Untitled

Ranjith G

6 May 2019

library(rjson)

## Warning: package 'rjson' was built under R version 3.5.2

library(bit64)

## Warning: package 'bit64' was built under R version 3.5.2

## Loading required package: bit

## Warning: package 'bit' was built under R version 3.5.2

## Attaching package bit

## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)

## creators: bit bitwhich

## coercion: as.logical as.integer as.bit as.bitwhich which

## operator: ! & | xor != ==

## querying: print length any all min max range sum summary

## bit access: length<- [ [<- [[ [[<-

## for more help type ?bit

##   
## Attaching package: 'bit'

## The following object is masked from 'package:base':  
##   
## xor

## Attaching package bit64

## package:bit64 (c) 2011-2012 Jens Oehlschlaegel

## creators: integer64 seq :

## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bin

## logical operator: ! & | xor != == < <= >= >

## arithmetic operator: + - \* / %/% %% ^

## math: sign abs sqrt log log2 log10

## math: floor ceiling trunc round

## querying: is.integer64 is.vector [is.atomic} [length] format print str

## values: is.na is.nan is.finite is.infinite

## aggregation: any all min max range sum prod

## cumulation: diff cummin cummax cumsum cumprod

## access: length<- [ [<- [[ [[<-

## combine: c rep cbind rbind as.data.frame

## WARNING don't use as subscripts

## WARNING semantics differ from integer

## for more help type ?bit64

##   
## Attaching package: 'bit64'

## The following object is masked from 'package:bit':  
##   
## still.identical

## The following objects are masked from 'package:base':  
##   
## %in%, :, is.double, match, order, rank

library(httr)

## Warning: package 'httr' was built under R version 3.5.3

library(devtools)

## Warning: package 'devtools' was built under R version 3.5.3

## Warning: package 'usethis' was built under R version 3.5.3

library(twitteR)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ----------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.8  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.2

## Warning: package 'tidyr' was built under R version 3.5.2

## Warning: package 'readr' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'dplyr' was built under R version 3.5.2

## Warning: package 'forcats' was built under R version 3.5.2

## -- Conflicts -------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::id() masks twitteR::id()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::location() masks twitteR::location()

library(tm)

## Warning: package 'tm' was built under R version 3.5.3

## Loading required package: NLP

## Warning: package 'NLP' was built under R version 3.5.2

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

## The following object is masked from 'package:httr':  
##   
## content

library(lexicon)

## Warning: package 'lexicon' was built under R version 3.5.3

library(syuzhet)

## Warning: package 'syuzhet' was built under R version 3.5.3

library(lubridate)

## Warning: package 'lubridate' was built under R version 3.5.3

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

library(ggplot2)  
library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:syuzhet':  
##   
## rescale

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(dplyr)  
library(wordcloud)

## Warning: package 'wordcloud' was built under R version 3.5.3

## Loading required package: RColorBrewer

api\_key<- "dQ68PceTqWtzBtx1lmLZdRIrh"  
api\_secret<- "3qowi6iEPuy3YOGTt0618KO8iZFGoJIol2a4flGHWa1Tbqs74S"  
access\_token<- "1274329567-iP09cKiCTcvC4hzS8S6NvInghop8lHOKbN5VVBj"  
access\_token\_secret<- "tnObWgn0R72vObdU0Ai6SXP4avviGjDWXKr8gQ0uT9lxj"  
setup\_twitter\_oauth(api\_key,  
 api\_secret,  
 access\_token,  
 access\_token\_secret)

## [1] "Using direct authentication"

rm(api\_key)  
rm(api\_secret)  
rm(access\_token)  
rm(access\_token\_secret)

ev= searchTwitter("ev",since='2016-01-01',n=2000, lang="en")  
length(ev)

## [1] 2000

evexp = searchTwitter("electric vehicle tata",since='2016-01-01',n=2000, lang="en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 2000 tweets were requested but the  
## API can only return 74

evtata= searchTwitter("electric vehicle Tata",since='2016-01-01', n=2000, lang="en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 2000 tweets were requested but the  
## API can only return 74

length(evtata)

## [1] 74

evhero = searchTwitter("electric vehicle hero",since='2016-01-01', n=2000, lang="en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 2000 tweets were requested but the  
## API can only return 1

evmah<-searchTwitter("electric vehicle mahindra",since='2016-01-01', n=2000, lang="en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 2000 tweets were requested but the  
## API can only return 17

evi= searchTwitter("India ev ",since='2016-01-01',n=2000, lang="en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 2000 tweets were requested but the  
## API can only return 676

tatatw<-sapply(evtata, function(x) x$getText())  
herotw<-sapply(evhero, function(x) x$getText())  
evtw<-sapply(ev, function(x) x$getText())  
mahtw<-sapply(evmah, function(x) x$getText())  
evitw<-sapply(evi, function(x) x$getText())  
  
  
catch.error = function(x)  
{  
 y = NA  
 catch\_error = tryCatch(tolower(x), error=function(e) e)  
 if (!inherits(catch\_error, "error"))  
 y = tolower(x)  
 return(y)  
}  
  
cleanTweets<- function(tweet){  
 # Clean the tweet for sentiment analysis  
 # remove html links, which are not required for sentiment analysis  
 tweet = gsub("(f|ht)(tp)(s?)(://)(.\*)[.|/](.\*)", " ", tweet)  
 # First we will remove retweet entities from the stored tweets (text)  
 tweet = gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", " ", tweet)  
 # Then remove all "#Hashtag"  
 tweet = gsub("#\\w+", " ", tweet)  
 # Then remove all "@people"  
 tweet = gsub("@\\w+", " ", tweet)  
 # Then remove all the punctuation  
 tweet = gsub("[[:punct:]]", " ", tweet)  
 # Then remove numbers, we need only text for analytics  
 tweet = gsub("[[:digit:]]", " ", tweet)  
 # finally, we remove unnecessary spaces (white spaces, tabs etc)  
 tweet = gsub("[ \t]{2,}", " ", tweet)  
 tweet = gsub("^\\s+|\\s+$", "", tweet)  
 # if anything else, you feel, should be removed, you can. For example "slang words" etc using the above function and methods.  
 # Next we'll convert all the word in lower case. This makes uniform pattern.  
 tweet = catch.error(tweet)  
 tweet  
}

cleanTweetsAndRemoveNAs<- function(Tweets) {  
 TweetsCleaned = sapply(Tweets, cleanTweets)  
 # Remove the "NA" tweets from this tweet list  
 TweetsCleaned = TweetsCleaned[!is.na(TweetsCleaned)]  
 names(TweetsCleaned) = NULL  
 # Remove the repetitive tweets from this tweet list  
 TweetsCleaned = unique(TweetsCleaned)  
 TweetsCleaned  
}

tatatwCleaned = cleanTweetsAndRemoveNAs(tatatw)  
herotwCleaned = cleanTweetsAndRemoveNAs(herotw)  
mahtwCleaned <- cleanTweetsAndRemoveNAs(mahtw)  
Indiatwcleaned= cleanTweetsAndRemoveNAs(evitw)  
evitwcleaned= cleanTweetsAndRemoveNAs(evtw)  
length(evtw)

## [1] 2000

tata1<-Corpus(VectorSource(tatatwCleaned))  
corpusevi <- iconv(evitwcleaned, to = "utf-8")  
  
corpusevi1<-Corpus(VectorSource(evitwcleaned))  
evi1<-Corpus(VectorSource(corpusevi1))  
evicorpus<- tm\_map(evi1, tolower)

## Warning in tm\_map.SimpleCorpus(evi1, tolower): transformation drops  
## documents

mah1<-Corpus(VectorSource(mahtwCleaned))  
  
  
  
corpusIndia<-Corpus(VectorSource(Indiatwcleaned))  
India1<-Corpus(VectorSource(corpusIndia))  
Indiacorpus<- tm\_map(India1, tolower)

## Warning in tm\_map.SimpleCorpus(India1, tolower): transformation drops  
## documents

#to remove stop words  
evi <- tm\_map(evi1, removeWords, stopwords('english'))

## Warning in tm\_map.SimpleCorpus(evi1, removeWords, stopwords("english")):  
## transformation drops documents

eviplot <- tm\_map(evicorpus, removeWords, stopwords('english'))

## Warning in tm\_map.SimpleCorpus(evicorpus, removeWords,  
## stopwords("english")): transformation drops documents

mah <- tm\_map(mah1, removeWords, stopwords('english'))

## Warning in tm\_map.SimpleCorpus(mah1, removeWords, stopwords("english")):  
## transformation drops documents

tata <- tm\_map(tata1, removeWords, stopwords('english'))

## Warning in tm\_map.SimpleCorpus(tata1, removeWords, stopwords("english")):  
## transformation drops documents

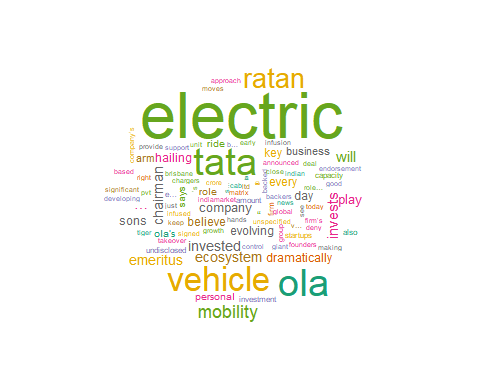
India <- tm\_map(India1, removeWords, stopwords('english'))

## Warning in tm\_map.SimpleCorpus(India1, removeWords, stopwords("english")):  
## transformation drops documents

# 

# TATA WORDCLOUD

tatacloud<-wordcloud(tata,min.freq = 2,  
 colors=brewer.pal(8, "Dark2"),  
 scale = c(4,0.3),  
 random.color = TRUE,max.words = 500)



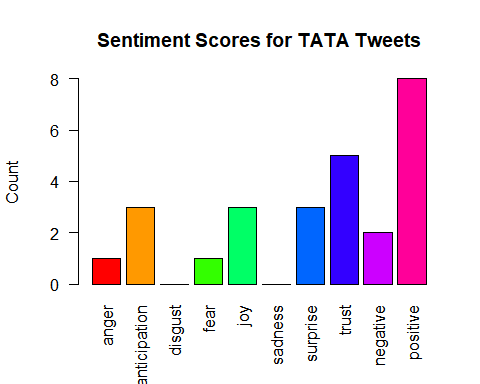
# 

# TATA BARPLOT

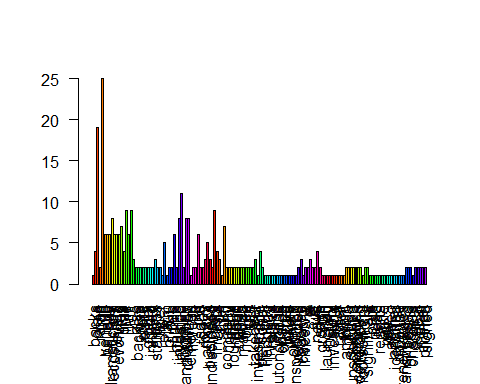
tatastring=toString(tata)  
  
sentitata <- get\_nrc\_sentiment(tatastring)

## Warning: package 'bindrcpp' was built under R version 3.5.3

barplot(colSums(sentitata),  
 las = 2,  
 col = rainbow(10),  
 ylab = 'Count',  
 main = 'Sentiment Scores for TATA Tweets')

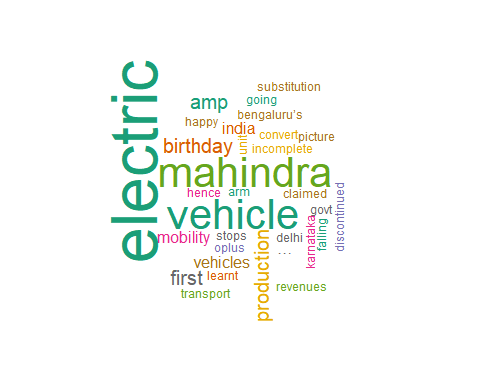
 #words Repetative in tata tweets

wordtata<-TermDocumentMatrix(tata)  
wordtata<- as.matrix(wordtata)  
# Bar plot  
w <- rowSums(wordtata)  
w <- subset(w, w<=25)  
barplot(w,  
 las = 2,  
 col = rainbow(50))



# MAHINDERA wORD CLOUD

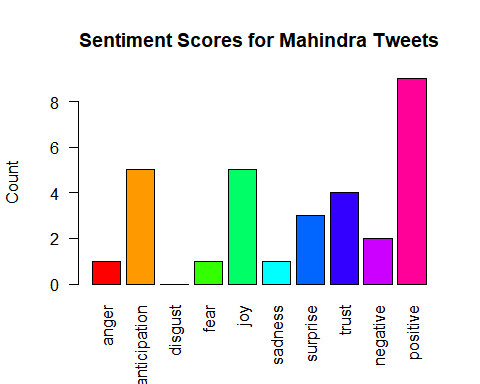
mahcloud<-wordcloud(mah,min.freq = 2,  
 colors=brewer.pal(8, "Dark2"),  
 scale = c(4,0.3),  
 random.color = TRUE,max.words = 500)



# 

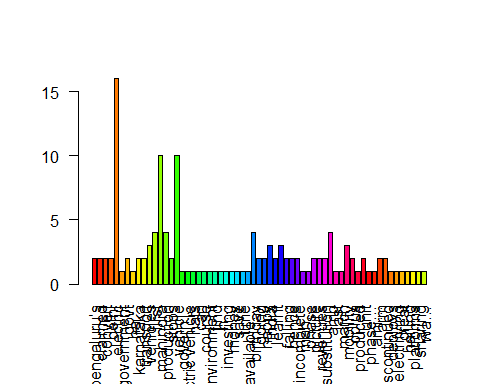
# MAHINDERA BAR PLOT

mahstring=toString(mah)  
  
sentimah <- get\_nrc\_sentiment(mahstring)  
  
  
barplot(colSums(sentimah),  
 las = 2,  
 col = rainbow(10),  
 ylab = 'Count',  
 main = 'Sentiment Scores for Mahindra Tweets')



#words Repetative in Mahindra tweets

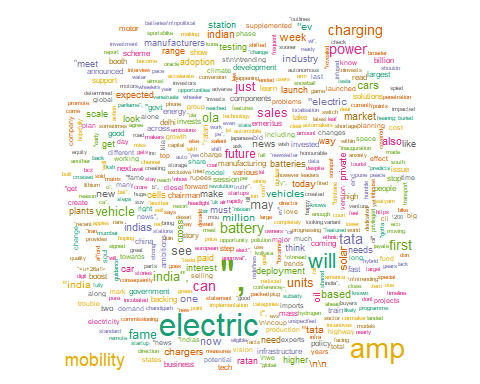
wordmah<-TermDocumentMatrix(mah)  
wordmah<- as.matrix(wordmah)  
# Bar plot  
m <- rowSums(wordmah)  
m <- subset(m, m<=25)  
barplot(m,  
 las = 2,  
 col = rainbow(50))



# INDIA TWEET WORD CLOUD

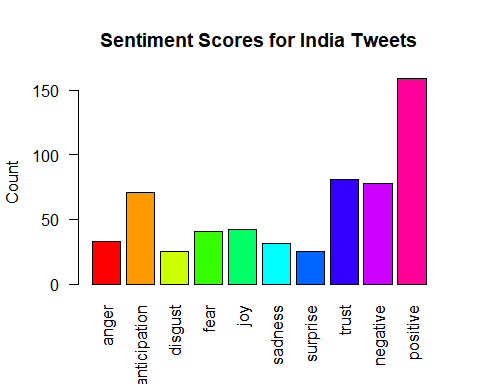
Indiacloud<-wordcloud(India,min.freq = 2,  
 colors=brewer.pal(8, "Dark2"),  
 scale = c(4,0.3),  
 random.color = TRUE,max.words = 500)

## Warning in wordcloud(India, min.freq = 2, colors = brewer.pal(8,  
## "Dark2"), : india could not be fit on page. It will not be plotted.



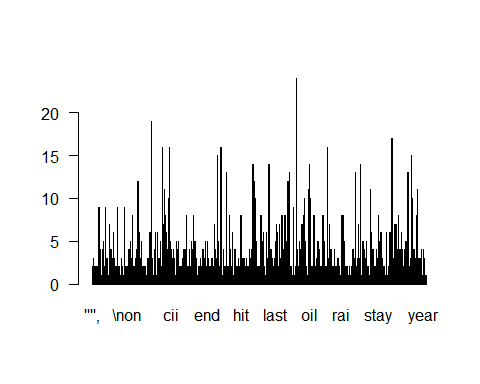
#INDIA TWEET BAR PLOT

Indiastring=toString(India)  
  
sentiIndia <- get\_nrc\_sentiment(Indiastring)  
  
  
barplot(colSums(sentiIndia),  
 las = 2,  
 col = rainbow(10),  
 ylab = 'Count',  
 main = 'Sentiment Scores for India Tweets')



#words Repetative in India tweets

wordIndia<-TermDocumentMatrix(India)  
wordIndia<- as.matrix(wordIndia)  
# Bar plot  
I <- rowSums(wordIndia)  
I <- subset(I, I<=25)  
barplot(I,  
 las = 1,  
 col = rainbow(50))



# TOTAL EV TWEET WORD CLOUD

evinew <- read.csv("file:///C:/Users/RANJIT/Documents/evfile.csv", header = T)  
evicloud<-wordcloud(evinew$text,min.freq = 2,  
 colors=brewer.pal(8, "Dark2"),  
 scale = c(4,0.3),  
 random.color = TRUE,max.words = 500)

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation):  
## transformation drops documents

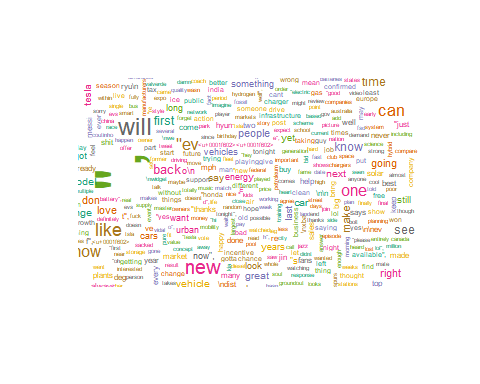
## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents

## Warning in wordcloud(evinew$text, min.freq = 2, colors = brewer.pal(8,  
## "Dark2"), : vehicle could not be fit on page. It will not be plotted.

## Warning in wordcloud(evinew$text, min.freq = 2, colors = brewer.pal(8,  
## "Dark2"), : electric could not be fit on page. It will not be plotted.



evicloud<-wordcloud(eviplot,min.freq = 2,  
 colors=brewer.pal(8, "Dark2"),  
 scale = c(4,0.3),  
 random.color = TRUE,max.words = 500)

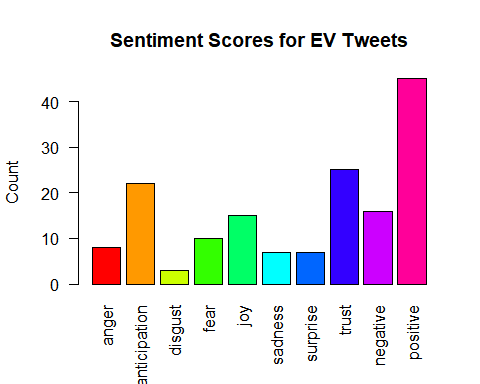


# TOTAL EV TWEET BAR PLOT

n=toString(evinew$text)  
# Obtain sentiment scores  
s <- get\_nrc\_sentiment(n)  
head(s)

## anger anticipation disgust fear joy sadness surprise trust negative  
## 1 8 22 3 10 15 7 7 25 16  
## positive  
## 1 45

# Bar plot  
barplot(colSums(s),  
 las = 2,  
 col = rainbow(10),  
 ylab = 'Count',  
 main = 'Sentiment Scores for EV Tweets')

 #words Repetative in EV Total tweets

wordevi<-TermDocumentMatrix(evi)  
wordevi<- as.matrix(wordevi)  
# Bar plot  
e <- rowSums(wordevi)  
e <- subset(e, e<=25)  
barplot(e, las = 1,  
 col = rainbow(50))

