Twitter Sentiment Analysis in R

Sentiment analysis, also referred to as Opinion Mining, implies extracting opinions, emotions and sentiments in text. The most common applications of sentiment analysis is to track attitudes and feelings on the web, especially for tacking products, services, brands or even people. The main idea is to determine whether they are viewed positively or negatively by a given audience.

First step to perform Twitter Analysis is to create a twitter application. This application will allow us to perform analysis by connecting R console to the twitter using the Twitter API.

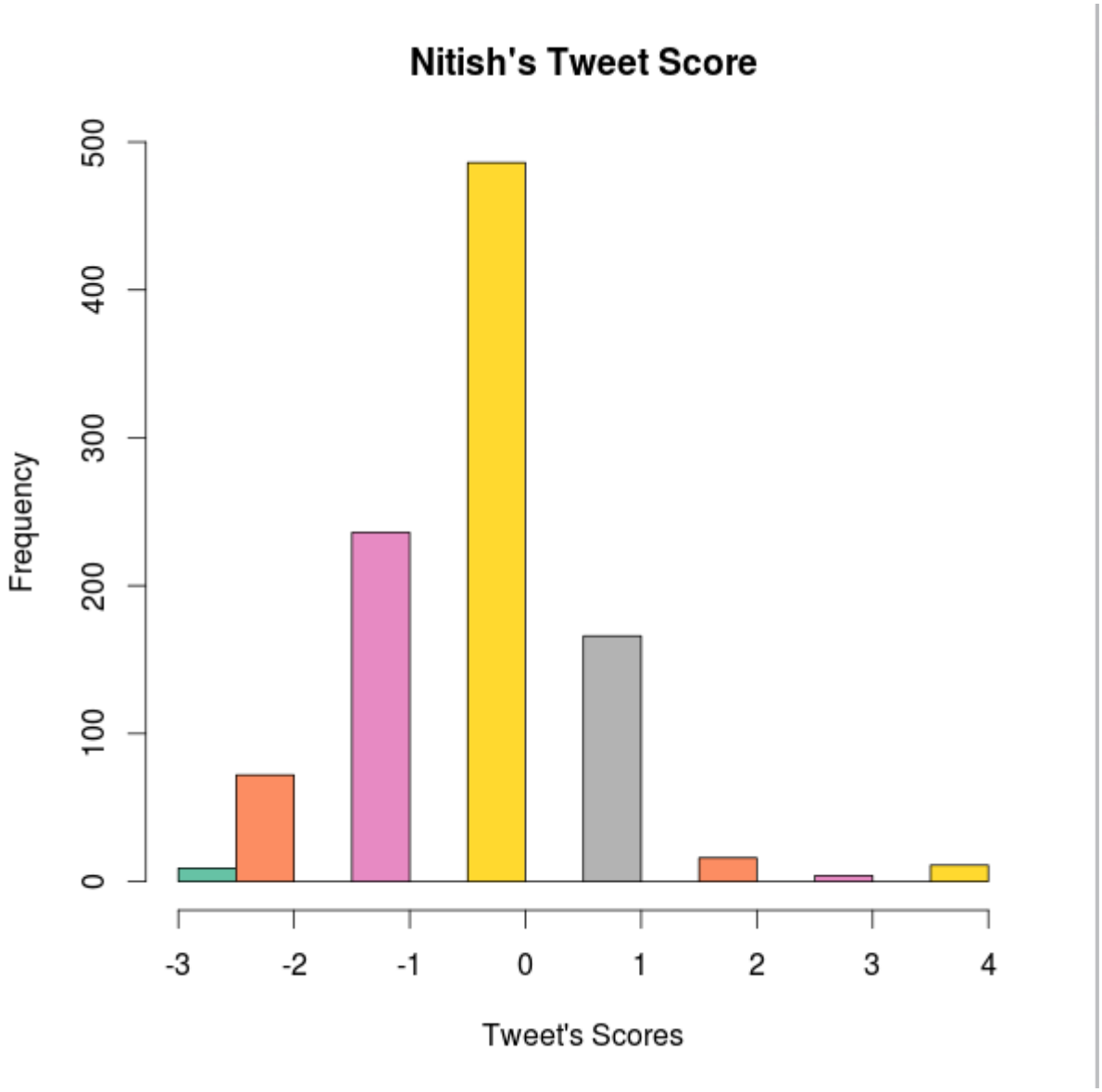
I have performed the Sentiment Analysis techniques on “#Nitish” and “#Lalu”. Bihar is going through Assembely Elections so lots of news coming for these two hashtags.

Source Code:

```
save.image("twitter.Rdata")
load("~/twitter.Rdata")
library("twitteR")

#Extracting tweets using api
consumerKey <-"ubAQyKmntRo1IbS31g8Il"
consumerSecret <- "X8wSL3mkQe5imJrUrCDocBp9pIsBqFSchLqsxupeDPHEv4"
accessToken <- "4069043834-caTSprlm70KD2FrqWFMew4HQYYJfxbscuy63"
accessSecret <- "9Uof4I3Dk46RjKcST0H8XNlSUg21FGvJwuQ35WM"
setup_twitter_oauth(consumerKey,consumerSecret,accessToken,accessSecret)

#Saving tweets
Nitish.list <- searchTwitter("#Nitish",n=1000)
Nitish.df = twListToDF(Nitish.list)
write.csv(Nitish.df,file="~/Desktop/nitish.csv",row.names = F)

#Load sentiment word lists
pos.words <- scan("/home/chiggy/twitter/positive-words.txt",what="character")
neg.words <- scan("/home/chiggy/twitter/negative-words.txt",what="character")

#Sentiment Function
library(stringr)
library(plyr)

score.sentiment <- function(sentences, pos.words, neg.words, .progress='none')
{
        require(plyr)
        require(stringr)

        #For each element of a list or vector, apply function and store result into an array
        scores = laply(sentences, function(sentence, pos.words, neg.words) {
```

```
        # clean up sentences with R's regex-driven global substitute, gsub():
        sentence = gsub('[[:punct:]]', '', sentence)          # remove punctuation
        sentence = gsub('[[:cntrl:]]', '', sentence)          # remove control characters
        sentance = gsub('(RT|via)((?:\\b\\W*@\\w+)+)', '', sentance) # remove retweet entities
        sentance = gsub('@\\w+', '', sentance)                # remove at people
        sentance = gsub('http\\w+', '', sentance)             # remove html links
        sentance = gsub('[ \t]{2,}', '', sentance)            # remove unnecessary spaces
        sentance = gsub('^\\s+|\\s+$', '', sentance)
        sentance = gsub('[[:digit:]]', '', sentance)          # remove numbers
        sentence = gsub('\\d+', '', sentence)
        # and convert to lower case:
        sentence = tolower(sentence)

        # split into words. str_split is in the stringr package
        word.list = str_split(sentence, '\\s+')

        # it simplifies list to produce a vector
        words = unlist(word.list)

        # compare our words to the dictionaries of positive & negative terms
        pos.matches = match(words, pos.words)
        neg.matches = match(words, neg.words)

        # match() returns the position of the matched term or NA
        # we just want a TRUE/FALSE:
        pos.matches = !is.na(pos.matches)
        neg.matches = !is.na(neg.matches)

        # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
        score = sum(pos.matches) - sum(neg.matches)

        return(score)
    }, pos.words, neg.words, .progress=.progress )

    scores.df = data.frame(score=scores, text=sentences)
    return(scores.df)
}

# Score all tweets
Nitish.score <- score.sentiment(Nitish.df$text,pos.words,neg.words,.progress = 'text')
write.csv(Nitish.score."/home/chiggy/twitter/NitishScore.csv")

#Plot histogram
library(RColorBrewer)
hist(Nitish.score$score,xlab="Tweet's Scores",main="Nitish's Tweet Score",col=brewer.pal(9,"Set2"))
```

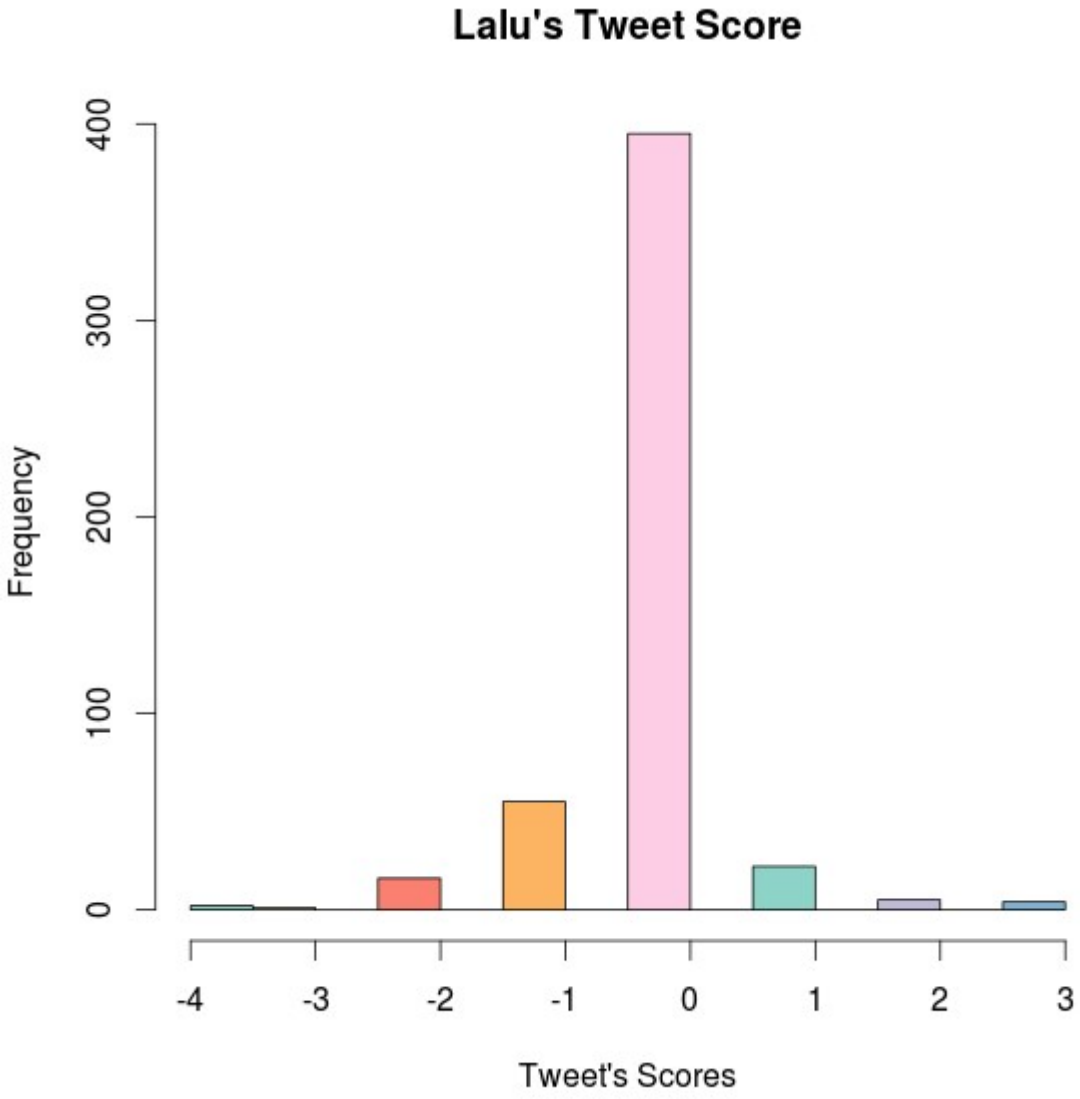
Visualizing the tweets



I created visual histograms and other plots to visualize the sentiments of the user. This can be done by using hist function. I have used a package RColorBrewer to play with colors.

The above histogram shows the frequency of tweets with respect of scores allotted to each tweets. The x-axis shows the score of each tweet as a negative and positive integer or zero. A positive score represents positive or good sentiments associated with that particular tweet whereas a negative score represents negative or bad sentiments associated with that tweet. A score of zero indicates a neutral sentiment. The more positive the score, the more positive the sentiments of the person tweeting and vice-versa. The above histogram is skewed towards negative score which shows that the sentiments of people regarding Mr. Nitish are negative.

Similarly, I have plotted histogram for Mr. Lalu. It is also negatively skewed.



Future Plans:

Above histogram has more frequency of neutral tweets. I can explain using this example tweet.

“Nitish kumar was an excellent CM but criticised for resignation after 2014 Lok sabha Election”

This string has 1 positive word “excellent” and 1 negative word “criticised” but overall string seems to be positive sentiment. It analyzed as a neutral. I'll use some supervised learninng and try to train my model so that it will evaluate above tweet as a positive sentiment.

I can use logistic regression and do labelling. First i will combine positive and negative words and if these words are occuring in any tweet i'll assign 1 to this words and 0 to other words which are not occuring.
By using several training examples I can train my system If it is positive sentiment output will be 1 and if it is negative entiment output will be 0. For this classification logistic regression will be the best algorithm.