**Name: Kavya Janga**

This document contains both Project report and code.

**Project: Twitter Sentiment Analysis for product review**

The thought of this extend is to gather client assumption among two exceptionally well known need items of our day to day lives and compare their client fulfillment level. The products I have chosen for this study is iPhone-11, the newest version of iPhone which is very popular in United States. However, some recent research shows that iPhone popularity is declining day by day. The main reasons identified for this declining sale is the brand’s high price range and not being able to meet user satisfaction who are paying the price.

The usability and feature range of Android phones are improving day by day. There are many products in the market that offer the same features as iPhone but with a much lower price. Thus, user expectations about the product quality and service are also lower for these products. The other product that will be compared in this project along with iPhone-11 is Samsung Galaxy S10, which is considered as having equal features, quality and service range as iPhone 11 but with a much lower price.

The target of this investigation is to gather client comments and reviews directly from Twitter, Amazon destinations, which are famous among individuals to communicate their feeling about different items. This project will compare what are some of the positive and negative sides of these 2 products from user sentiment.

The business problem this project aims to identify is such: what is the user sentiment who is using these two products in their daily lives. Also, what are some of the meanings behind the reviews and what impact those latent meanings might have in the future of these products. From this analysis, both the brands producing these phones can learn the most important aspects that users love and hate about the phones. Thus this project can help the companies improve their products as well as sale and user satisfaction by gaining knowledge from the user sentiment available in the internet.

**Data Source**

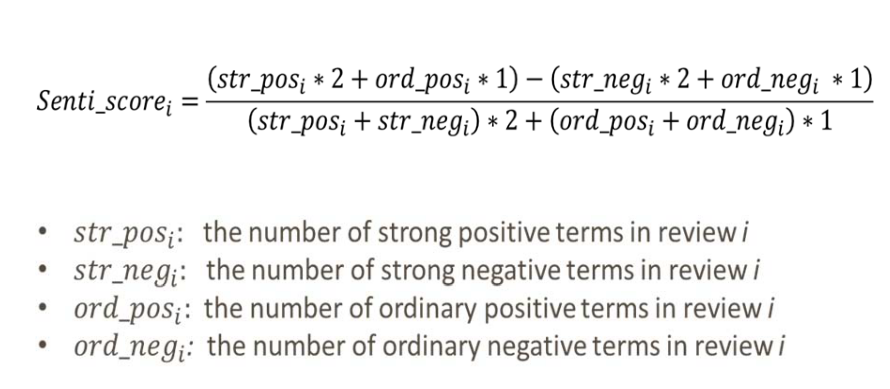
The data for this project has been collected from various sources. The reviews of iPhone-11 have been collected from Twitter API. Twitter offers an API by which anyone can collect tweets with API keys. The collection of the tweets for iPhone-11 was filtered by the string “iPhone 11” and only the tweets written in English were collected. The tweets were collected in the file “iphone11.txt” for a period of 10-12 hours and the initial file size of tweets was 10MB.

The next dataset was collected about Samsung Galaxy S10 phones through the Twitter API. After a span of about 24 hours the number of tweets collected in “samsungGalaxys10.txt” file was very small with a size of only 2MB. Thus, more reviews of users were collected by web scraping. User reviews from Amazon and twitter API. All the tweets and reviews were then combined into a csv file (S10tweets.csv) for further analysis with sentiment analysis.

Sentiment analysis will be done on the user reviews to understand public sentiment on these products as well as their intuition behind the reviews.

### Sentiment Analysis:

For sentiment analysis, the most frequent words are identified first to understand what are the popular topics the users are reviewing about these phones? Next with positive and negative word dictionaries, the words from the text files are matched to find the positive, neutral and negative words. The words are identified with the sentiment score formula, which is:

[](https://user-images.githubusercontent.com/5343403/42731323-479c5b9c-87d0-11e8-9112-35f34811e1a1.PNG)

If the score is greater than 0, then the word is identified as positive, if the score is equal to 0 then neutral and negative otherwise. After identifying the positive, negative and neutral words, the intuition behind each group of words is analyzed based on popular ideas about the products.

# Analysis:

The analysis on both the phones based on user reviews is described below:

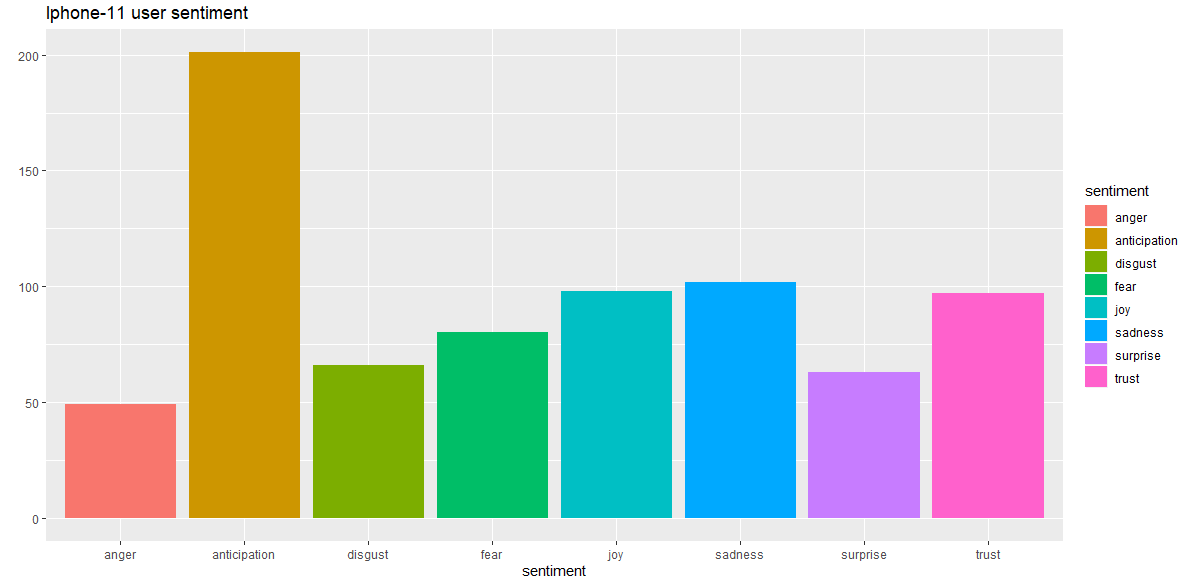
## iPhone 11:

### Data pre-processing:

After saving the tweets in a csv file (S10tweets.txt). First the texts were saved in a corpus and then all the alphabets were converted to lower case. Next all the numbers, punctuations, spaces were removed from the text. Then the English stop-words along with self-defined stop-words were removed from the tweets. In the case of iPhone, the self-defined stop-words decided are "iPhone", "apple", "plus", “phone”, because these words are likely to be appeared in the tweets most. Next stemming is done on the words.

After the above-mentioned processes, the corpus is converted into a term-document matrix which computes the term frequencies of each word in each corpus document. Next the term-document matrix is written in a csv file named “iphone11.csv”.

A qplot is plotted initially to know customer feelings about the product in the reviews. From the plot it shows that users reviews indicate different user opinions about the product.



Next the frequency of each word out of whole corpus is calculated and the top 50 most frequent words are visualized in a word-cloud plot.

[](https://user-images.githubusercontent.com/5343403/42731397-11663f1e-87d2-11e8-8415-35badef8bce9.png)

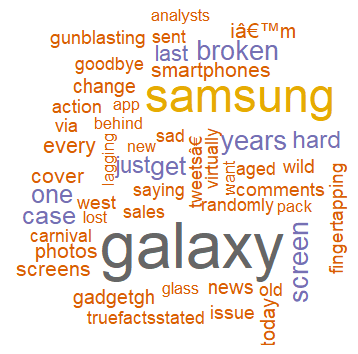
From the above plot we can see that the top words for iPhone-11 includes words like Samsung and Galaxy, indicating that the competitive Samsung product is mentioned several times in the tweets. Also, we can see the words amp, charger, protector, screen, power which indicate the tweets are mostly about charging and screen protectors for the phone.

Next the positive and negative words are identified from the tweets using the positive word and negative word dictionary. The words that match with the positive word dictionary are saved in a csv file. This process is done by generating a score for each word defining if the word is positive or not. Likewise, the negative words are identified by the same process and saved for further analysis. Also, the words that don’t fall in any of the categories are identified as neutral words.

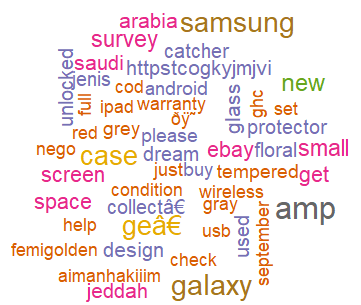
Next the positive, negative and neutral words are visualized to understand the sentiments of the reviewers. First, we will analyze the meaning behind the positive words:

[](https://user-images.githubusercontent.com/5343403/42731405-417ca2b0-87d2-11e8-81b9-bf953c85ba00.png)

From the top 50 positive words identified from the positive-word dictionary we can see that the words “win”, ”giveaway”, ”deal/deals”, ”freebies”, “free” are mostly talked about in the reviewers. These words indicate that the positive tweets were about various deals offered by different sites for the phones. This can be a strong indication that the price range of the phone is so high that most of the positive/happy tweets are from users who got a good deal or interested to free giveaways happening on the iPhones.

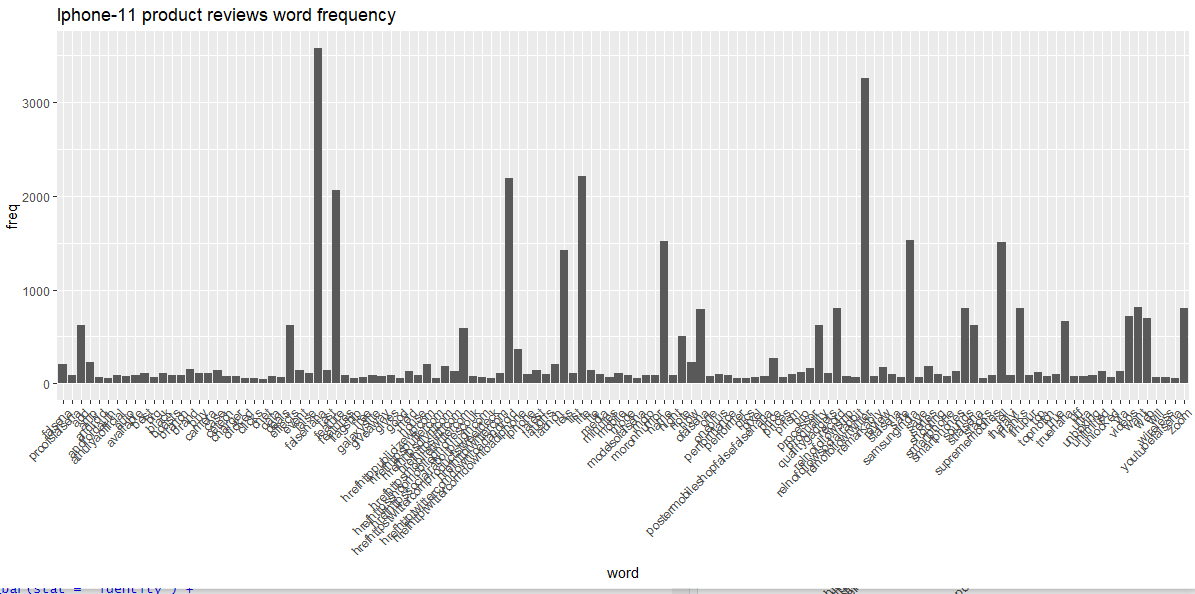
[](https://user-images.githubusercontent.com/5343403/42731417-605a47e6-87d2-11e8-9892-d4d551b2265c.png)

Next in the word-cloud of negative words we see that the word “Galaxy” is the most frequent with “Samsung” the second most frequent word in the list. These two words indicate that the users were not happy with the iPhones and comparing the iPhone-11 with Samsung Galaxy phones, which are the main competitors of iPhone in the market. Also, there are several other negative words like “goodbye”, “issue”, “change”, “sad” indicating that the users are talking about switching brands from iPhone to other phones, possibly Samsung Galaxy phone considering the frequency of the words “Samsung” and “Galaxy”.

[](https://user-images.githubusercontent.com/5343403/42731420-7a1956ea-87d2-11e8-9669-3187a2580d9d.png)

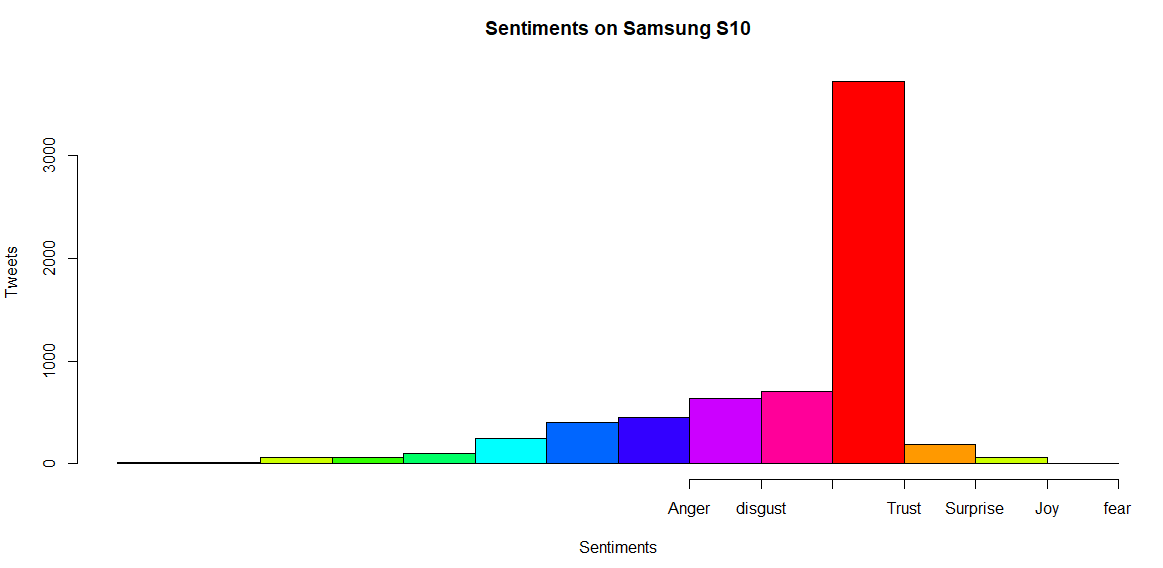
Next in the neutral word cloud we can see the words “Samsung”, “Galaxy”, “android” indicating that the reviewers might have neutrally comparing the iPhone with Android phones.

A ggplot is plotted for iphone11 data for better comparison with Samsung galaxy s10 user reviews. This ggplot shows the words and the frequency the words are used so better idea about the product reviews can be achieved.

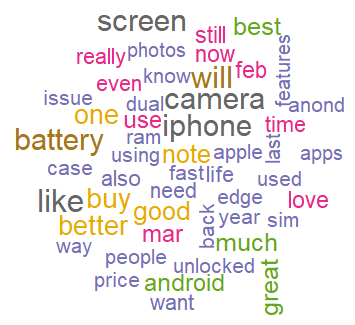


## Samsung Galaxy S10:

The pre-processing part of the tweets for Samsung phone is same as iPhones. A plot which is used to compare with iphone 11 is plotted a histogram which shows user feelings about the product.

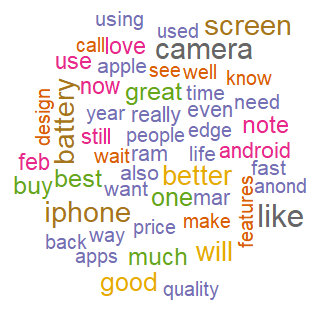


As well as the parts of creating term-document matrix and identifying most frequent words. The self-stop words identified for this case are: "Samsung", "galaxy", "plus", "phone", "got", "get", "new", "anonymous", "phones", "just", "can".

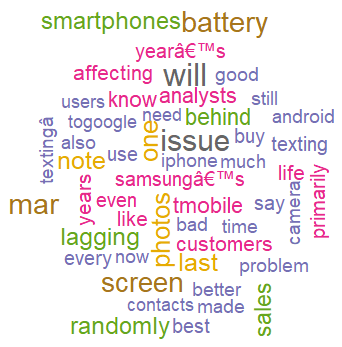
[](https://user-images.githubusercontent.com/5343403/42731429-ced5954a-87d2-11e8-9e6d-c017f60f82fa.png)

Above is the word-cloud for top 50 most frequent words found in the user reviews about Samsung Galaxy S10 phones. Again, here the most frequent word is “iPhone” and “Apple” indicating reviewers are comparing the Samsung galaxy phone with iPhones. Also, the words “edge”, “Note” indicates the Galaxy phones are compared to other Samsung phones from Edge and Note series. Also, the words “Great”, “Best”, “Love” indicates the reviewers loving the Galaxy phone. The words “camera”, “photos” indicate one of the most popular features of the Galaxy phones which is good picture quality.

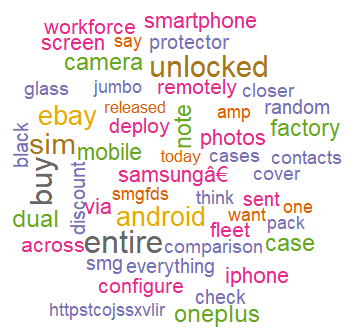
Next the positive, negative and neutral words are identified from the dictionaries and analyzed for Samsung Galaxy S10 reviews:

[](https://user-images.githubusercontent.com/5343403/42731451-2b2a6370-87d3-11e8-8873-164b0cdfdb64.png)

The top mentioned positive words in this case are camera, screen, quality, fast, ram, price, design, apps, indicating reviewers love these features from the Samsung Galaxy S10 phones. Again the words apple and iphone are seen, which means the comparison between Iphone and Galaxy phone is very popular among the users.

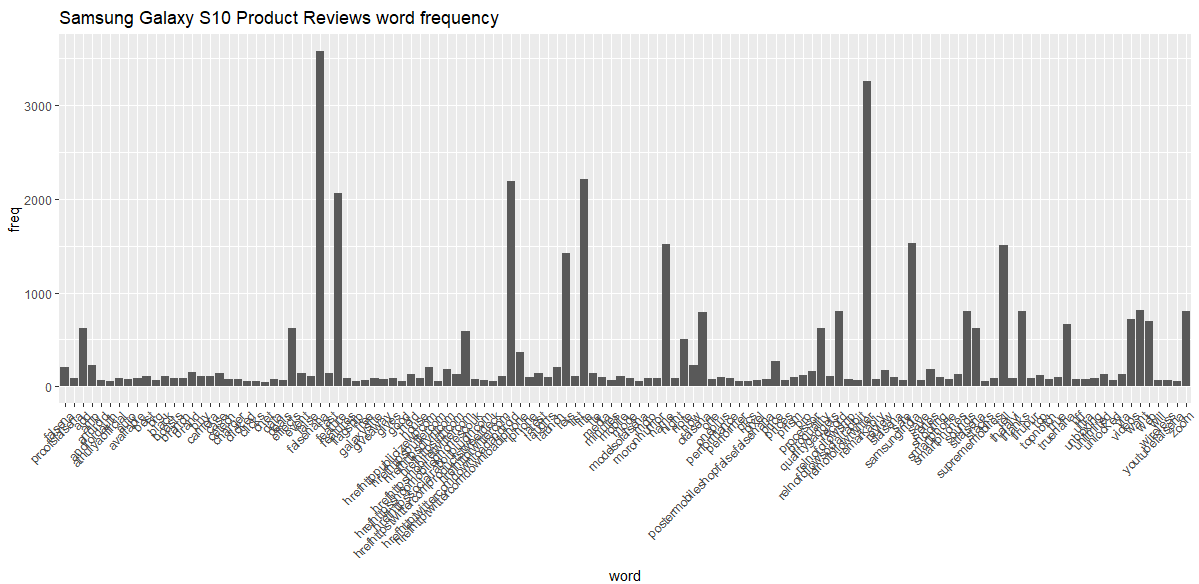
[](https://user-images.githubusercontent.com/5343403/42731457-4922e12c-87d3-11e8-9406-2255ee49b885.png)

In the negative reviews there don’t have many words that can give intuitions about negative user sentiment. However, the words “battery”, “lagging” indicates most of the negative reviews are about smaller battery life and slow speed of the phone.

[](https://user-images.githubusercontent.com/5343403/42731463-6fc65cb4-87d3-11e8-8520-66a2c15eee4c.png)

The neutral reviews also don’t give much intuition about the user sentiment on the Galaxy phones. However the word “Oneplus” is indicating that people are talking about another popular android phone brand that is a major competitor for the samsung phones in Android market. So the business team of Samsung can compare public sentiment on one plus phones in future to better understand user demand on android phones.

A ggplot is plotted which shows all the frequency of words used in the product review and which words are used more.This plot gives better comparison with iphone 11 ggplot which represents words and frequency in the product reviews given by user.



## A pie diagram and a pie diagram-3d is used to represent positive and negative tweets given by users for Samsung Galaxy S10 product review,

## 

## 

## Results & Conclusion:

In this project user reviews were collected directly from Twitter and amazon that sell the phones to identify public sentiment on two of the most popular phones at this time. The reviews were collected randomly to prevent any bias from the user side. In the sentiment analysis part of the comparison we have looked at the top mentioned words as well as the positive, negative and neutral words about both the iPhone 11 and Samsung Galaxy S10 phones. The most common factor found after the sentiment analysis is that users are very prone to compare models that come with similar features. So, knowing about pros and cons of the other features can help the individual brands to grow.

For iPhone, the conclusion after sentiment analysis can be drawn as such: the price range of these phones is a big factor and people are looking for deals and giveaways to buy iPhone. Also, the negative reviews indicate that the Samsung Galaxy phones are mentioned the most along with people talking about changing or switching brand. So, this can be a warning for the brand that Samsung Galaxy phones are a major setback for their market share.

The sentiment on Samsung Galaxy S10 is mostly about the camera, photos, lagging and battery life. While in the positive reviews, users talk about the design of the phone, ram, speed and camera, in the negative reviews users are complaining about lagging and smaller battery life. So, necessary steps should be taken by the brand to improve the user sentiment.

However, one observation should be mentioned about the source of the data which is reviews collected directly from selling sites (like Amazon or eBay) are more specific and intuitive than tweets collected from Twitter where.

However, some of the major issues that can be identified from the user reviews by a human are absent in the results of the analysis. For example, both the phones are known for having screens that are broken very easily. While the top words in the analysis consists, words like “protector”, “case”,” screen”, they do not give us any specific intuition about the user sentiment behind using these words in the reviews.

**References**:

Research papers:

[1] Sarlan A, Nadam C,Basri C, "Twitter Sentiment Analysis", in the proceedings of International Conference on Information Technology and Multimedia (ICIMU), 2014.

[2] Gokulakrishnan B et al., "Opinion mining and sentiment analysis on a twitter data",  in the proceedings of International Conference on  Advances in ICT for Emerging Regions (ICTer), 2012.

Web links:

<http://thinktostart.com/twitter-authentification-with-r/>

<https://www.scrapehero.com/tutorial-how-to-scrape-amazon-product-details-using-python-and-selectorlib/>

<https://towardsdatascience.com/a-light-introduction-to-text-analysis-in-r-ea291a9865a8>

<https://www.tidytextmining.com/tidytext.html>

<https://cran.r-project.org/web/packages/tm/vignettes/>

<https://www.rdocumentation.org/>

Positive & Negative dictionary <http://sentiwordnet.isti.cnr.it/>

Book:

TextMining with R A tidy approach by Julia Silge & David Robinson

**CODE:**

**Twitterwebscrape.R, Amazonwebscrape.R, Iphone11.R, SamsungGalaxayS10.R, S10\_histogram&piediagram.R**

**Twitter webscrape.R:**

library(twitteR)

library(ROAuth)

consumer\_key <-'YqHkgQcVkt5pVclIc1Bb4eit8'

consumer\_secret <-'OUhInyhAmb9WDQ4tKkQHK9GKLBBSKb2F6MGSdqYJO1k42wRdJL'

access\_token <- '908445785670393856-p5Kp68fhQQlOFA7JT97NHQoQFFwtchK'

access\_secret <- 'dgmZUxjF7WnlpdywAy7fermRZ4yfrAqT3CZXXOZlL2Nu2'

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token,access\_secret)

**Amazonwebscrape.R :**

> library(dplyr)

> library(rvest)

> get\_reviews <- function(node){

+

+ scrapeR.title <- html\_nodes(node, ".a-color-base") %>%

+ html\_text()

+

+ scrapeR.author <- html\_nodes(node, ".author") %>%

+ html\_text()

+

+ df <- data.frame(

+ title = ifelse(length(r.title) == 0, NA, scrapeR.title),

+ author = ifelse(length(r.author) == 0, NA, scrapeR.author),

+ stringsAsFactors = F)

+

+ return(df)

+ }

> url <- read\_html("https://www.amazon.de/product-reviews/3980710688/ref=cm\_cr\_dp\_d\_show\_all\_btm?ie=UTF8&reviewerType=all\_reviews&pageNumber=42&sortBy=recent") %>% html\_nodes("div[id\*=customer\_review]")

> library(dplyr)

> library(rvest)

> get\_reviews <- function(node){

+

+ scrapeR.title <- html\_nodes(node, ".a-color-base") %>%

+ html\_text()

+

+ scrapeR.author <- html\_nodes(node, ".author") %>%

+ html\_text()

+

+ df <- data.frame(

+ title = ifelse(length(scrapeR.title) == 0, NA, scrapeR.title),

+ author = ifelse(length(scrapeR.author) == 0, NA, scrapeR.author),

+ stringsAsFactors = F)

+

+ return(df)

+ }

> url <- read\_html("https://www.amazon.com/product-reviews/ B07N4M412B") %>% html\_nodes("div[id\*=customer\_review]")

> out <- lapply(url, get\_reviews) %>% bind\_rows()

> out

title

1 \n\n\n\n\n\n\n\n \n \n My phone was damaged\n \n

2 \n\n\n\n\n\n\n\n \n \n Works good like new\n \n

3 \n\n\n\n\n\n\n\n \n La peor compra!!!!\n \n \n

4 \n\n\n\n\n\n\n\n \n \n Like new\n \n

5 \n\n\n\n\n\n\n\n \n \n 5Stars\n \n

6 \n\n\n\n\n\n\n\n \n \n Fantastic\n \n

7 \n\n\n\n\n\n\n\n \n \n wasn’t too happy with phone turning off a few times on its own once setting up.\n \n

**Iphone 11.R:**

library(tm)

library(pdftools)

library(qdapTools)

library(plyr)

library(syuzhet)

library(stringr)

library(SnowballC)

library(wordcloud)

library(Matrix)

#library(lsa)

library(RColorBrewer)

library(twitteR)

library(ROAuth)

consumer\_key <-'YqHkgQcVkt5pVclIc1Bb4eit8'

consumer\_secret <- 'OUhInyhAmb9WDQ4tKkQHK9GKLBBSKb2F6MGSdqYJO1k42wRdJL'

access\_token <- '908445785670393856-p5Kp68fhQQlOFA7JT97NHQoQFFwtchK'

access\_secret <- 'dgmZUxjF7WnlpdywAy7fermRZ4yfrAqT3CZXXOZlL2Nu2'

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token,access\_secret)

Iphone11.tweets = searchTwitter("Iphone11",lang = "en", n=25000)

#Extracting textual part of the tweets

filepath <-"C:/Dev/workspaceR/Project CSC 522/Final Project/output csv"

#setwd(filepath)

file<-'Iphone11.csv'

text = file(file,open="r")

text.decomposition =readLines(text)

text.decomposition[1]

text.decomposition[2]

corpus <- Corpus(VectorSource(text.decomposition)) # colection of documents containing texts

corpus

# analysis and the sentimental result is ploted using ggplot

msgs = Iphone11[,c("text")]

head(msgs, 3)

library(ggplot2)

library(sentimentr)

d = get\_nrc\_sentiment(msgs)

head(d)

td = data.frame(t(d))

td\_new = data.frame(rowSums(td[1:500]))

# transformation and cleaning

names(td\_new)[1] <- "count"

td\_new <- cbind("sentiment" = rownames(td\_new), td\_new)

rownames(td\_new) <- NULL

td\_new2<- td\_new[1:8,]

qplot(sentiment, data =td\_new2, weight = count, geom="bar", fill= sentiment)+

ggtitle("Iphone-11 tweets")

#using corpus making data into vector

docs = Corpus(VectorSource(msgs))

docs

# removing the stop words

corpus <-tm\_map(corpus,PlainTextDocument)

corpus <-tm\_map(corpus,tolower)

corpus <-tm\_map(corpus,removeNumbers)

stopwords("english")

selfstopwords <- c("iphone","apple", "plus", "phone","now", "iphone11")

corpus <- tm\_map(corpus,removeWords, c(stopwords("english"), selfstopwords))

writeLines(as.character(corpus[[2]]))

corpus <-tm\_map(corpus,removePunctuation)

corpus <-tm\_map(corpus,stripWhitespace)

writeLines(as.character(corpus[[2]]))

corpus.text <-TermDocumentMatrix(corpus)

colnames(corpus.text)<-(1:dim(corpus.text)[2])

write.csv(as.matrix(corpus.text),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/cleaned\_Iphone11.csv"))

frequency<- rowSums(as.matrix(corpus.text))

frequency <- sort(frequency, decreasing=TRUE)

d <- data.frame(word = names(frequency), freq = frequency)

head(frequency)

words <- names(frequency)

wordcloud(words[1:50], frequency[1:50],

scale = c(2, 0.9),colors=brewer.pal(8,"Dark2")) # color palette

#Extracting positive and negative words

positives= readLines("positive-words.txt") #dictionary: we match if theres more +ve/-ve words in our doc

negatives = readLines("negative-words.txt")

which\_pos <-Terms(corpus.text) %in% positives

which\_neg <- Terms(corpus.text) %in% negatives

#find the +ve documents and save the matrix

score\_pos <- colSums(as.matrix(corpus.text[which\_pos, ]))

score\_neg <- colSums(as.matrix(corpus.text[which\_neg, ]) )

score<-score\_pos-score\_neg

score

text\_pos<-corpus.text[ ,score>0]

write.csv(as.matrix(text\_pos),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/pos.csv"))

text\_neg<-corpus.text[ ,score<0]

write.csv(as.matrix(text\_neg),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/neg.csv"))

# find the neutral docs and save the matrix

text\_neu<-corpus.text[ ,score==0]

write.csv(as.matrix(text\_neu),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/neu.csv"))

# plot positive word cloud

frequency\_pos <- rowSums(as.matrix(text\_pos))

frequency\_pos <- sort(frequency\_pos, decreasing=TRUE)

words\_pos <- names(frequency\_pos)

wordcloud(words\_pos[1:50], frequency\_pos[1:50], scale = c(1, 0.7), colors=brewer.pal(8,"Dark2"))

# plot negative word cloud

frequency\_neg <- rowSums(as.matrix(text\_neg))

frequency\_neg <- sort(frequency\_neg, decreasing=TRUE)

words\_neg <- names(frequency\_neg)

wordcloud(words\_neg[1:50], frequency\_neg[1:50], scale = c(1, 0.5), colors=brewer.pal(8,"Dark2"))

# neutral

frequency\_neu <- rowSums(as.matrix(text\_neu))

frequency\_neu <- sort(frequency\_neu, decreasing=TRUE)

words\_neu <- names(frequency\_neu)

wordcloud(words\_neu[1:50], frequency\_neu[1:50], scale = c(2, 0.9), colors=brewer.pal(8,"Dark2"))

findFreqTerms(corpus.tdm, lowfreq = 4)

findAssocs(corpus.tdm, terms = "happy", corlimit =0.3)

head(d, 10)

#barplot(d[1:10,]$freq, las = 2,names.arg = d[1:10,]$words,

# col="lightblue", main="Most frequent words", ylab = " frequencies",xlab ="words")

subset(d, freq > 50) %>%

ggplot(aes(word, freq)) +

geom\_bar(stat = "identity") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

**Samsung Galaxy S10.R:**

#install.packages("dplyr")

library(dplyr)

library(plyr)

#install.packages(c('tm', 'wordcloud'))

library(tm)

filepath <-"C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/S10tweets.csv"

setwd(filepath)

file<-'S10tweets.csv'

text = file(file,open="r")

text.decomposition =readLines(text)

text.decomposition[1]

text.decomposition[2]

corpus <- Corpus(VectorSource(text.decomposition)) # colection of documents containing texts

corpus

corpus <-tm\_map(corpus,PlainTextDocument)

corpus <-tm\_map(corpus,tolower)

corpus <-tm\_map(corpus,removeNumbers)

stopwords("english")

selfstopwords <- c("samsung","galaxy","plus","phone", "got","get", "new", "anonymous", "phones", "just","can")

corpus <- tm\_map(corpus,removeWords, c(stopwords("english"), selfstopwords))

writeLines(as.character(corpus[[2]]))

corpus <-tm\_map(corpus,removePunctuation)

corpus <-tm\_map(corpus,stripWhitespace)

writeLines(as.character(corpus[[2]]))

corpus.tdm <-TermDocumentMatrix(corpus)

colnames(corpus.tdm)<-(1:dim(corpus.tdm)[2])

write.csv(as.matrix(corpus.tdm),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/Galaxys10.csv"))

frequency <- rowSums(as.matrix(corpus.tdm))

frequency <- sort(frequency, decreasing=TRUE)

d <- data.frame(word = names(frequency), freq = frequency)

head(d, 10)

head(frequency)

words <- names(frequency)

wordcloud(words[1:50], frequency[1:50],

scale = c(2, 1),colors=brewer.pal(8,"Dark2")) # color palette

positives= readLines("positive-words.txt") #dictionary: we match if theres more +ve/-ve words in our doc

negatives = readLines("negative-words.txt")

which\_pos <-Terms(corpus.tdm) %in% positives

which\_neg <- Terms(corpus.tdm) %in% negatives

#find the +ve documents and save the matrix

score\_pos <- colSums(as.matrix(corpus.tdm[which\_pos, ]))

score\_neg <- colSums(as.matrix(corpus.tdm[which\_neg, ]) )

score<-score\_pos-score\_neg

score

tdm\_pos<-corpus.tdm[ ,score>0]

write.csv(as.matrix(tdm\_pos),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/tdm\_s10\_pos.csv"))

tdm\_neg<-corpus.tdm[ ,score<0]

write.csv(as.matrix(tdm\_neg),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/tdm\_s10\_neg.csv"))

# find the neutral docs and save the matrix

tdm\_neu<-corpus.tdm[ ,score= =0]

write.csv(as.matrix(tdm\_neu),file=file.path("C:/Dev/workspaceR/Project CSC 522/Final Project/output csv/tdm\_s10\_neu.csv"))

# plot positive word cloud

frequency\_pos <- rowSums(as.matrix(tdm\_pos))

frequency\_pos <- sort(frequency\_pos, decreasing=TRUE)

words\_pos <- names(frequency\_pos)

wordcloud(words\_pos[1:50], frequency\_pos[1:50],scale = c(2, 1),colors=brewer.pal(8,"Dark2"))

# plot negative word cloud

frequency\_neg <- rowSums(as.matrix(tdm\_neg))

frequency\_neg <- sort(frequency\_neg, decreasing=TRUE)

words\_neg <- names(frequency\_neg)

wordcloud(words\_neg[1:50], frequency\_neg[1:50],scale = c(2, 0.8),colors=brewer.pal(8,"Dark2")) #scale = c(2, 0.8)

# neutral

frequency\_neu <- rowSums(as.matrix(tdm\_neu))

frequency\_neu <- sort(frequency\_neu, decreasing=TRUE)

words\_neu <- names(frequency\_neu)

wordcloud(words\_neu[1:50], frequency\_neu[1:50],scale = c(2, 0.8),colors=brewer.pal(8,"Dark2"))

findFreqTerms(corpus.tdm, lowfreq = 4)

findAssocs(corpus.tdm, terms = "happy", corlimit =0.3)

head(d, 10)

#barplot(d[1:10,]$freq, las = 2,names.arg = d[1:10,]$words,

# col="lightblue", main="Most frequent words", ylab = " frequencies",xlab ="words")

subset(d, freq > 50) %>%

ggplot(aes(word, freq)) +

geom\_bar(stat = "identity") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

**S10\_histogram&piediagram.R:**

library(twitteR)

library(ROAuth)

library(plyr)

consumer\_key <-'YqHkgQcVkt5pVclIc1Bb4eit8'

consumer\_secret <- 'OUhInyhAmb9WDQ4tKkQHK9GKLBBSKb2F6MGSdqYJO1k42wRdJL'

access\_token <- '908445785670393856-p5Kp68fhQQlOFA7JT97NHQoQFFwtchK'

access\_secret <- 'dgmZUxjF7WnlpdywAy7fermRZ4yfrAqT3CZXXOZlL2Nu2'

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token,access\_secret)

# Extracting data from twitter after connection authorization

s10.tweets = searchTwitter("Samsung+Galaxy+s10",lang = "en", n=25000)

s10tweets <- s10.tweets

#dim(s10.tweets)

#class(s10.tweets)

#str(s10.tweets)

#Extracting textual part of the tweets

sample=NULL #Initialising

for (tweet in s10.tweets)

sample = c(sample,tweet$getText())

#converts to data frame

df <- do.call("rbind", lapply(s10.tweets, as.data.frame))

#remove odd characters

df$text <- sapply(df$text,function(row) iconv(row, "latin1", "ASCII", sub="")) #remove emoticon

df$text = gsub("(f|ht)tp(s?)://(.\*)[.][a-z]+", "", df$text) #remove URL

sample <- df$text

setwd("C:/Dev/workspaceR/Project CSC 522")

neg.words = scan("negative-words.txt", what="character", comment.char=";")

pos.words = scan("positive-words.txt", what="character", comment.char=";")

# clean the tweets

score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{

require(plyr)

require(stringr)

list=lapply(sentences, function(sentence, pos.words, neg.words)

{

sentence = gsub('[[:punct:]]',' ',sentence)

sentence = gsub('[[:cntrl:]]','',sentence)

sentence = gsub('\\d+','',sentence) #removes decimal number

sentence = gsub('\n','',sentence) #removes new lines

sentence = tolower(sentence)

word.list = str\_split(sentence, '\\s+')

words = unlist(word.list) #changes a list to character vector

pos.matches = match(words, pos.words)

neg.matches = match(words, neg.words)

pos.matches = !is.na(pos.matches)

neg.matches = !is.na(neg.matches)

pp = sum(pos.matches)

nn = sum(neg.matches)

score = sum(pos.matches) - sum(neg.matches)

list1 = c(score, pp, nn)

return (list1)

}, pos.words, neg.words)

score\_new = lapply(list, `[[`, 1)

pp1 = lapply(list, `[[`, 2)

nn1 = lapply(list, `[[`, 3)

scores.df = data.frame(score = score\_new, text=sentences)

positive.df = data.frame(Positive = pp1, text=sentences)

negative.df = data.frame(Negative = nn1, text=sentences)

list\_df = list(scores.df, positive.df, negative.df)

return(list\_df)

}

library(stringr)

result = score.sentiment(sample, pos.words, neg.words)

#install.packages("stringr")

#library(stringr)

#Reshaping the tweets into table

#install.packages("reshape")

library(reshape)

library(plyr)

#Creating a copy of result data frame

test1=result[[1]]

test2=result[[2]]

test3=result[[3]]

#Creating three different data frames for Score, Positive and Negative

#Removing text column from data frame

test1$text=NULL

test2$text=NULL

test3$text=NULL

#Storing the first row(Containing the sentiment scores) in variable q

q1=test1[1,]

q2=test2[1,]

q3=test3[1,]

qq1=melt(q1,var='Score')

qq2=melt(q2,var='Positive')

qq3=melt(q3,var='Negative')

qq1['Score'] = NULL

qq2['Positive'] = NULL

qq3['Negative'] = NULL

#Creating data frame

table1 = data.frame(Text=result[[1]]$text, Score=qq1)

table2 = data.frame(Text=result[[2]]$text, Score=qq2)

table3 = data.frame(Text=result[[3]]$text, Score=qq3)

#Merging three data frames into one

table\_final=data.frame(Text=table1$Text, Score=table1$value, Positive=table2$value, Negative=table3$value)

#View(table\_final)

allwords<-c("Anger","disgust","Anticipation","fear","Surprise","Joy","Trust")

#Histogram

hist(table\_final$Positive, col=rainbow(10))

hist(table\_final$Negative, col=rainbow(10))

hist(table\_final$Score, col=rainbow(10),xaxt='n',xlab = "Sentiments",ylab = "Tweets",main="Sentiments on Samsung S10")

axis(1, at = seq(-3,3,by=1), labels = allwords)

#Pie

slices <- c(sum(table\_final$Positive), sum(table\_final$Negative))

labels <- c("Negative", "Positive")

#install.packages("plotrix")

library(plotrix)

pie(slices, labels = labels, col=rainbow(length(labels)), main="Sentiment Analysis")

pie3D(slices, labels = labels, col=rainbow(length(labels)),explode=0.00, main="Twitter Sentiment Analysis of Samsung S10")