Chase_Twitter

Anurag Garg

November 15, 2015

1. Getting Data

To get data for this project, we need to authorize first and then we can access twitter API. Following script will do it and the output will be a data frame of 3200 tweets for @Chase.

```
#install_github("geoffjentry/twitteR")

# Initialization of required packages
library(RCurl)
library(RJSONIO)
library(digest)
library(devtools)
library(twitteR)

# Download the certificate needed for authentication
 if (!file.exists('cacert.perm')){
   download.file(url = 'http://curl.haxx.se/ca/cacert.pem',
                 destfile='cacert.perm')
 }

# Setup twitter keys and secrets

# These can be accessed from your application's page on dev.twitter.com
# under the names Consumer key, Consumer secret, Access token, and
# Access token secret respectively.

consumerKey = '8jqqM7OlEk6FkdncOaGLjUtXM'
consumerSecret = '8U61tPinhxbGRYJnCk257RUPNTOumxybwNOHVqrQwxuH3sfmEk'
accessToken = '2198592541-R6BnXREHUn6SeMVI9w1fZPof4jpVnr41p1RNJ92'
accessTokenSecret = 'aRp5f97FdrT9dXYTMZDtzw6CQPfHaLlHYNg3r79jOzkPs'
setup_twitter_oauth(consumerKey, consumerSecret,
                    accessToken, accessTokenSecret)

#get the tweets for @Chase.

tweets<-searchTwitter("@Chase", n=3200)
#length(tweets)

# Transform tweets list into a data frame
tweets.df = do.call("rbind",lapply(tweets,as.data.frame))

# Save data into an R file for using it later.
saveRDS(tweets.df,file="tweets_data.Rda")
```

Note:

For most of the tweets (>99%), location (latitude and longitude) is disabled by the user and hence we are not doing any analysis related to location.

2. Cleaning Data

Now we will perform the task of cleaning data. Data need to be cleaned because:

a. We will remove the columns which are not needed. We will retain following 5 columns for our analysis:

- text
- created
- id
- screenName
- retweetCount

b. There are duplicate tweets in the data.

c. We do not need tweets which are tweeted by Chase itself (tweets by the official handle of Chase.)

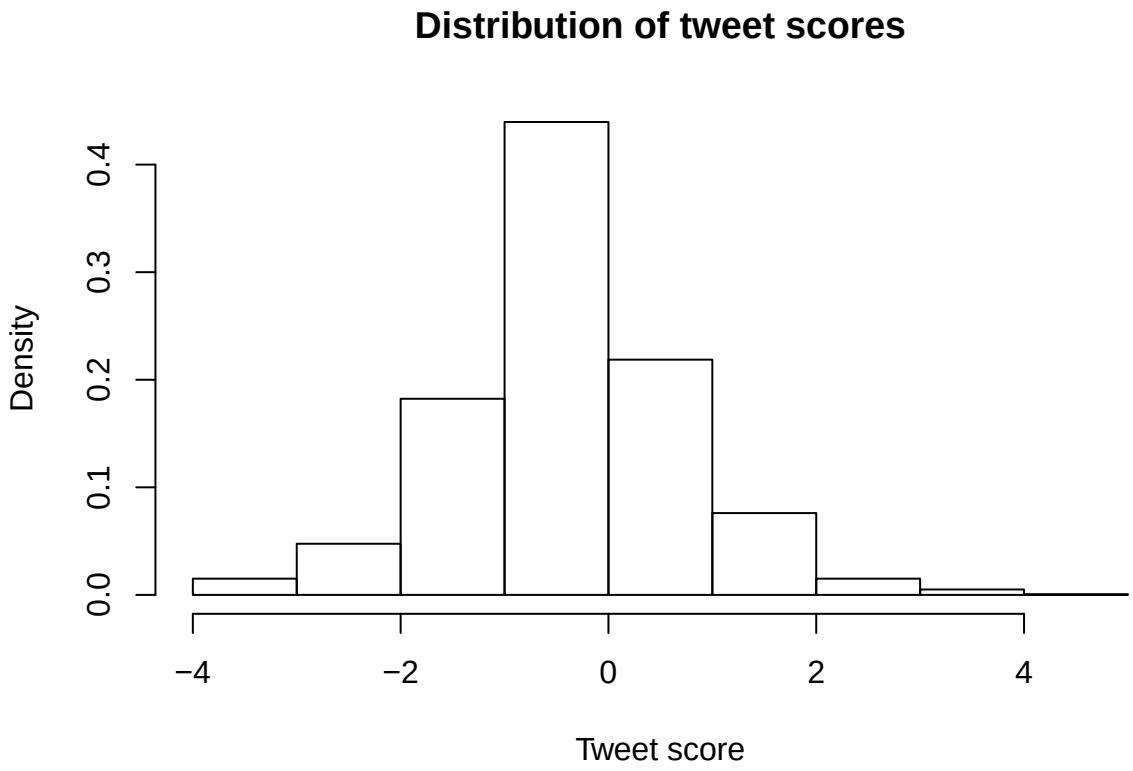
```
library("plyr")

# Load the tweets data.

tweets.df<-readRDS(file="tweets_data.Rda")

#clean data text

#1. removing column which are not needed
col_needed<-c("text","created","id","screenName","retweetCount")
needed_col_df<-tweets.df[,col_needed]


#2. removing rows which are duplicate tweets
needed_data_df<- subset(needed_col_df, !duplicated(needed_col_df[,1]))

#3. removing rows where user is chase
data_df<- subset(needed_data_df,!(needed_data_df$screenName=="Chase"))
```

3. Exploratory Data Analysis

a. General sentiment about @Chase.

```
source("ts_score_sentiment.R")
library("plyr")
library(stringr)

#Getting a score for each tweet using sentiment analysis

# Getting lexicon of positive and negative words.

pos <- scan('positive-words.txt', what='character', comment.char=';')
neg <- scan('negative-words.txt', what='character', comment.char=';')

#Giving a score to a tweet based on net (positive-negative) words.
```

```
analysis <- ts_score_sentiment(data_df$text, pos, neg)
data_df<-merge(data_df,analysis)

#Multiplying the score with the retweet count to get the effective score.
#add 1 to retweet count to get effective number of times the tweet is posted.

data_df$effScore<-(data_df$score*(data_df$retweetCount+1))

#Plotting a histogram to see score distribution for chase at this point.
hist(data_df$score, probability = TRUE,xlab = "Tweet score",
     main = "Distribution of tweet scores")
```



We can see that density distribution of the tweet scores is nearly normal and also the histogram is little misleading. Although maximum tweets are in range of 0 to -1 which shows a little negative outlook but from the normal distribution we can see that the mean score is 0 which means a neutral outlook.

b. Sentiment about individual products

We will now categorize the tweets according to different products offered by Chase. Major products which we considered are:

i. Sapphire Credit Card

ii. Freedom Credit Card

iii. Slate Credit Card

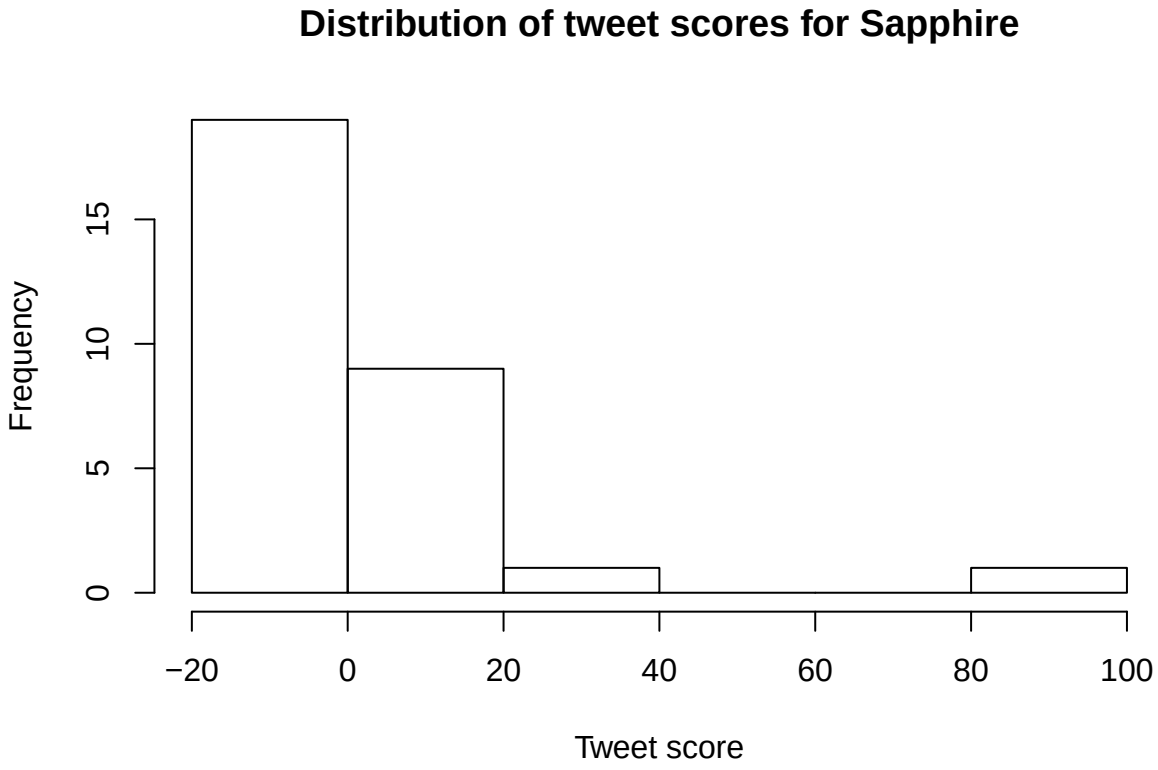
iv. Credit Card(Misc tweets about credit cards)

v. Checking account

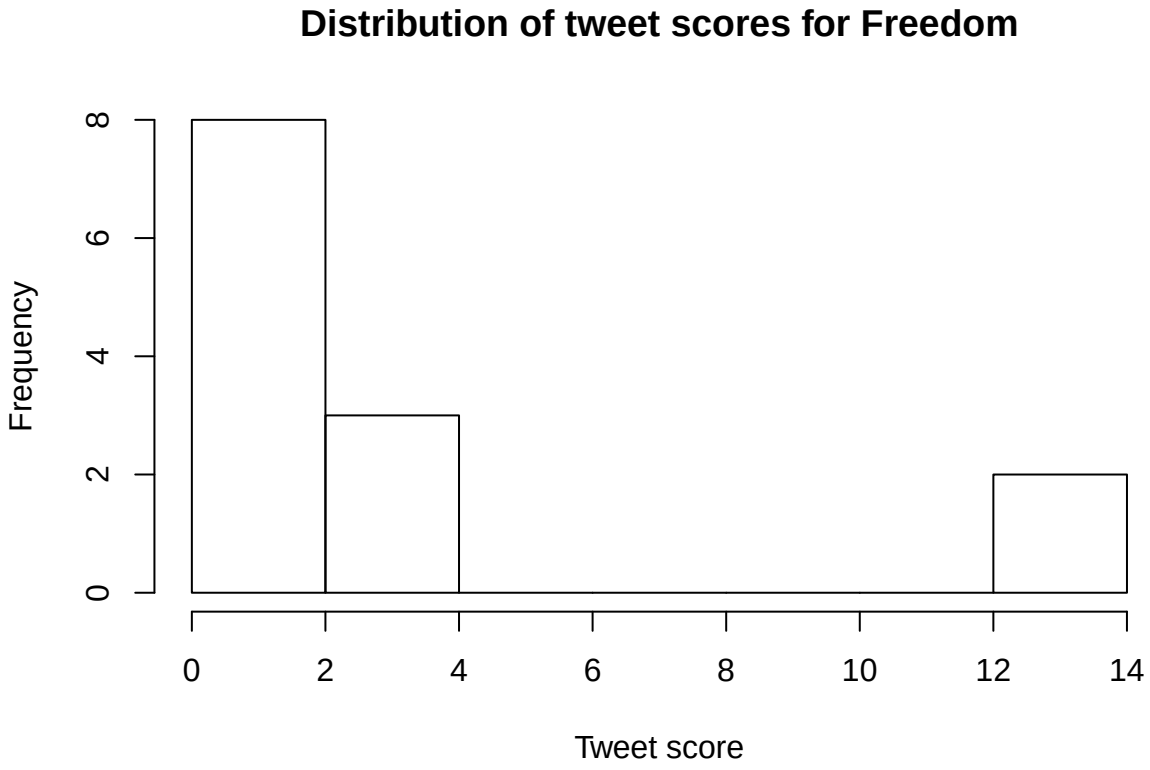
vi. Debit card

```
#categorize data according to product

#Sapphire
data_sapphire<- subset(data_df,grepl("sapphire",data_df$text,
                                                    ignore.case = T))
data_sapphire$product<-rep("Sapphire",nrow(data_sapphire))

#Freedom
data_freedom<- subset(data_df,grepl("freedom",data_df$text,
                                                    ignore.case = T))
data_freedom$product<-rep("Freedom",nrow(data_freedom))

#Slate
data_slate<- subset(data_df,grepl("slate",data_df$text,
                                                    ignore.case = T))
data_slate$product<-rep("Slate",nrow(data_slate))

#Misc Credit card
data_allcredit<-rbind.data.frame(data_slate, data_freedom,data_sapphire)
data_credit<- subset(data_df,!(data_df$id %in% data_allcredit$id))
credit_match<-c("credit card","creditcard")
data_credit<- subset(data_credit,grepl(paste(credit_match,collapse="|"),data_credit$text,
                                                  ignore.case = T))
data_credit$product<-rep("Misc_Credit",nrow(data_credit))

#Checking Account
checking_match<-c("checking account","checkingaccount","checking")
data_checkingacc<- subset(data_df,grepl(paste(checking_match,collapse="|"),data_df$text,
                                                  ignore.case = T))
data_checkingacc$product<-rep("Checking",nrow(data_checkingacc))

#Debit card
debit_match<-c("debit card","debitcard")
data_debit<- subset(data_df,grepl(paste(debit_match,collapse="|"),data_df$text,
                                                  ignore.case = T))
data_debit$product<-rep("Debit",nrow(data_debit))
```

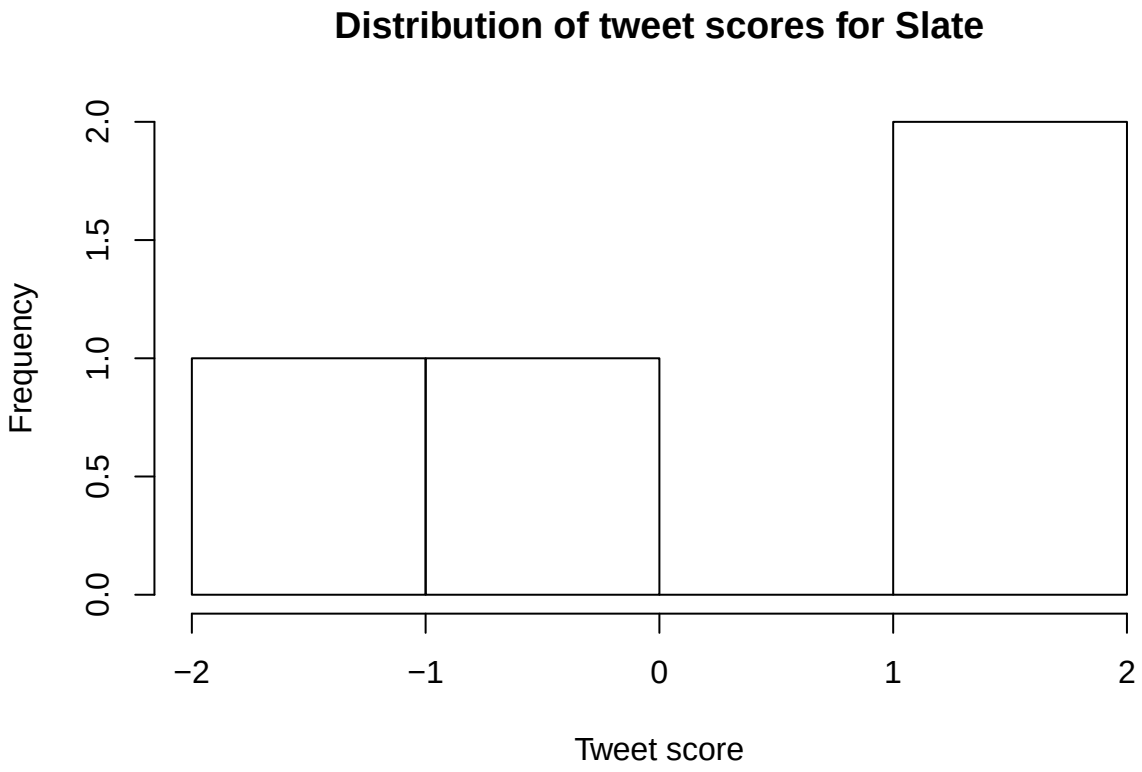
i. Sentiment analysis of Sapphire credit card.



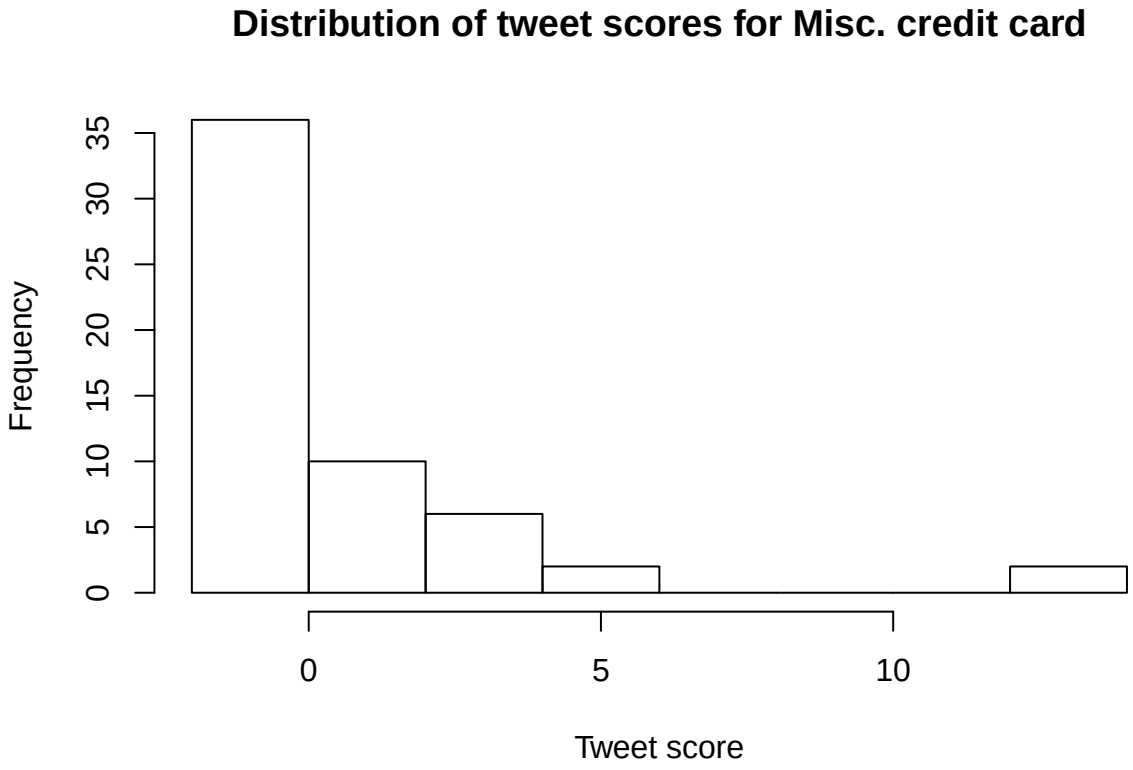
ii. Sentiment analysis of Freedom credit card.



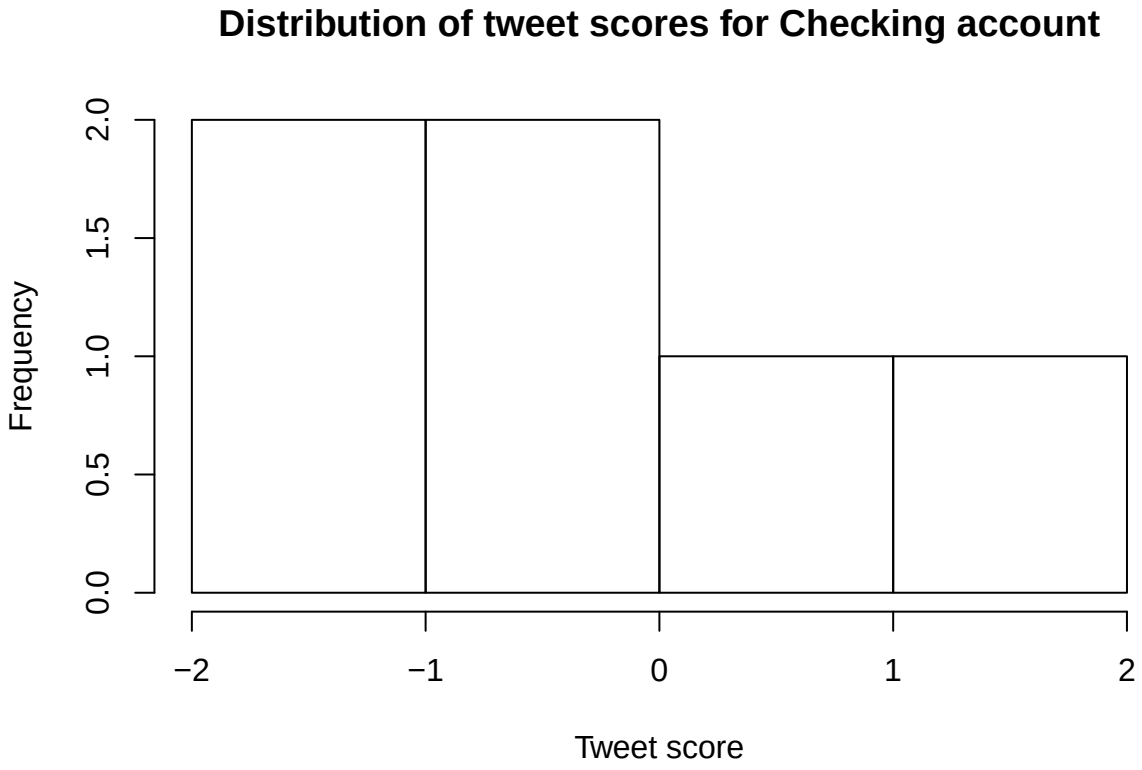
iii. Sentiment analysis of Slate credit card.



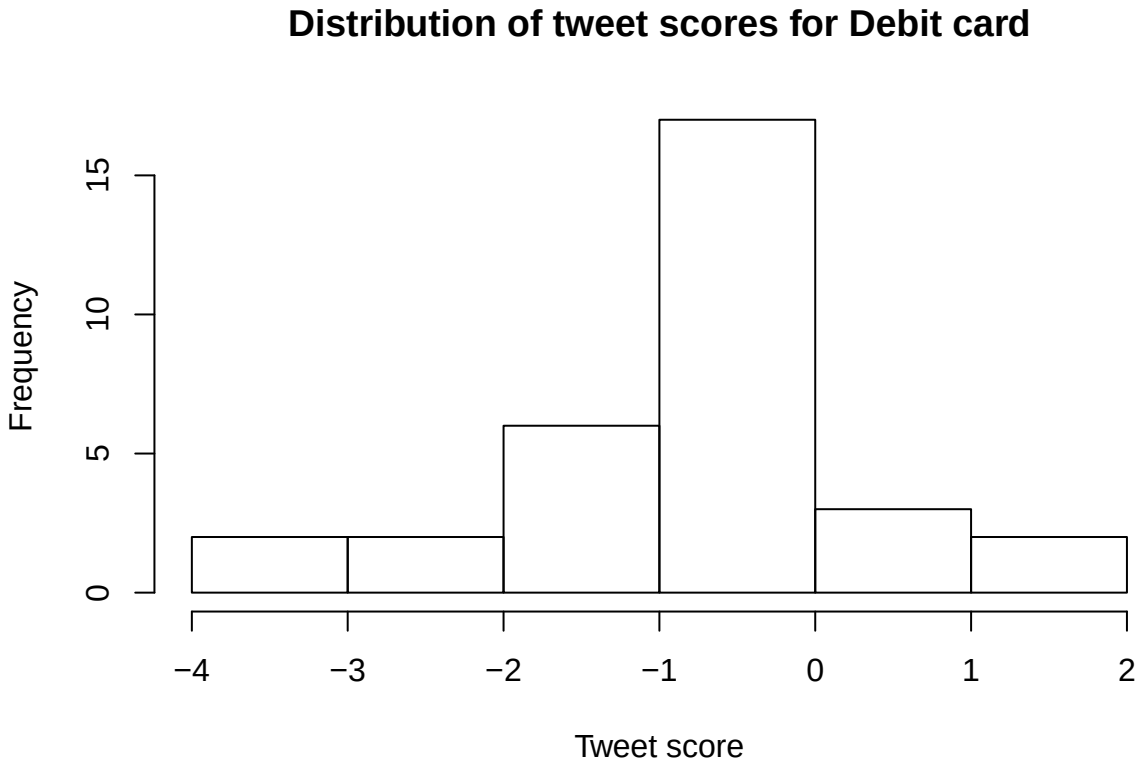
iv. Sentiment analysis of Misc. tweets about credit card.



v. Sentiment analysis of Checking account.



vi. Sentiment analysis of Debit card.



We can see from histograms that products with positive outlook are:

- Sapphire
- Freedom

We can see from histograms that products with negative outlook are:

-

We can see from histograms that products with neutral outlook are:

- Slate

Note:

For **Slate credit card** and **Checking account**, we need to consider student's t-distribution as the number of observations is pretty low.

4. Graph Analysis

We can create a bi-partite graph of users and products. Edges weight can be the score of the tweet which user posted about a specific product.