Module 6-CaseStudy

Text Mining

Load the required packages and libraries

**library**(tm)

## Loading required package: NLP

**library**(SnowballC) **library**(stringr)

**library**(readr)

**library**(wordcloud)

## Loading required package: RColorBrewer

**library**(DT)

Laod the data ‘tweets.csv’, don’t forget to add the argument ‘stringsAsFactors = FALSE’.

tweets <- read.csv("tweets.csv", stringsAsFactors = FALSE) View(tweets)

summary(tweets)

## Tweet Avg

## Length:1181 Min. :-2.0000

## Class :character 1st Qu.:-0.6000 ## Mode :character Median : 0.0000 ## Mean :-0.1931

## 3rd Qu.: 0.2000

## Max. : 2.0000

Now, extract the relevant variable, the one containing the text.

r1 = as.character(tweets$Tweet)

Set the seed to 100 for code reproducibility

set.seed(100)

Run the following command, ‘sample = sample(r1, (length(r1)))’, in your RStudio, now you are ready for Bag of Words

sample = sample(r1, (length(r1)))

Data Preprocessing using Bag of Words Technique

Create a Corpus - which, in simple terms, is nothing but a collection of text documents.

corpus = Corpus(VectorSource(list(sample)))

Now, remove punctuations

corpus = tm\_map(corpus, removePunctuation)

Next, change the case of the word to lowercase so that same words are not counted as different because of lower or upper case

corpus = tm\_map(corpus, content\_transformer(tolower))

Next, remove numbers

corpus = tm\_map(corpus, removeNumbers)

Next, remove whitespaces

corpus = tm\_map(corpus, stripWhitespace)

Now, remove unhelpful terms, also referred as stopwords

corpus = tm\_map(corpus, removeWords, stopwords('english'))

Now, please carry out the process of stemming, motivated by the desire to represent words with different endings as the same word.

corpus = tm\_map(corpus, stemDocument)

Creat a document term matrix from the corpus

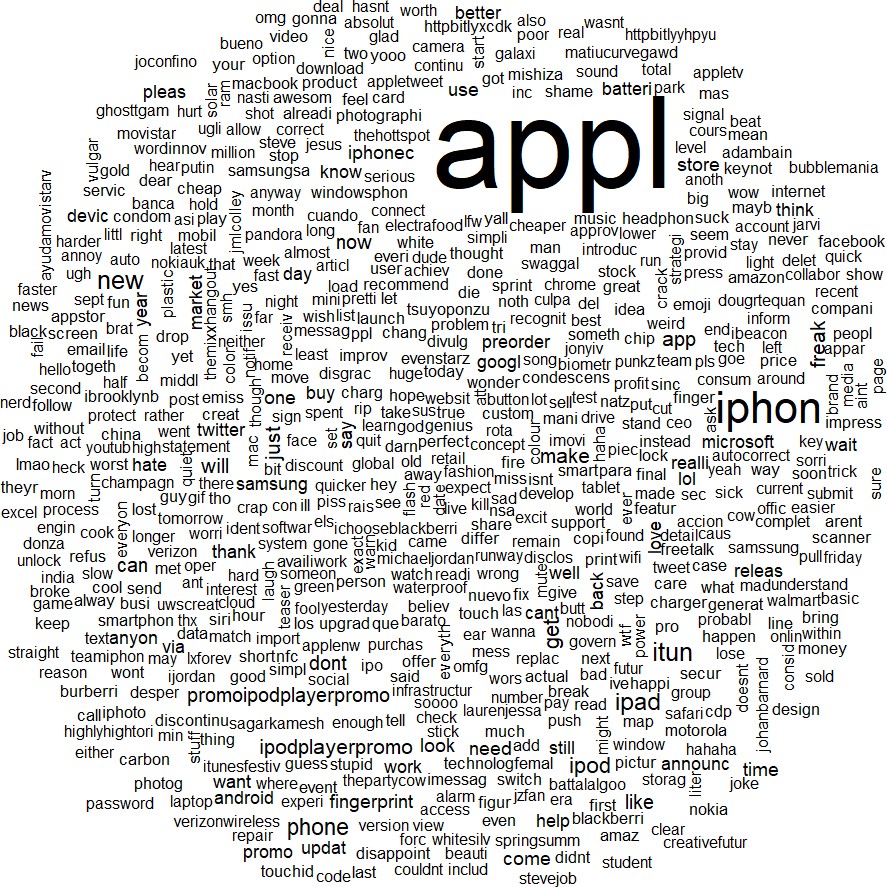
frequencies = DocumentTermMatrix(corpus)

Now create the data frame from the output of the above line

allTweets = as.data.frame(as.matrix(frequencies))

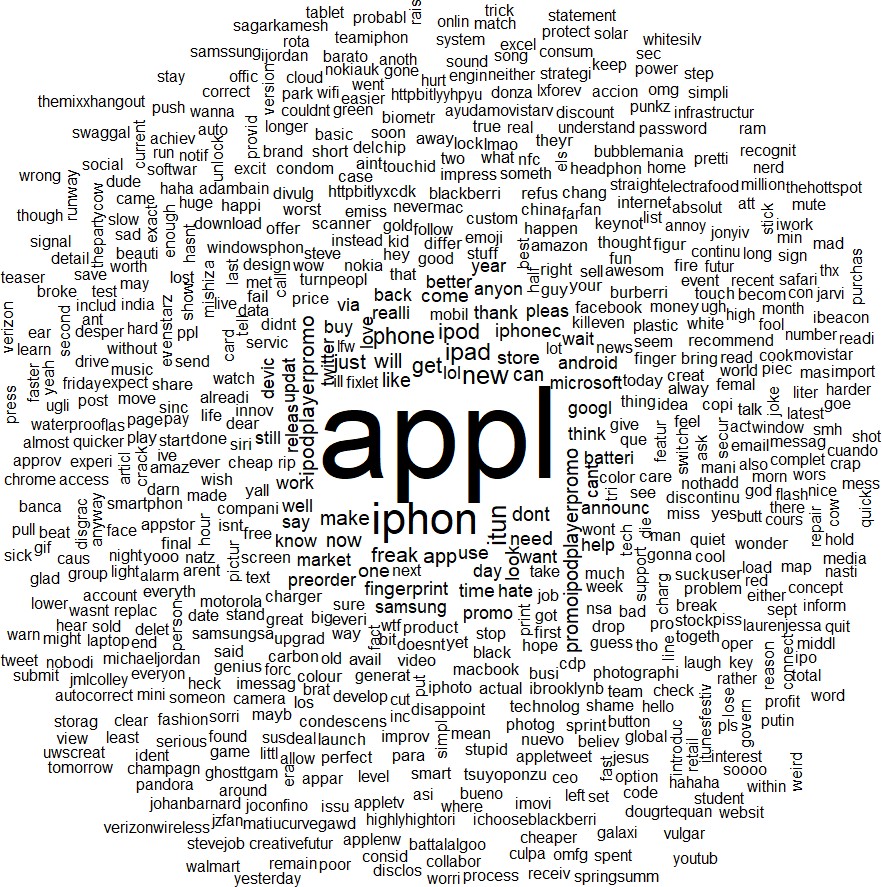
Create a word cloud and set random.order = TRUE:

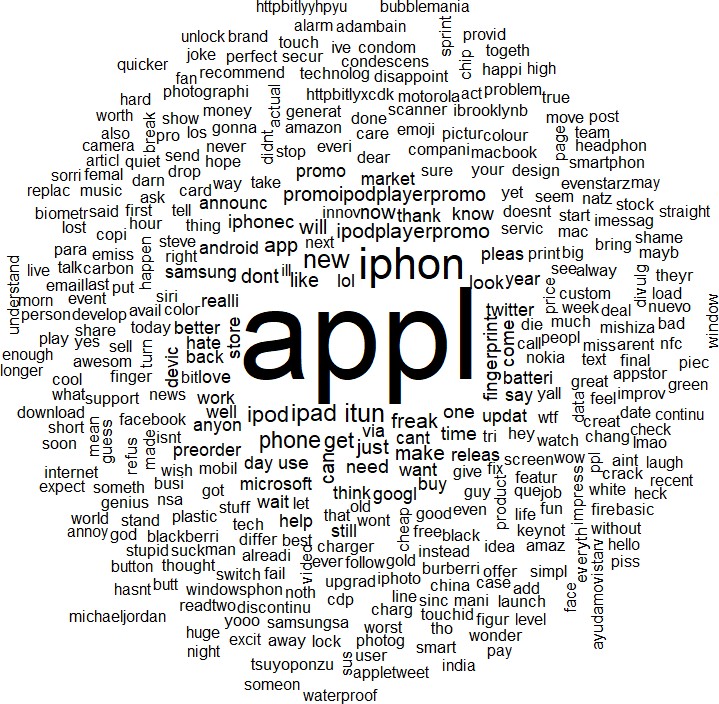
wordcloud(colnames(allTweets), colSums(allTweets), random.order=TRUE)



Create a word cloud and set random.order = FALSE:

wordcloud(colnames(allTweets), colSums(allTweets), random.order=FALSE)



In the above word cloud, adjust the frequency level with min.freq parameter set at 5

wordcloud(colnames(allTweets), colSums(allTweets), random.order=FALSE, min.freq=5)

Creating Word Cloud for Negative Tweets

Create a new dataframe from the original data ‘tweets’ which only includes negative tweets, where the Avg Value is less than zero

negativeTweets = subset(tweets, Avg < 0)

Run the follwoing commands in your R Studio Extracting relevant negative tweets as character

n1 = as.character(negativeTweets$Tweet)

Set the seed to 100 for code reproducibility

set.seed(100)

Sample

sample2 = sample(n1, (length(n1)))

Bag of Words - Run the above codes

1 - Create a Corpus 2 - Remove punctuations 3 - Convert to lowercase 4 - Remove Numbers 5 - Remove whitespaces 6 - Remove stopwords 7 - Perform Stemming

corpus2 = Corpus(VectorSource(list(sample2))) corpus2 = tm\_map(corpus2, removePunctuation)

corpus2 = tm\_map(corpus2, content\_transformer(tolower)) corpus2 = tm\_map(corpus2, removeNumbers)

corpus2 = tm\_map(corpus2, stripWhitespace)

corpus2 = tm\_map(corpus2, removeWords, stopwords('english')) corpus2 = tm\_map(corpus2, stemDocument)

Create a document term matrix from the resultant corpus

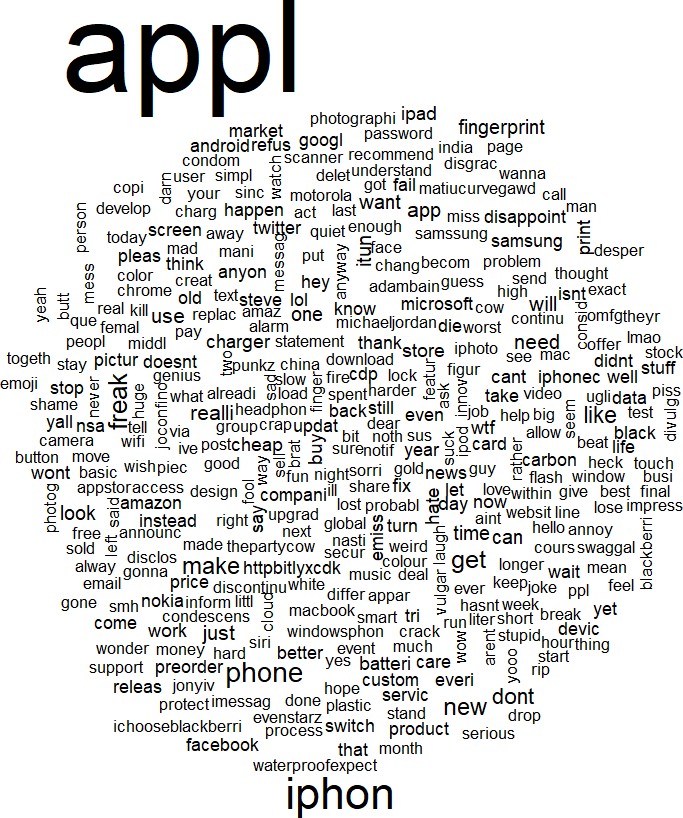
frequencies2 = DocumentTermMatrix(corpus2)

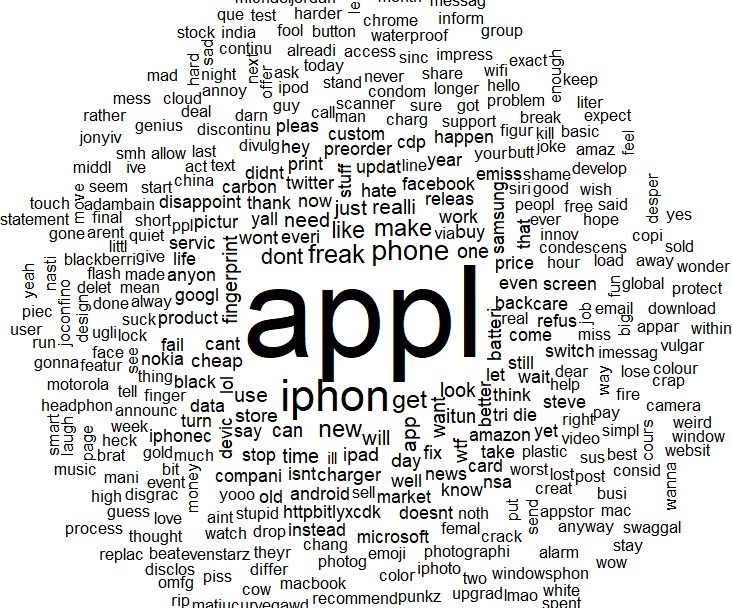
Now create the data frame from the output of the above line

negTweets = as.data.frame(as.matrix(frequencies2))

Create three word clouds using the following three instructions WordCloud 1 - Create a word cloud and set random.order = TRUE. WordCloud 2 - Create a word cloud and set random.order = FALSE WordCloud 3 - In the above word cloud, adjust the frequency level with min.freq parameter set at 5

wordcloud(colnames(negTweets), colSums(negTweets), random.order=TRUE)





wordcloud(colnames(negTweets), colSums(negTweets), random.order=FALSE)

wordcloud(colnames(negTweets), colSums(negTweets), random.order=FALSE, min.freq=5)

Creating Word Cloud for Positive Tweets

create a new dataframe from the original data ‘tweets’ which only includes positive tweets, where the Avg Value is greater than zero, and name the data frame as ‘positiveTweets’

positiveTweets = subset(tweets, Avg > 0)

Now, run the follwoing commands in your R Studio Extracting relevant positive tweets

p1 = as.character(positiveTweets$Tweet)

Set the seed to 100 for code reproducibility

set.seed(100)

Sample

sample3 = sample(p1, (length(p1)))

Bag of Words - Run the above codes 1 - Create a Corpus 2 - Remove punctuations 3 - Convert to lowercase 4 - Remove Numbers 5 - Remove whitespaces 6 - Remove stopwords 7 - Perform Stemming

corpus3 = Corpus(VectorSource(list(sample3))) corpus3 = tm\_map(corpus3, removePunctuation)

corpus3 = tm\_map(corpus3, content\_transformer(tolower)) corpus3 = tm\_map(corpus3, removeNumbers)

corpus3 = tm\_map(corpus3, stripWhitespace)

corpus3 = tm\_map(corpus3, removeWords, stopwords('english')) corpus3 = tm\_map(corpus3, stemDocument)

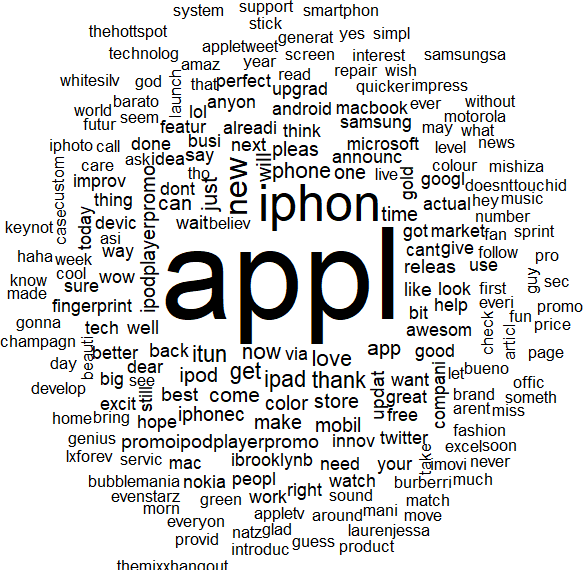
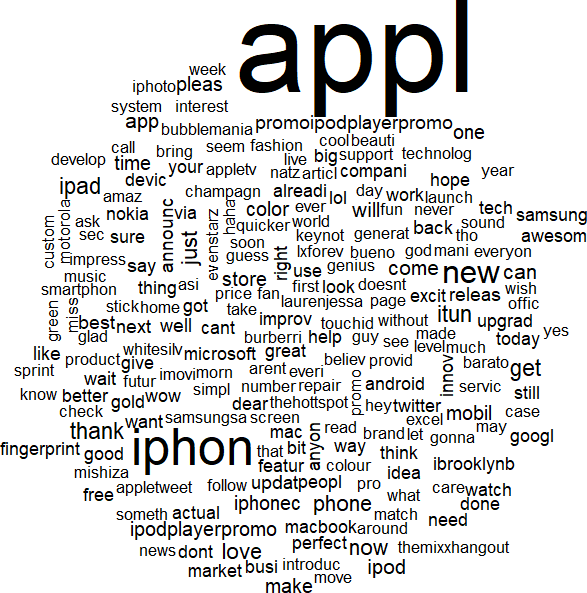
Create a document term matrix from the resultant corpus

frequencies3 = DocumentTermMatrix(corpus3)

Now create the data frame from the output of the above line

posTweets = as.data.frame(as.matrix(frequencies3))

Create three word clouds using the following three instructions

WordCloud 1 - Create a word cloud and set random.order = TRUE. WordCloud 2 - Create a word cloud and set random.order = FALSE WordCloud 3 - In the above word cloud, adjust the frequency level with min.freq parameter set at 5

wordcloud(colnames(posTweets), colSums(posTweets), random.order=TRUE)

wordcloud(colnames(posTweets), colSums(posTweets), random.order=FALSE)

wordcloud(colnames(posTweets), colSums(posTweets), random.order=FALSE, min.freq=5)

