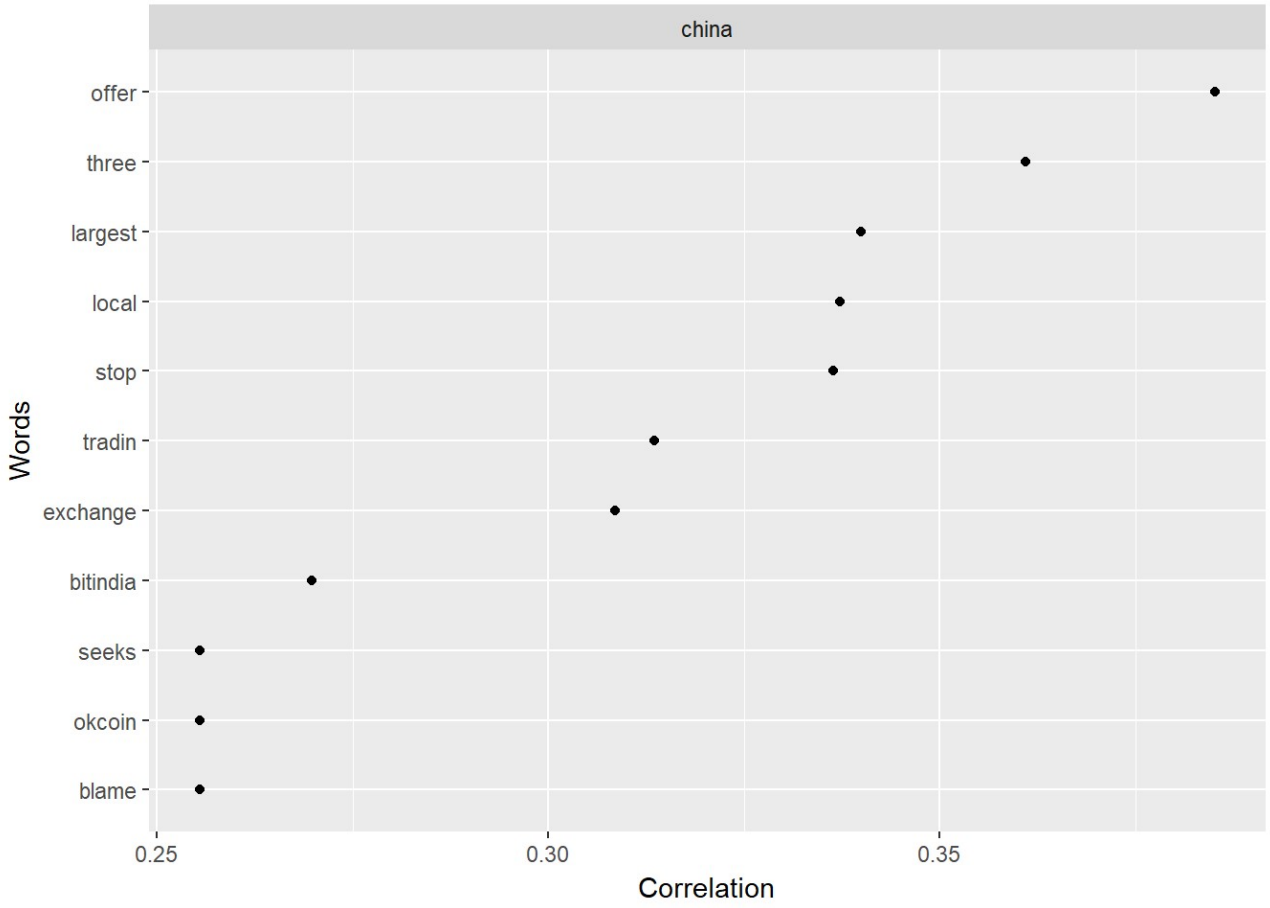
```
ggplot(TermDataFrame, aes(reorder(term, freq),freq)) + theme_bw() + geom_bar(stat = "iden
tity")  + coord_flip() +labs(list(title="Term Frequency Chart", x="Terms", y="Term Count
s"))
```



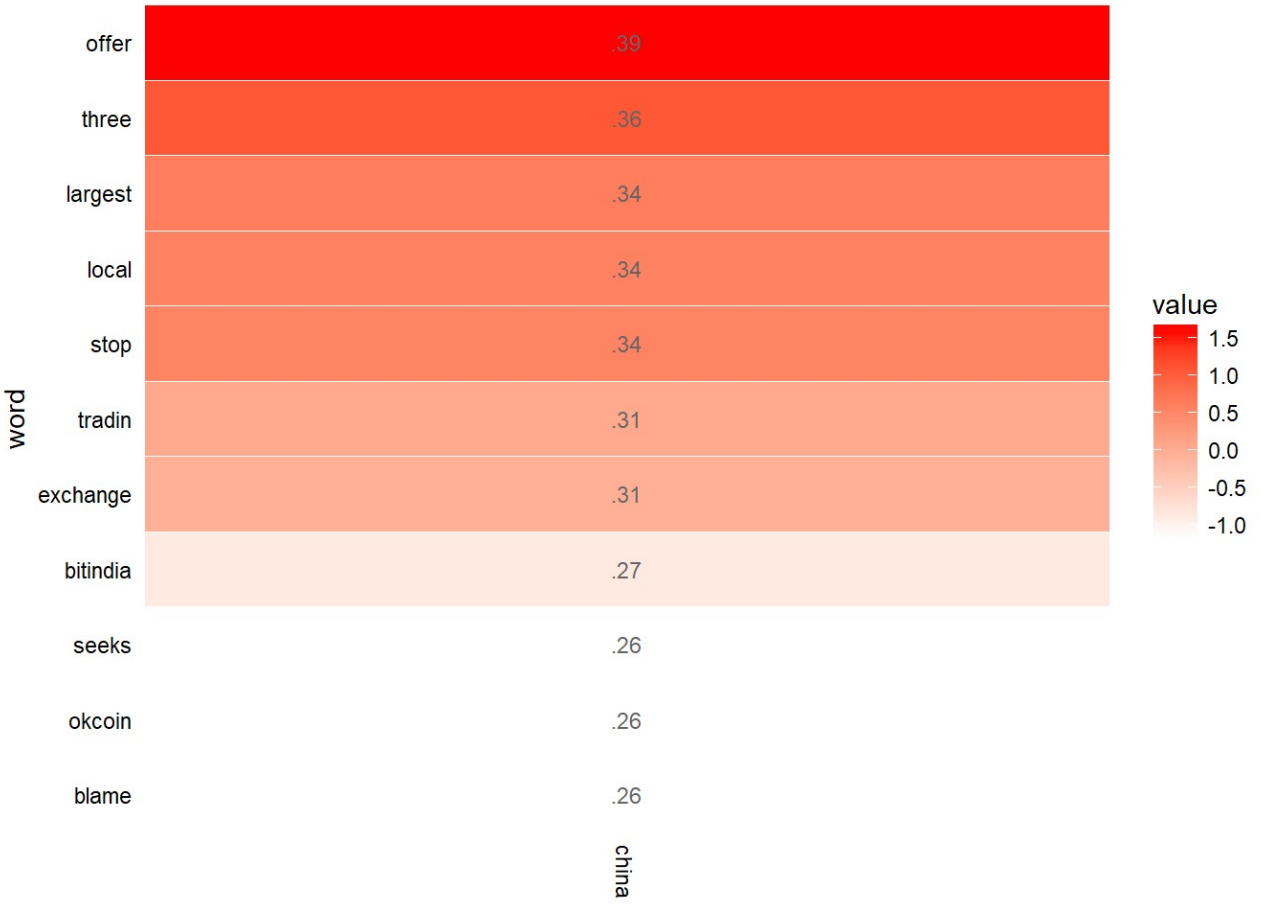
As it is all about “bitcoin” we are seeing huge chunk of tweets has that word. When we look at this, China is mentioned as 5th most tweeted word along with bit coin. as no other country we can find in this list other than China. We can say that China is positively or negatively impacting the bitcoin.**#####calculate the frequency of words and sort it by frequency and setting up the Wordcloud

```
TweetWordCloud  <-sort(rowSums(as.matrix(TwitterTDM)), decreasing= F)
pal<- brewer.pal(8, "Dark2")
wordcloud(words = names(TweetWordCloud), freq =TweetWordCloud, min.freq = 2, random.orde
r = F, colors = pal, max.words = 100)
```


```
# Identify and plot word correlations. For example
WordCorr <- apply_as_df(TweetDataCorpus[1:500], word_cor, word = "china", r=.25)
plot(WordCorr)
```




```
qheat(vect2df(WordCorr[[1]], "word", "china"), values=TRUE, high="red",
digits=2, order.by ="china", plot = FALSE) + coord_flip()
```




```
# Messages with word - love
Termdf <- data.frame(text=sapply(TweetDataCorpus, `[[`, "content"), stringsAsFactors=FALS
E)
head(unique(Termdf[grep("china", Termdf$text), ]), n=10)
```

```
##  [1] "remind vladuceanu bitcoin price maintains valu china situation become irreleva
n"
##  [2] " bitcoin trade lower chinese exchange global volume even china quite spread okco
in doin"
##  [3] "bitcoin price maintains valu china situation become irreleva
n"
##  [4] " china three largest bitcoin exchange will stop offer local trade jonruss
e"
##  [5] "officialmcafee india china war cryptocurre india china done bitindia bitc
o"
##  [6] "weekend recap everything want know bitcoin blockchain china distrig
g"
##  [7] "bitcoin plummet rise china worrie devli
f"
##  [8] " tech china three largest bitcoin exchange will stop offer local tradi
n"
##  [9] " google gmail youtub facebook twitter wikipedia now bitcoin exchange strangers c
hina triangle ban r"
## [10] " org good back zug even council accept bitcoin word sinc chinaicoban companies f
looding"
```

Finding association with a specific keyword in the tweets

```
findAssocs(TwitterTDM, "india", 0.2)
```

```
## $india
##       mulls        govt    consider    bitindia        call      lakshm
##        0.52        0.47        0.44        0.43        0.42        0.37
##     lakshmi         may     similar        done     central        name
##        0.37        0.36        0.31        0.30        0.25        0.25
##        fiat       legal     studies      africa      arabia       bitcu
##        0.24        0.24        0.24        0.21        0.21        0.21
##      brasil      canada    dislikes       grant        raje rajeshbhaya
##        0.21        0.21        0.21        0.21        0.21        0.21
##     reserve       rupee       saudi       south   switzerla    tailspin
##        0.21        0.21        0.21        0.21        0.21        0.21
##      tender
##        0.21
```

```
findAssocs(TwitterTDM, "china", 0.2)
```

```
## $china
##    three  largest    local    offer     stop   tradin exchange   closur
##     0.35     0.33     0.33     0.33     0.33     0.30     0.29     0.25
##      war  beijing bitindia    japan      soc     amid    blame   okcoin
##     0.25     0.22     0.22     0.22     0.22     0.21     0.21     0.21
##    order    seeks
##     0.21     0.21
```

** When finding association words for India and China w.r.t bit coin. We can see that there are lot other countries associated with India with respect to bitcoin. We can assume that there is a prospect of using bitcoins for India when trading with other countries. India and “mull”(considers) is 0.52, we have to notice. and there is no significant negative word for India. for China, we see “Stop” as 0.33, war as 0.25, blame as “0.21”. We can assume that though in china many people speak about bit coin, as a country it is still conservative in implementing it.**

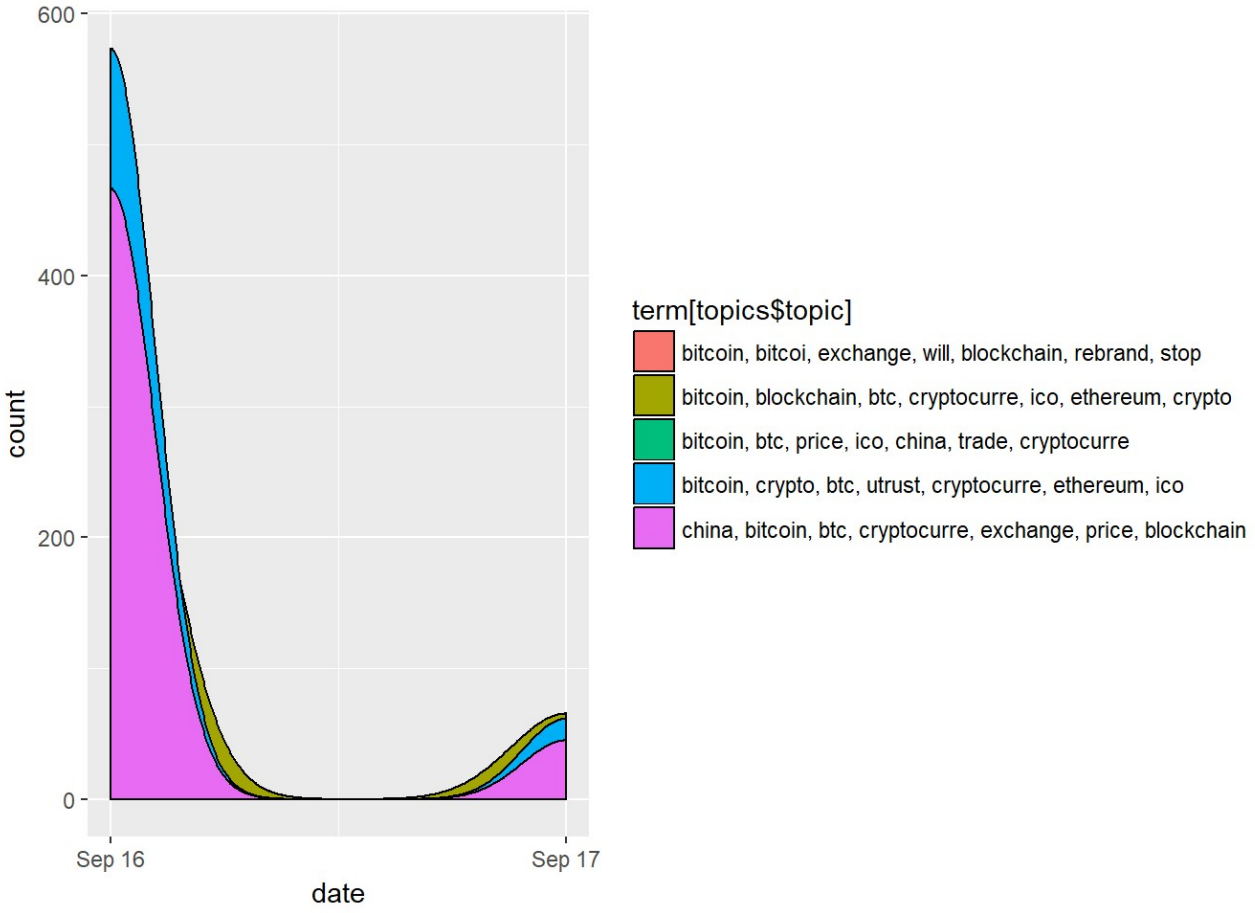
Topic Modeling to identify latent/hidden topics using Linear Dicriminant technique

```
dtm <- as.DocumentTermMatrix(TwitterTDM)

rowTotals <- apply(dtm , 1, sum)

#NullDocs <- dtm[rowTotals==0, ]
NonNullDocs <- dtm[rowTotals!=0, ]
dtm   <- dtm[rowTotals > 0, ]



if (length(NonNullDocs$dimnames$Docs) > 0) {
  TweetDF <- TweetDF[as.numeric(NonNullDocs$dimnames$Docs),]
}


lda <- LDA(dtm, k = 5) # find 5 topic
term <- terms(lda, 7) # first 7 terms of every topic
(term <- apply(term, MARGIN = 2, paste, collapse = ", "))
```

```
##                                                          Topic 1
##           "bitcoin, btc, price, ico, china, trade, cryptocurre"
##                                                          Topic 2
##    "bitcoin, bitcoi, exchange, will, blockchain, rebrand, stop"
##                                                          Topic 3
## "china, bitcoin, btc, cryptocurre, exchange, price, blockchain"
##                                                          Topic 4
##  "bitcoin, blockchain, btc, cryptocurre, ico, ethereum, crypto"
##                                                          Topic 5
##      "bitcoin, crypto, btc, utrust, cryptocurre, ethereum, ico"
```

```
topics<- topics(lda)
topics<- data.frame(date=as.Date(TweetDF$created), topic = topics)
qplot (date, ..count.., data=topics, geom ="density", fill= term[topics$topic], position
="stack")
```

```
## Warning: `position` is deprecated
```



Sentiment Analysis to identify positive/negative tweets

```
# Use qdap polarity function to detect sentiment
sentiments <- polarity(TweetDF$text)
```

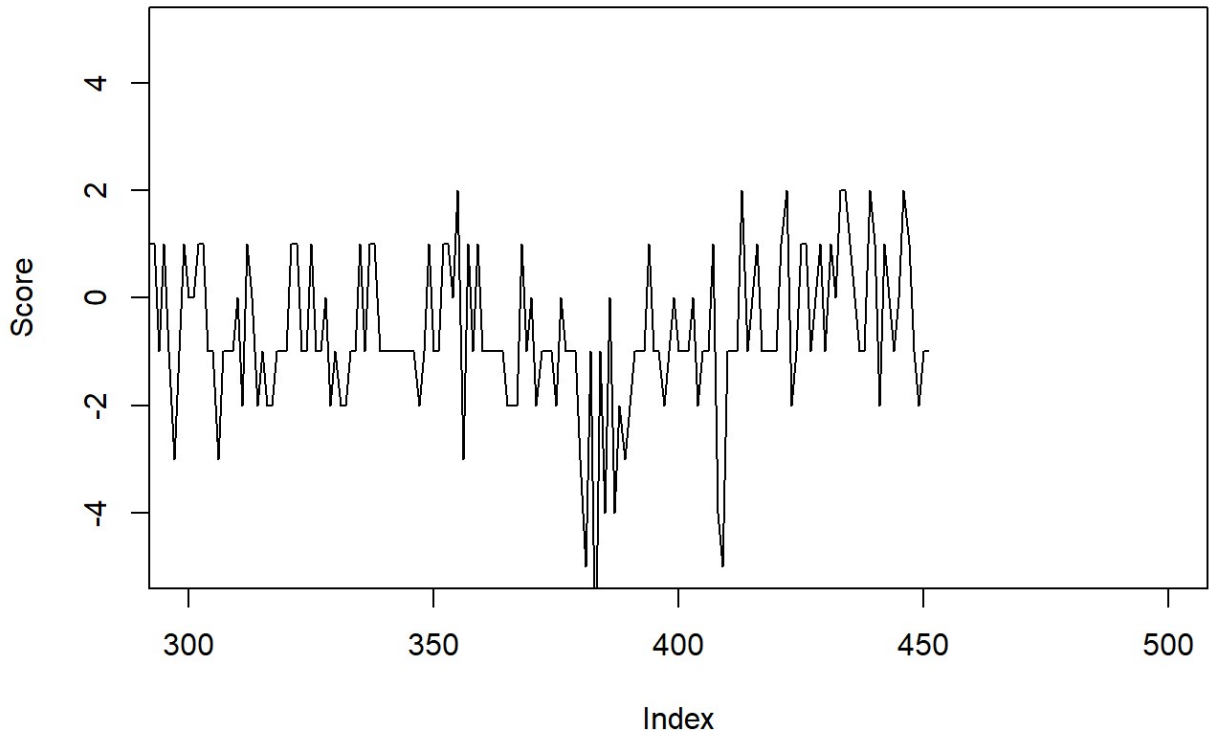
```
## Warning in polarity(TweetDF$text):
##   Some rows contain double punctuation.  Suggested use of `sentSplit` function.
```

```
sentiments <- data.frame(sentiments$all$polarity)

sentiments[["polarity"]] <- cut(sentiments[[ "sentiments.all.polarity"]], c(-5,0.0,5), la
bels = c("negative","positive"))

table(sentiments$polarity)
```

```
##
## negative positive
##      446      217
```

Sentiment Plot by date

```
sentiments$score<- 0
sentiments$score[sentiments$polarity == "positive"]<- 1
sentiments$score[sentiments$polarity == "negative"]<- -1
sentiments$date <- (TweetDF$created)
result <- aggregate(score ~ TweetDF$created, data = sentiments, sum)
plot(x= result$score, type = "l", xlim = c(300,500), ylim = c(-5,5), ylab = "Score")
```

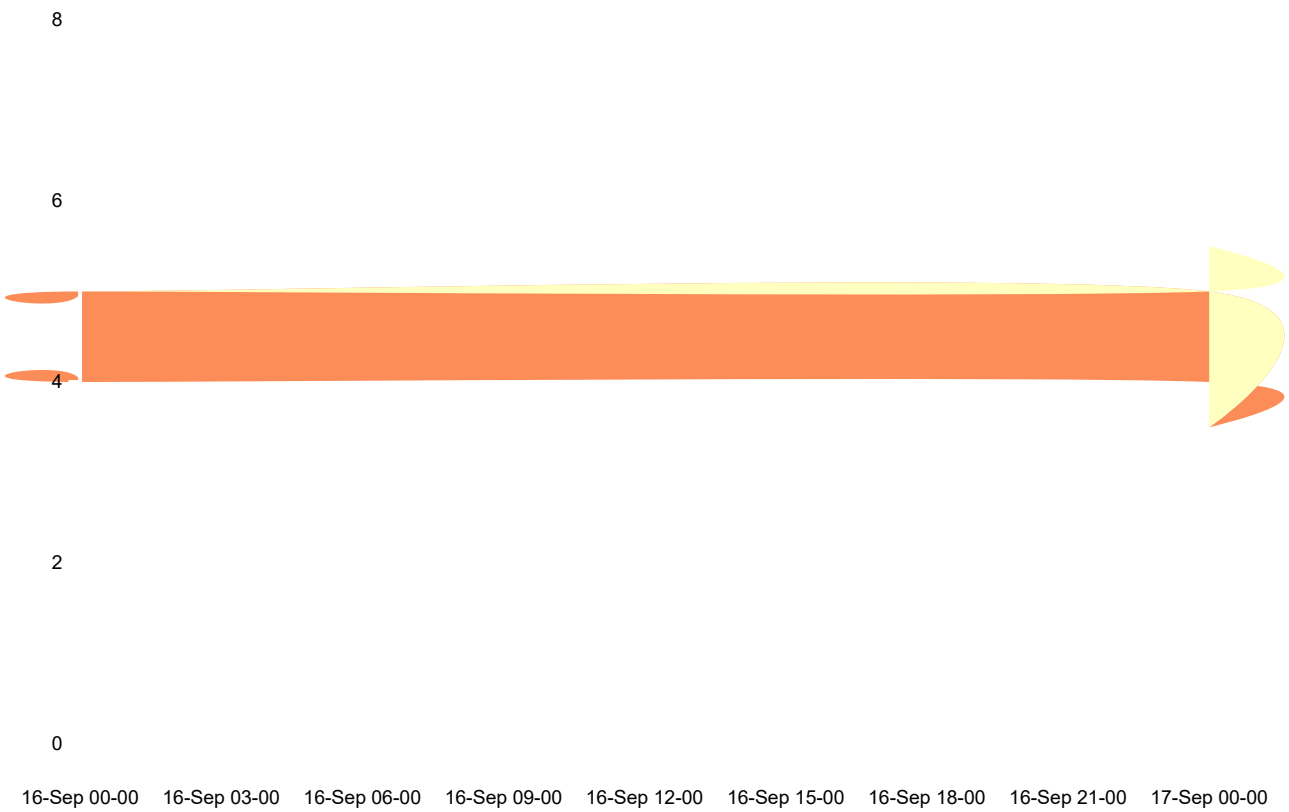


Stream Graph for sentiment by date

```
Data<-data.frame(sentiments$polarity)
colnames(Data)[1] <- "polarity"
Data$Date <- TweetDF$created
Data$text <- NULL
Data$Count <- 1
graphdata <- aggregate(Count ~ polarity + as.character(Date),data=Data,FUN=sum)
colnames(graphdata)[2] <- "Date"
str(graphdata)
```

```
## 'data.frame':    520 obs. of  3 variables:
##  $ polarity: Factor w/ 2 levels "negative","positive": 2 1 2 2 2 2 1 1 1 1 ...
##  $ Date    : chr  "2017-09-16 05:46:32" "2017-09-16 05:46:33" "2017-09-16 05:46:33" "2
017-09-16 05:46:34" ...
##  $ Count   : num  1 2 1 1 2 2 1 1 1 2 ...
```

```
graphdata %>%
streamgraph(polarity, Count, Date) %>%
  sg_axis_x(20) %>%
  sg_axis_x(1, "Date","%d-%b %H-%M") %>%
  sg_legend(show=TRUE, label="Polarity: ")
```



Overall 67% we have negative sentiment about bit coins.