Sentiment Analysis On Consumer Reviews

By,
Sarabu Sai Prashanth.

INTRODUCTION

It is important for the business to take an effective decision to keep it successful all days. Almost all the organizations in the world from small start - ups to largely scaled companies rely on what data is speaking about. Now a days, the effective decisions are taken based on the insights provided by the data. In this report, we are going to discuss about the sentiment analysis of consumer reviews which helps business understand the sentiment of the consumers and their state of emotion.

By using text mining concepts, sentiment of the reviews can be classified into different emotions and visualized using the plots and graphs. The whole text of the consumer reviews undergoes the sentiment analysis to provide insights of the data then plotted comparing with different factors to make it simple for the reader to understand the data insights.

I will be using R programming for analysing the sentiment of the consumer reviews. R is extensively used which is the best statistical tool for the analysis of the data. R is mostly used for the statistical analysis and it is convenient to analyse the data with the help of its extensive packages available.

The whole process is detailedly discussed below step by step to understand the procedure of the analysis of the text of consumer reviews. Each step provided the clear understanding of the necessity with the help of the code and its explanation.

SOLUTION

Step 1 : Load the necessary packages

```
Source on Save
1  library(DT)
2  library(tidytext)
3  library(dplyr)
4  library(stringr)
5  library(sentimentr)
6  library(ggplot2)
7  library(RColorBrewer)
8  library(readr)
9  library(SnowballC)
10 library(tm)
11 library(wordcloud)
12 library(reticulate)
13 library(crfsuite)
14
15
```

Step 2 : Import Consumer reviews from the system by using “read.delim” function.

```
# Read a txt file

customer_reviews <- read.delim(file.choose())
View(customer_reviews)

str(customer_reviews)
attach(customer_reviews)
```

Step 3 : Organizing the text with the help of tm package by applying cleaning functions to a corpus. Entire corpus makes scaling of the cleaning steps very easy.

```
customer_reviews <- as.character(customer_reviews)
customer_courpus <- Corpus(VectorSource(comments))
print(customer_courpus)

customer_courpus <- tm_map(customer_courpus, tolower)
inspect(customer_courpus[1:6])

customer_courpus <- tm_map(customer_courpus,removePunctuation)
inspect(customer_courpus[1:6])

customer_courpus <- tm_map(customer_courpus,removeNumbers)
inspect(customer_courpus[1:6])

customer_courpus <-tm_map(customer_courpus,stripWhitespace)
inspect(customer_courpus[1:6])

cleanset <- tm_map(customer_courpus,removeWords, stopwords('english'))
inspect(cleanset[1:6])
```

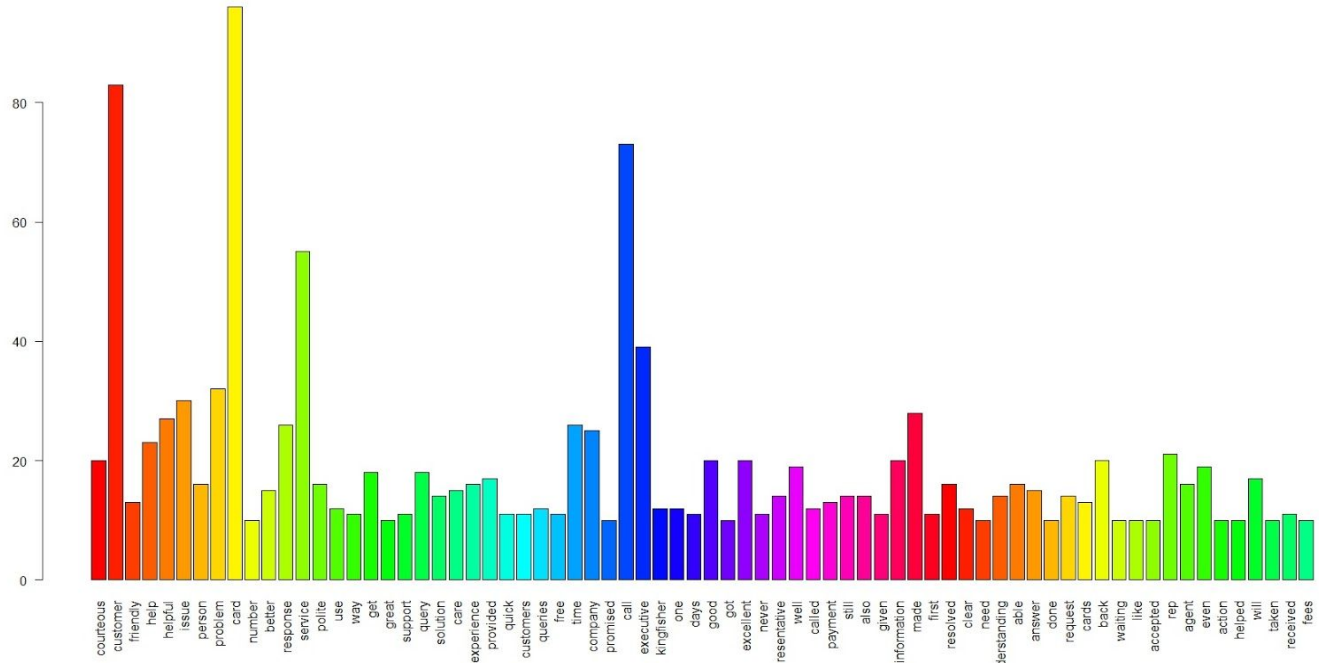
Step 4 : After Completing basic cleaning and pre-processing of text, Term document matrix is generated(TDM) and converted to matrix.

```
tdm <- TermDocumentMatrix(cleanset)
tdm


tdm <- as.matrix(tdm)
tdm[1:10,1:20]
```

Step 5 : Now finding the frequently repeated words and building the bar plot of repeated words.

```
w <- rowSums(tdm)  # provides the no of times a particular word has been used.
w <- subset(w, w>=10) # Pull words that were used more than 10 times.
barplot(w, las = 2, col = rainbow(50))
```

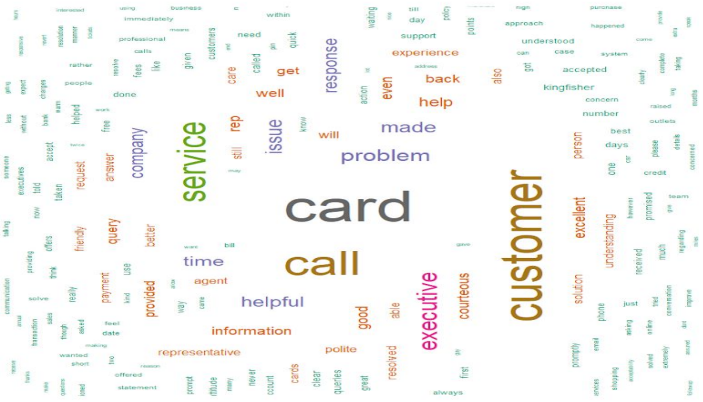
Output :



Step 6 : Creating a word cloud of frequent words using package “wordcloud”.

```
# Word Cloud :
w <- sort(rowSums(tdm), decreasing = TRUE) # Sort words in decreasing order.
set.seed(123)
wordcloud(words = names(w), freq = w,
          max.words = 250,random.order = F,
          min.freq =  3,
          colors = brewer.pal(8, 'Dark2'),
          scale = c(5,0.3),
          rot.per = 0.6)
```

Output :



Creating word cloud using the package “wordcloud2”.

```
library(wordcloud2)
w <- data.frame(names(w),w)
colnames(w) <- c('word','freq')
wordcloud2(w,size = 0.5, shape = 'triangle', rotateRatio = 0.5, minSize = 1)
```

Output :

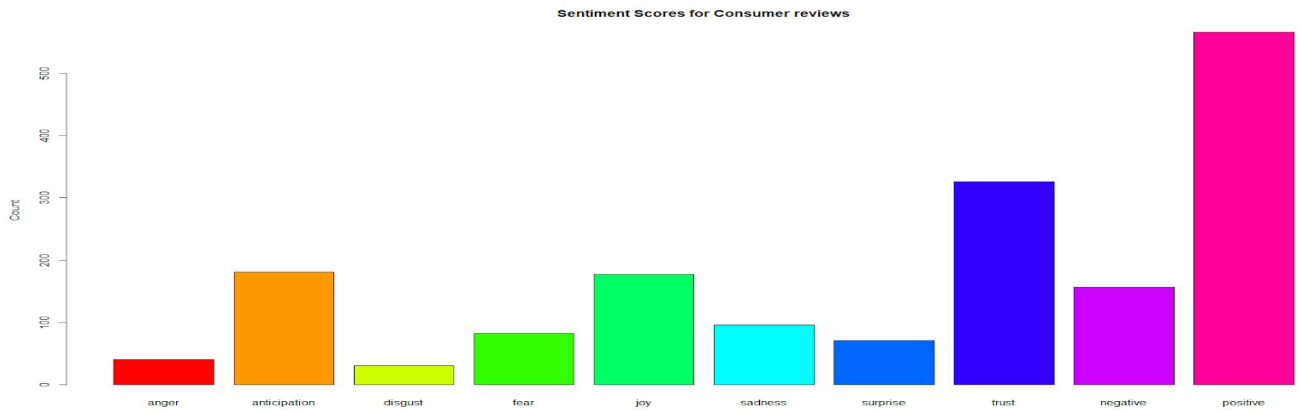
Step 7 : Finding the Sentiment of the reviews and its score.

This Code prints the sentiment of all the reviews present inside the customer_rnws. This is obtained by using the package called “syuzhet”. “get_nrc_sentiment” is a function used to find the sentiment of the reviews.

```
#Sentiment Analysis
library(syuzhet)
library(lubridate)
library(scales)
library(reshape2)


customer_rvws <- read.delim(file.choose())
customer_reviews <- iconv(customer_rvws$comments)

s <- get_nrc_sentiment(customer_reviews)

get_nrc_sentiment('delay')
get_nrc_sentiment('ugly')
```

Example :
Sentiment scores of word “delay”
get_nrc_sentiment('delay')
anger anticipation disgust fear joy sadness surprise trust negative positive
1 0 1 1 0 1 0 0 1 0

```
#Barplot
barplot(colSums(s),
        las = 2,
        col = rainbow(10),
        ylab = 'Count',
        main = 'Sentiment Scores for Consumer reviews')
```



Barplot of sentiment of consumer reviews.

By seeing the barplot, we can observe that positive reviews are dominating in overall followed by Trust, anticipation, and joy.

Step 8 : Now, finding the positive, negative scores of the consumer reviews.
This below code imports the positive and negative text files from the location and gives scores for consumer reviews.

```
#Now finding the negative, positive and neutral scores of the consumer reviews.

posText <- read.delim(file.choose(), header = FALSE, stringsAsFactors = FALSE)
posText <- posText$V1
posText <- unlist(lapply(posText, function(x) { str_split(x, "\n")}))
negText <- read.delim(file.choose(), header = FALSE, stringsAsFactors = FALSE)
negText <- negText$V1
negText <- unlist(lapply(negText, function(x) { str_split(x, "\n")}))
pos.words = c(posText, 'upgrade')
neg.words = c(negText, 'wtf', 'wait', 'waiting', 'epicfail', 'mechanical')
```

Importing negative and positive words to find the scores.

```
pos.words = scan(file.choose(), what='character', comment.char=';')
neg.words = scan(file.choose(), what='character', comment.char=';')

neg.words = c(neg.words, 'wtf', 'fail')
```

Now, Implementing sentiment scoring algorithm to obtain positive and negative scoring.

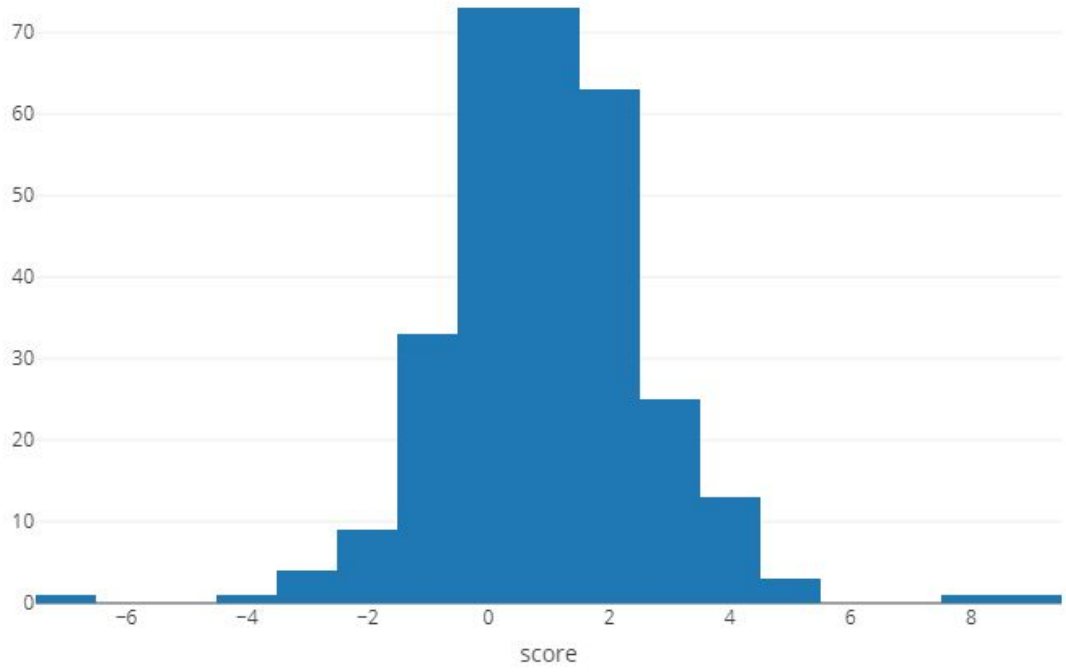
```
#Implementing our sentiment scoring algorithm
require(plyr)
require(stringr)
require(stringi)

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

  # we got a vector of sentences. plyr will handle a list
  # or a vector as an "l" for us
  # we want a simple array of scores back, so we use
  # "l" + "a" + "ply" = "laply":
  scores = laply(sentences, function(sentence, pos.words, neg.words) {

    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    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+')
    # sometimes a list() is one level of hierarchy too much
    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)
}

sentiment.scores= score.sentiment(customer_reviews, pos.words, neg.words, .progress='none')
```

Now, Finding the scores obtained from the above code. With the help of below command the scores are printed.

```
sentiment.scores= score.sentiment(customer_reviews, pos.words, neg.words, .progress='none')

score <- sentiment.scores$score
score
```

Histogram of sentiment scores by using library “plotly”

```
View(sentiment.scores)
hist(sentiment.scores$score)


library(plotly)

p <- plot_ly(x = ~score, type = "histogram")
p
```

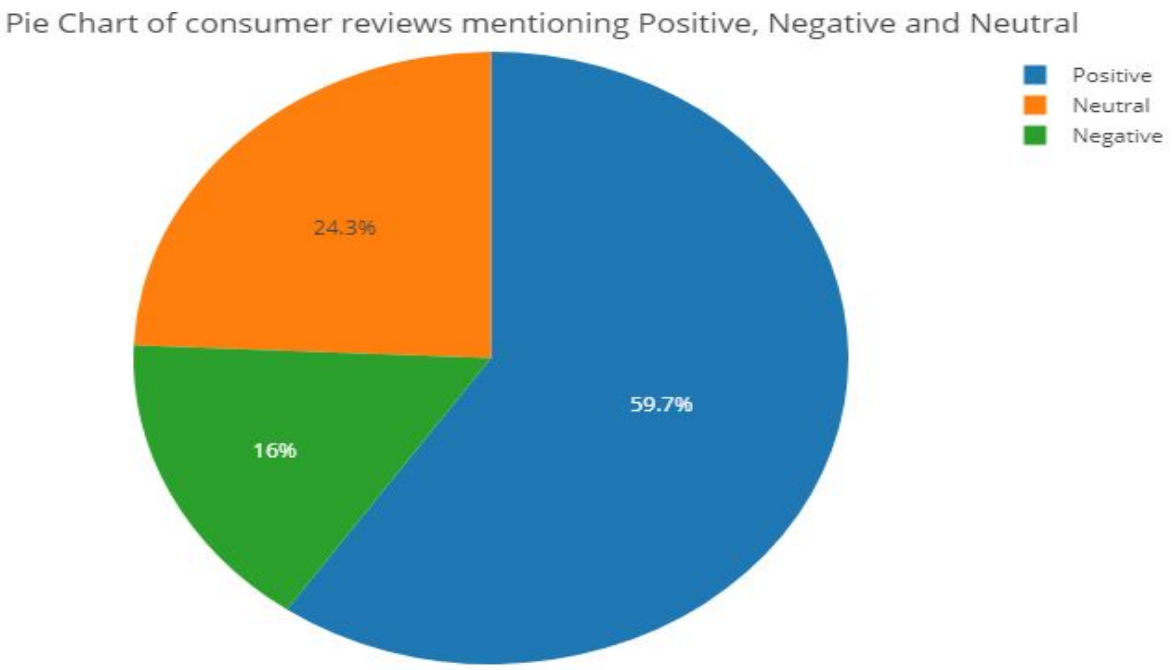


Step 9 : we are going to classify the reviews into positive, negative and neutral in order to explore. Consumer reviews will be divided based on the scores into positive, negative and neutral with the help of the below command.

```
sentiment.scores= score.sentiment(sentiment.scores$text, pos.words, neg.words, .progress='none')
pos.mentioned = subset(sentiment.scores, score > 0)
neu.mentioned = subset(sentiment.scores, score == 0)
neg.mentioned = subset(sentiment.scores, score < 0)

N= nrow(sentiment.scores)
Npos = nrow(pos.mentioned)
Nneu = nrow(neu.mentioned)
Nneg = nrow(neg.mentioned)

dftemp=data.frame(topic=c("Positive", "Negative", "Neutral" ),
                  number=c(Npos, Nneg, Nneu))
```

Pie chart of overall sentiment scores of negative, positive and neutral reviews. By using the library “plotly” we can visualize pie chart of the overall sentiment score portions.

```
#Pie chart of the negative, positive and neutral reviews.
library(plotly)
p <- plot_ly(data=dftemp, labels = ~topic, values = ~number, type = 'pie') %>%
  layout(title = 'Pie Chart of consumer reviews mentioning Positive, Negative and Neutral',
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
p
```



Step 10 : Now, writing the positive, negative and neutral reviews to a seperate text file. With the help of the below command negative, positive and neutral reviews are saved in the separate file in the system separately.

```
#Writing the negative reviews to a text file.
negative_Reviews = subset(sentiment.scores$text, score < 0)
View(negative_Reviews)

#All negative comments are written to a text file
write.table(negative_Reviews, "C:")
#-----------------------------------------------------------------
#Writing positive revuews into  a text file.
positive_reviews = subset(sentiment.scores, score > 0)
View(positive_reviews)
corpus_p = Corpus(VectorSource(positive_reviews))



#All positive reviews are written to a text file.
write.table(positive_reviews, "C:")

#-----------------------------------------------------------------
#writing the neutral reviews are written to a text file.
neutral_reviews = subset(sentiment.scores, score == 0)
View(neutral_reviews)
corpus_neutral = Corpus(VectorSource(neutral_reviews))

#All neutral reviews are written to a text file.
write.table(neutral_reviews, "C:")
```

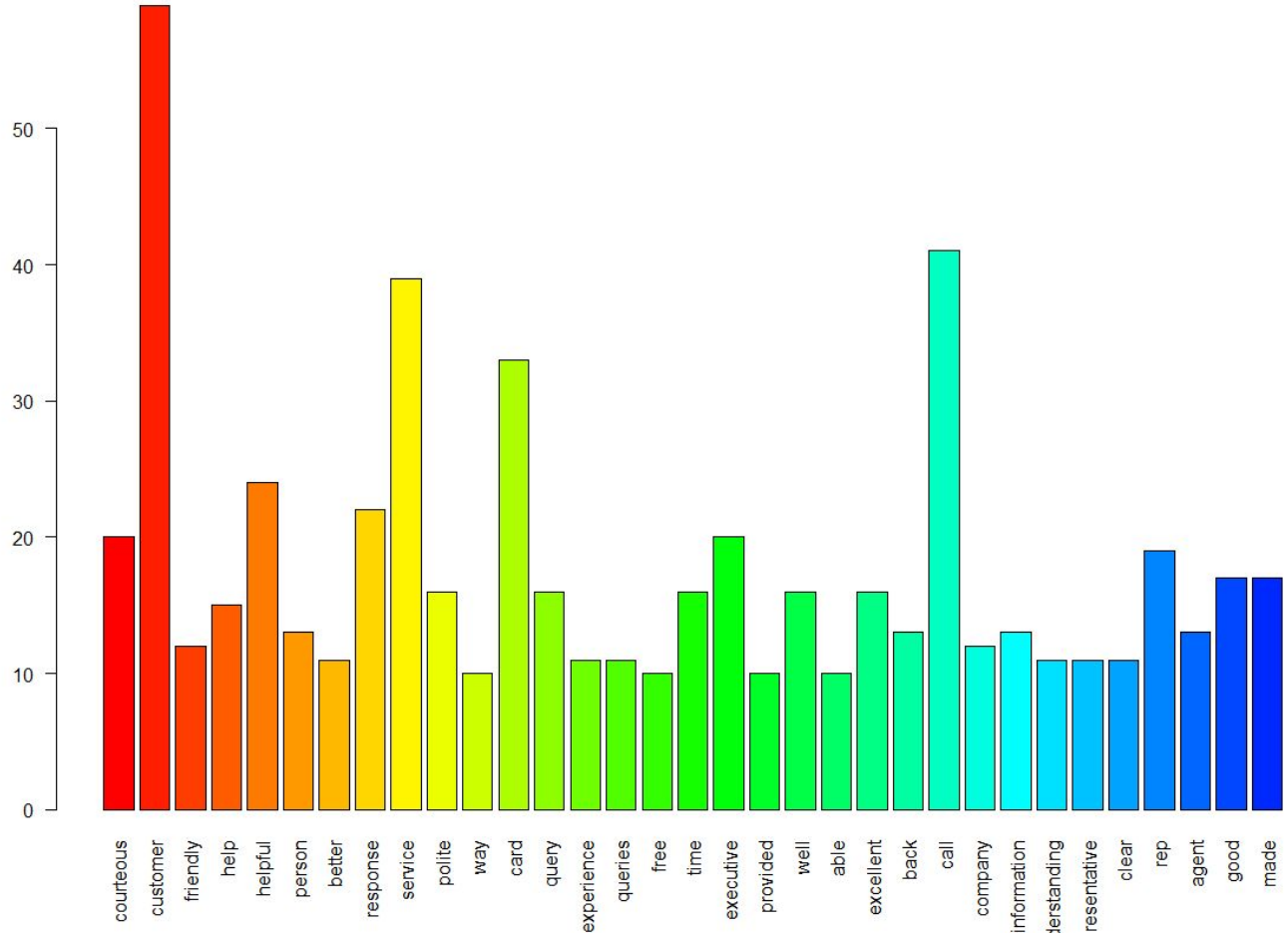
Now, negative, positive and neutral reviews be separately imported and create wordCloud of all the reviews.

Step 11 : Importing positive, negative reviews and creating the wordCloud of the reviews.

```
positive_reviews <- read.delim(file.choose(), header = FALSE)

View(positive_reviews)

str(positive_reviews)
attach(positive_reviews)
```

Now, Creating a barplot of frequently repeated words of the positive reviews.

```
w <- rowSums(tdm_positive)  # provides the no of times a particular word has been used.
w <- subset(w, w>=10) # Pull words that were used more than 25 times.
barplot(w, las = 2, col = rainbow(50))
```

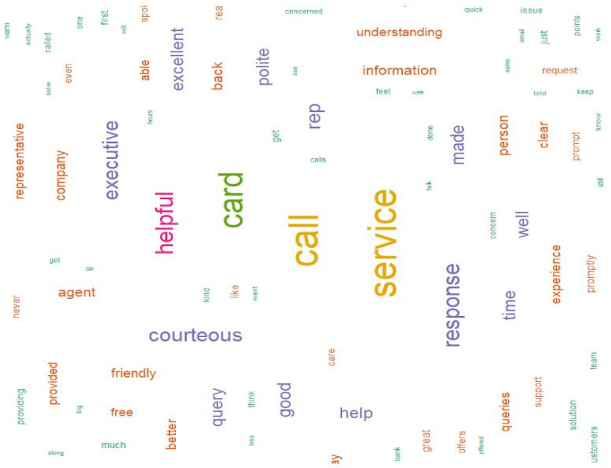


Customer, call and response are most frequently repeated words in the positive reviews.

Now, Creating a WordCloud of the positive reviews. With the help of package called “worCloud” the below wordCloud visualization is made. Below command helps to print the wordCloud.

```
# Word Cloud :
w <- sort(rowSums(tdm_positive), decreasing = TRUE) # Sort words in decreasing order.
set.seed(123)
wordcloud(words = names(w), freq = w,
          max.words = 250,random.order = F,
          min.freq =  3,
          colors = brewer.pal(8, 'Dark2'),
          scale = c(5,0.3),
          rot.per = 0.6)
```

Output :



Creating a wordCloud using the package “wordcloud2”

```
library(wordcloud2)
w <- data.frame(names(w),w)
colnames(w) <- c('word','freq')
wordcloud2(w,size = 0.5, shape = 'triangle', rotateRatio = 0.5, minSize = 2)
```

Output :

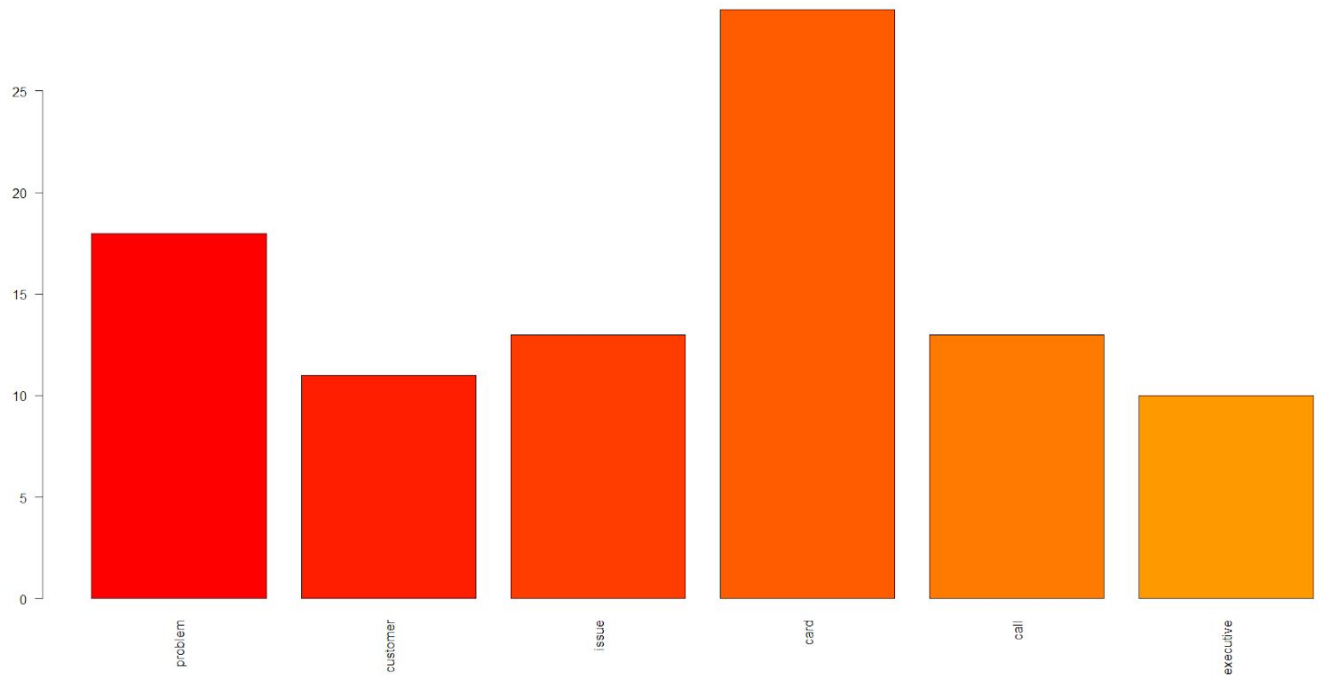
Now, Importing negative reviews which is written into the drive. With the help of the below code the negative reviews will be imported from the drive to the environment for the analysis.

```
#With the help R code, Negative, Positive and Neutral reviews differentiated.
#The Created negative, positive and neutral are saved in the drive with the help of write.table function

negative_Reviews <- read.delim(file.choose())

View(negative_Reviews)

str(negative_Reviews)
attach(negative_Reviews)
```

Finding frequently repeated words using the below command. With the help of below command the barplot is visualized.

```
w <- rowSums(tdm_negative)  # provides the no of times a particular word has been used.
w <- subset(w, w>=10) # Pull words that were used more than 25 times.
barplot(w, las = 2, col = rainbow(50))
```

Output :



Card is the most repeated word among the negative reviews followed by problem and customer.

WordCloud of the negative reviews by using the package “wordcloud”.

```
# Word Cloud :
w <- sort(rowSums(tdm_negative), decreasing = TRUE) # Sort words in decreasing order.
set.seed(123)
wordcloud(words = names(w), freq = w,
          max.words = 250,random.order = F,
          min.freq =  3,
          colors = brewer.pal(8, 'Dark2'),
          scale = c(5,0.3),
          rot.per = 0.6)
```

Output :

Creating the wordCloud using the package “wordcloud2”. Below command visualize wordCloud2.

```
library(wordcloud2)
w <- data.frame(names(w),w)
colnames(w) <- c('word','freq')
wordcloud2(w,size = 0.5, shape = 'triangle', rotateRatio = 0.5, minSize = 2)
```

Output :

Now, analysing the sentiment of 5 reviews from the consumer reviews.

```
#sentiment analysis - Emotion

library(syuzhet)

sentence <- "Warm, friendly, courteous & helpful nature of the person. Listening with intent to understand, help & resolve issue customer was facing
            problem sorted."


sentence2 <- "INCREASE NUMBER OF OUTLETS WHERE CARD IS ENTERTAINED"
sentence3 <- "THE PERSON TALKED TO ME VERY RESPECTFULLY AND TO THE POINT. HE WAS VERY RESPONSIVE OF MY QUERRIES."
sentence4 <- "Better response from customer service is excepted."
sentence5 <- "He was helpful and did tell me the best way to use my points."
```

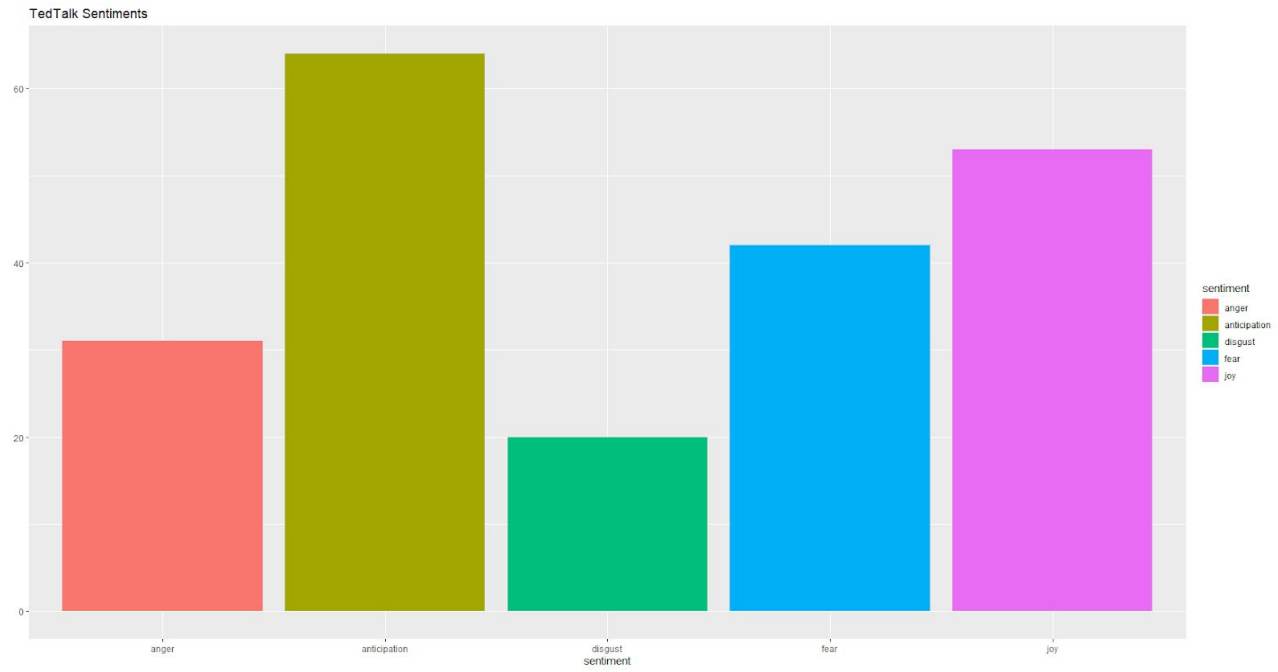
<u>Output</u> :

Sentiment of 5 reviews are printed with the help of the function get_nrc_sentiment.

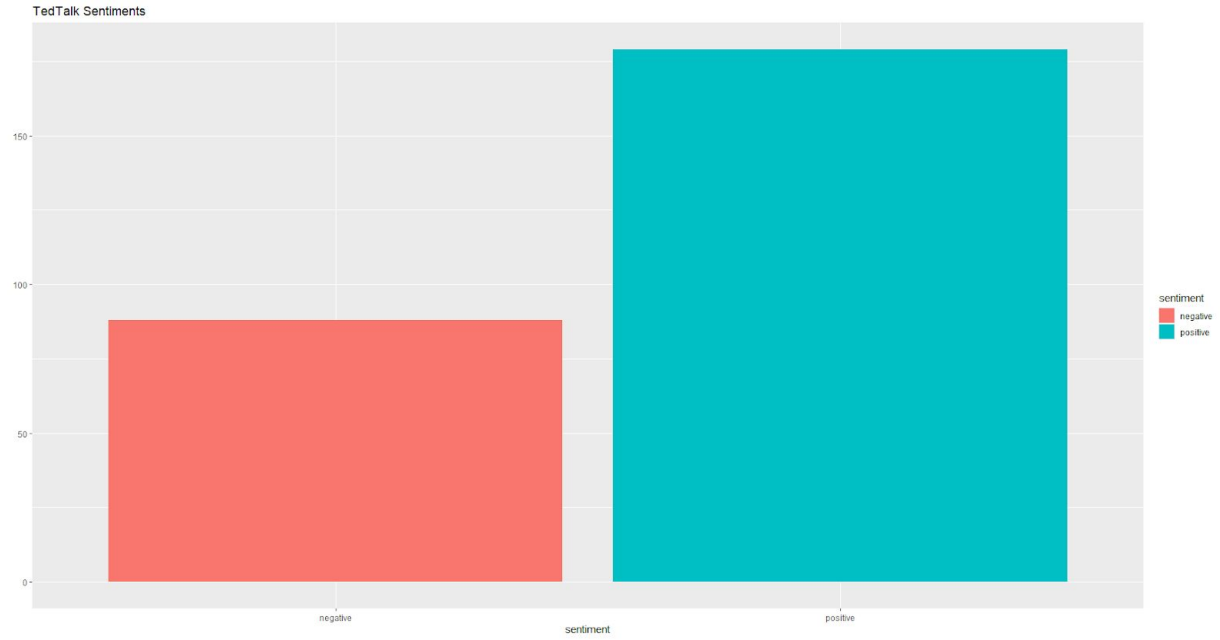
```
> get_nrc_sentiment(new_sentence)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            1       0    1   2       1        0     2        1        4
> new_sentence <- as.character(strsplit(sentence2, " "))
> get_nrc_sentiment(new_sentence)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            0       0    0   1       0        0     0        0        2
> new_sentence <- as.character(strsplit(sentence3, " "))
> get_nrc_sentiment(new_sentence)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            1       0    0   0       0        0     1        0        1
> new_sentence <- as.character(strsplit(sentence4, " "))
> get_nrc_sentiment(new_sentence)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            0       0    0   0       0        0     0        0        1
> new_sentence <- as.character(strsplit(sentence5, " "))
> get_nrc_sentiment(new_sentence)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            0       0    0   1       0        0     1        0        1
>
```

This Code plots the first 5 reviews sentiment. And also plots positive and negative plot.

```
#plot the first 5 rows,the distinct emotions
qplot(sentiment, data=new_result[1:5,], weight=count, geom="bar",fill=sentiment)+ggtitle("TedTalk Sentiments")

#------------------------------+------------------------

#plot the last 2 rows ,positive and negative
qplot(sentiment, data=new_result[9:10,], weight=count, geom="bar",fill=sentiment)+ggtitle("TedTalk Sentiments")
```



Positive and negative sentiment of 5 reviews.



CONCLUSION

From the above sentiment analysis, it is understood that positive reviews having the highest portion followed by neutral reviews and negative reviews. It seems that consumers are happy with the product and seems to be positive. Bar plot is indicating that trust has second highest score given by the sentiment score when plotted. It seems that consumers are having the trust on the product. As well as anticipation is also found while analysis.

In overall, card, customer and call are most frequently repeated words which found by plotting the barplot of the most frequent repeated words. These words have the strong correlation between them which has strong key role in the analysis. Negative reviews constitutes 16 percent in overall whereas the positive reviews constitutes 59.7 percent. Rest of the portion is neutral reviews.

In positive reviews, customer, call and service have repeated highest number of times among all the positive reviews. In negative reviews, card and problem has repeated highest number of times among negative reviews.

To sum up, it is understood that consumers has the trust and anticipation for the product which made them give positive reviews on overall even though it has some negative reviews but positivity surpassed the negative reviews. Finally, it is understood that majority of the consumers are happy about the product.

REFERENCES

Arnold, Taylor B. 2016. *cleanNLP: A Tidy Data Model for Natural Language Processing*. https://cran.r-project.org/package=cleanNLP.

Loughran, Tim, and Bill McDonald. 2011. “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks.” *The Journal of Finance* 66 (1). Blackwell Publishing Inc: 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x.

Mullen, Lincoln. 2016. *tokenizers: A Consistent Interface to Tokenize Natural Language Text*. https://CRAN.R-project.org/package=tokenizers.

Wickham, Hadley. 2007. “Reshaping Data with the reshape Package.” *Journal of Statistical Software* 21 (12): 1–20. http://www.jstatsoft.org/v21/i12/.