515k HOTEL REVIEW DATA IN EUROPE

GROUP : 9

Akash Gupta
Dharani Nalabolu
Deeksha Sudhakar
Chinmay Karandikar
Hardil trivedi

INDEX

ABSTRACT

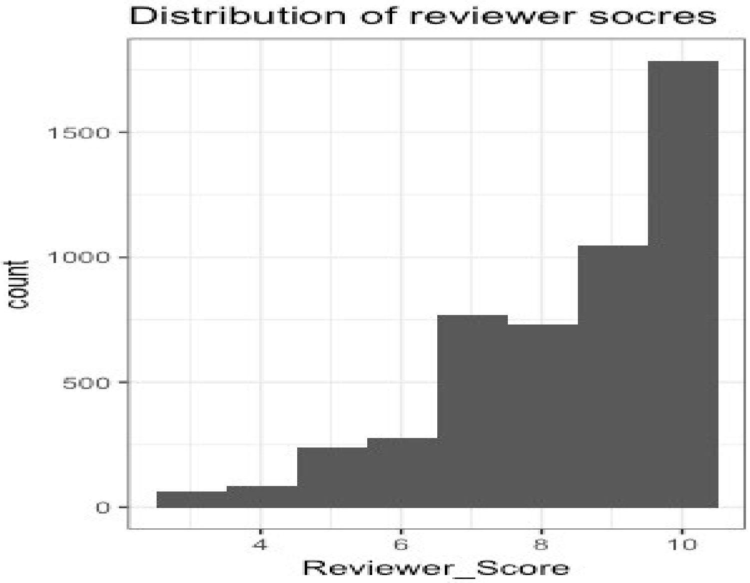
The purpose of the report is to find positivity and negativity of the different reviews given by the customers for numerous hotels within Europe. Our data set includes 16 columns including the latitude and longitude of the respective hotels which allowed us to plot the hotels in google maps graphic. Text Mining techniques are used to perform the sentimental analysis and create a analysis from the reviews provided. Text cleaning and various algorithms are used to create a meaningful output from the data provided in the dataset.

1. INTRODUCTION

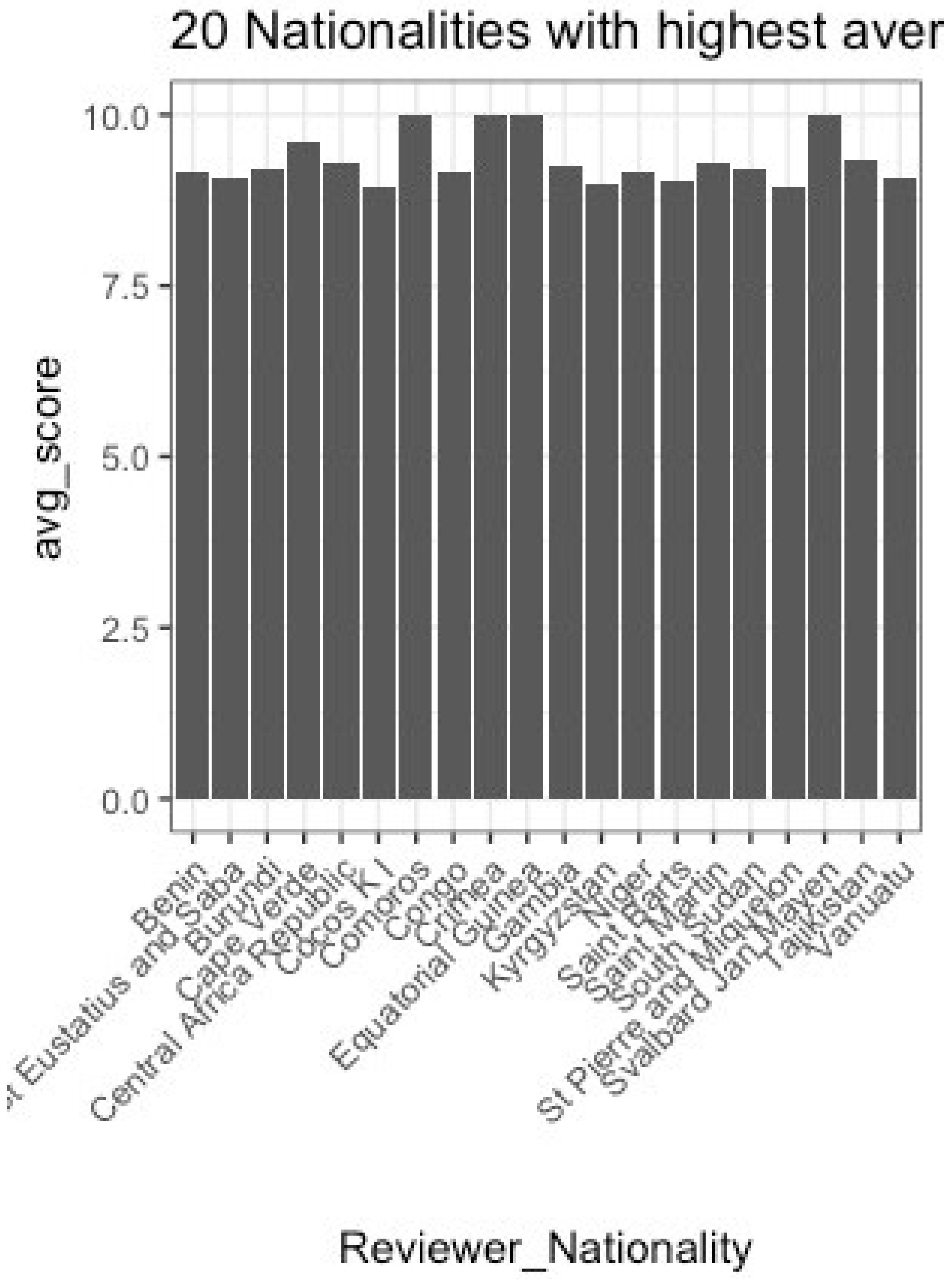
The data was scraped. Data is originally owned by Booking.comBooking.com is a travel fare aggregator website for lodging reservations. It is owned and operated by and is the primary revenue source of United States-based Booking Holdings. Booking.com is headquartered in Amsterdam. The website has more than 29,094,365 listings in 143,172 destinations in 230 countries and territories worldwide. Each day more than 1,550,000 room nights are reserved on the website. The site is available in 43 languages

1.1.EDA AND DATA CLEANING

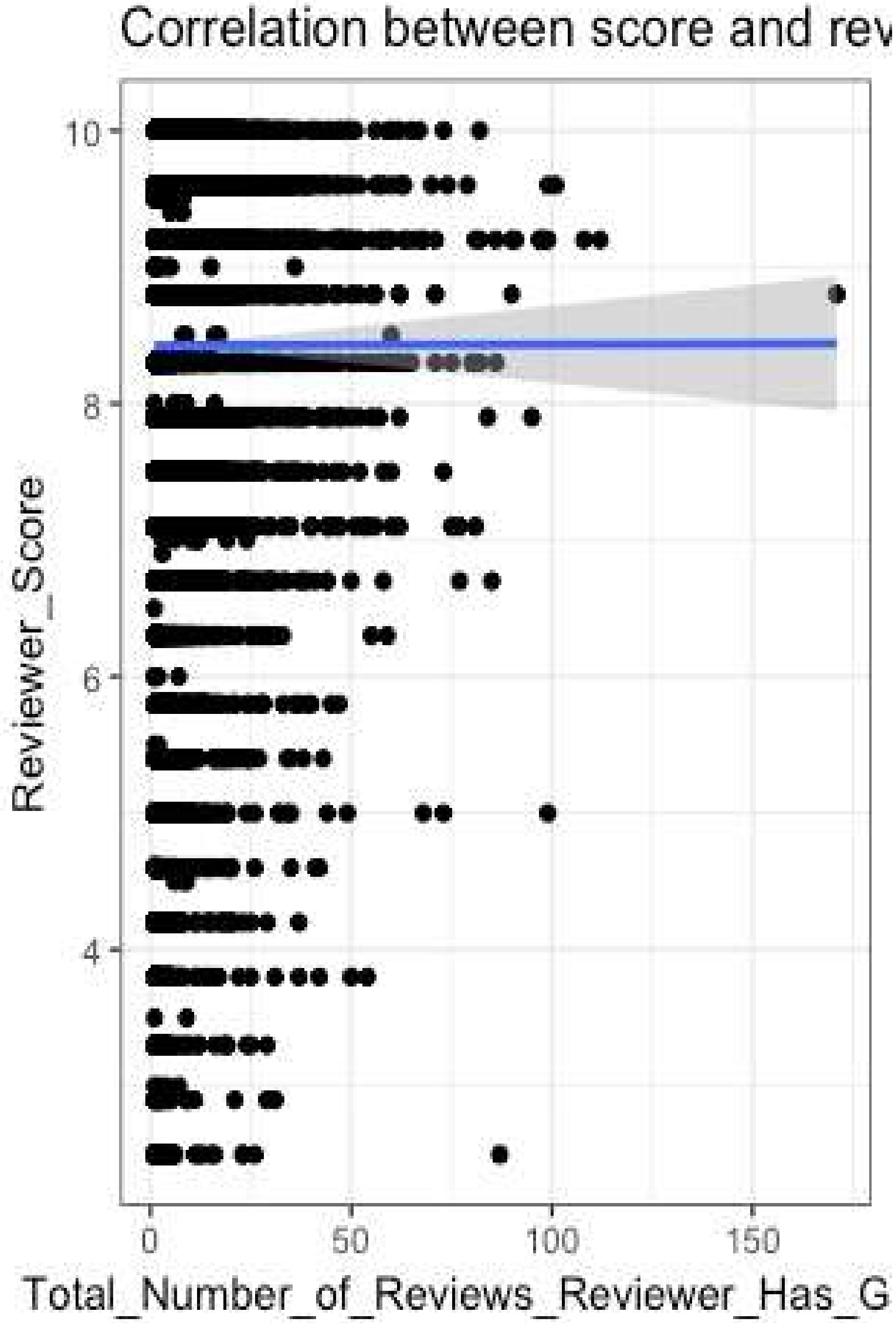
- In this section we describe the dataset and provide the insights got from exploratory analysis of the respective data set. We explore the data in order to identify distribution in data patterns, features and biases etc.
- It is important to really understand what each field tells us and how it is related to the outcome. In the following sections, we will list the different fields with the highest importance and their relationship with different variables.



Scoring of the reviewer
The above plot represents the relation between Reviewer score and the count on the scale of 0 to 1500.



The plot above represents the reviewer nationality which represents the 20 countries with the highest average score .



The plot represents the correlation between the total number of reviews the reviewer has given and the reviewer score.

DATA CLEANING

- We used the library **Dplyr** for manipulating the datasets, library **corpus** to make a world cloud from the reviews and eliminate all the words that are not required for analysis by the output from the word cloud to perform sentimental analysis on the data.
- We have used the xtabs to create the contingency table between two columns in the dataset to study the relationship between two categorical variables
- We have created normalized datasets to be used by algorithms like KNN which assume multi-dimensional normal data.

2. DATA COLLECTION PROCESS

We have used the dataset from Kaggle. This dataset contains 515,000 customer reviews and scoring of 1493 luxury hotels across Europe.

The csv file contains 17 fields. The description of each field is as below:

- Hotel_Address: Address of hotel.
- Review_Date: Date when reviewer posted the corresponding review.
- Average_Score: Average Score of the hotel, calculated based on the latest comment in the last year.
- Hotel_Name: Name of Hotel
- Reviewer_Nationality: Nationality of Reviewer
- Negative_Review: Negative Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Negative'
- Review_Total_Negative_Word_Counts: Total number of words in the negative review.
- Positive_Review: Positive Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Positive'
- Review_Total_Positive_Word_Counts: Total number of words in the positive review.
- Reviewer_Score: Score the reviewer has given to the hotel, based on his/her experience
- Total_Number_of_Reviews_Reviewer_Has_Given: Number of Reviews the reviewers has given in the past.
- Total_Number_of_Reviews: Total number of valid reviews the hotel has.
- Tags: Tags reviewer gave the hotel.
- days_since_review: Duration between the review date and scrape date.
- Additional_Number_of_Scoring: There are also some guests who just made a scoring on the service rather than a review. This number indicates how many valid scores without review in there.
- lat: Latitude of the hotel
- lng: longtitude of the hotel

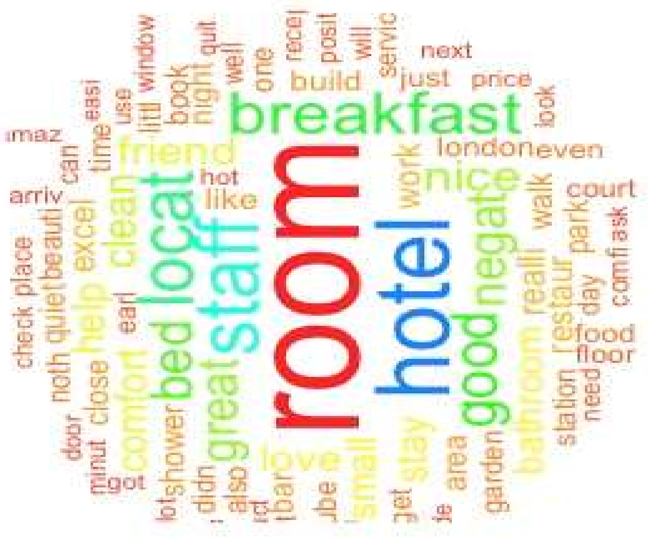
3. ALGORITHMS

In this section we briefly describe the about the algorithms we are going to use for this Project and discuss about outcomes for each algorithm, advantages, disadavntages and how well are the applied models fitting the data. The algorithms we used on the dataset are explained below.

3.1 Word cloud

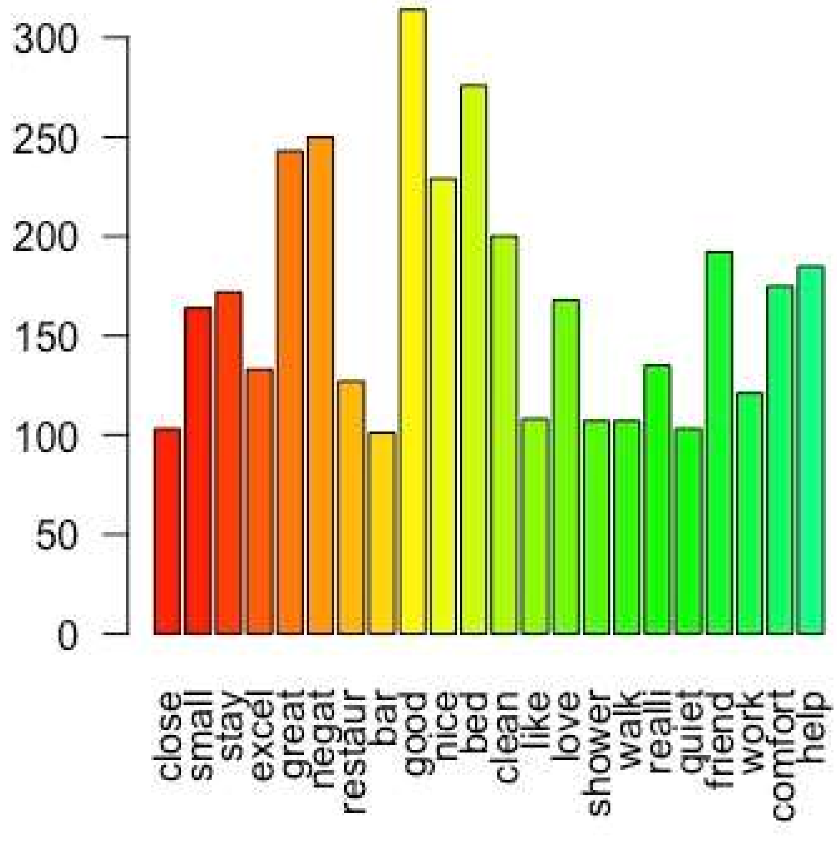
One can create a word cloud, also referred as text cloud or tag cloud, which is a visual representation of text data. The procedure of creating word clouds is very simple in R if you know the different steps to execute. The text mining package (*tm*) and the word cloud generator package are available in R for helping us to analyze texts and to quickly visualize the keywords as a word cloud.

Word clouds add simplicity and clarity. The most used keywords stand out better in a word cloud. Word clouds are a potent communication tool. They are easy to understand, to be shared and are impactful. Word clouds are visually engaging than a table data.



The world cloud represented above depicts the review which are positive and negative after cleaning the data with reviews the word with largest size represents that its frequency of occurrence is higher. corpus function is used to clean the data and then a stem document matrix is made after cleaning the reviews using the library **snowballC** .

Term document matrix is created and inspected, and the frequencies of the words are calculated and arranged in the descending order. Then the library **wordcloud2** is used to create a word cloud. The library **iconv** is used to convert the term document matrix so as to perform the sentimental analysis on the words attained in the word cloud.

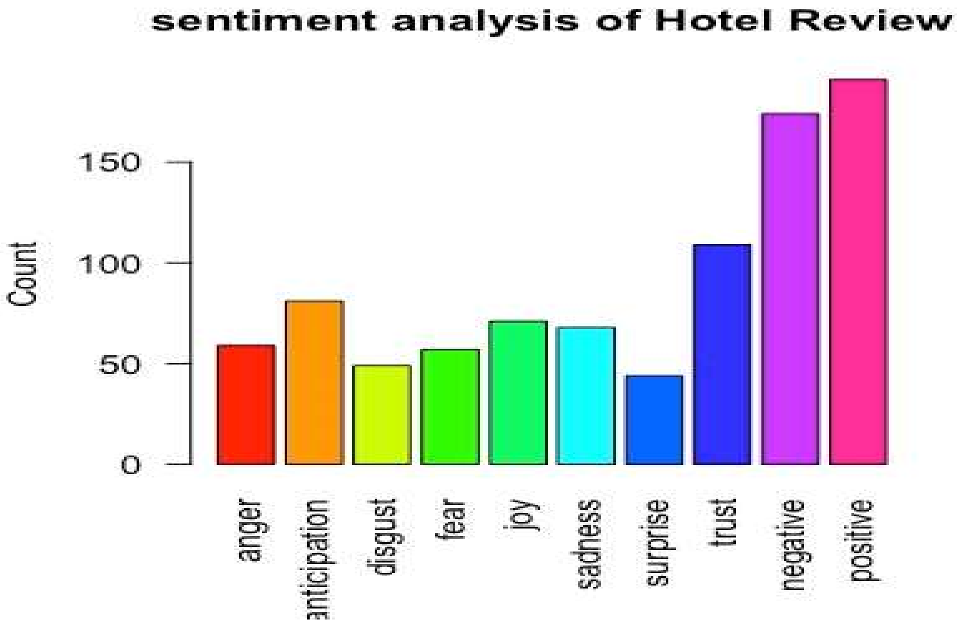


The above plot showcases different words used by the reviewer in the review and and their frequencies of occurrence

Disadvantage: Although the Word Cloud is designed to make words stand out according to their size based on their frequency of occurrence, there are other factors can affect the visual decoding of the data. The length of the word and the white space around the letter can make it look more or less important relative to others in the cloud. This can mislead the interpretation.

3.2 Sentimental Analysis

It's the process of analyzing online pieces of writing to determine the emotional tone they carry. In simple words, sentiment analysis is used to find one's attitude towards something. Some tools categorize pieces of writing as positive, neutral, or negative. The science behind sentiment analysis is based on algorithms using natural language processing to categorize pieces of writing as positive, neutral, or negative. The algorithm is designed to identify positive and negative words, such as "fantastic", "beautiful", "disappointing", "terrible", etc.

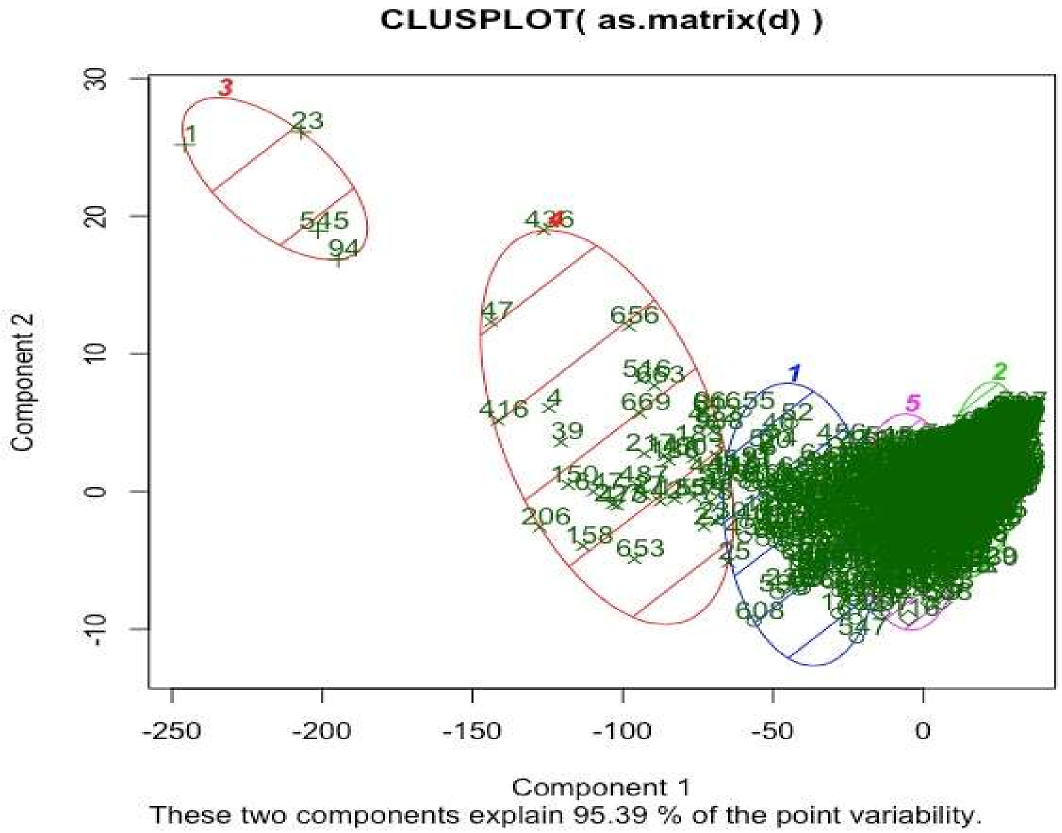


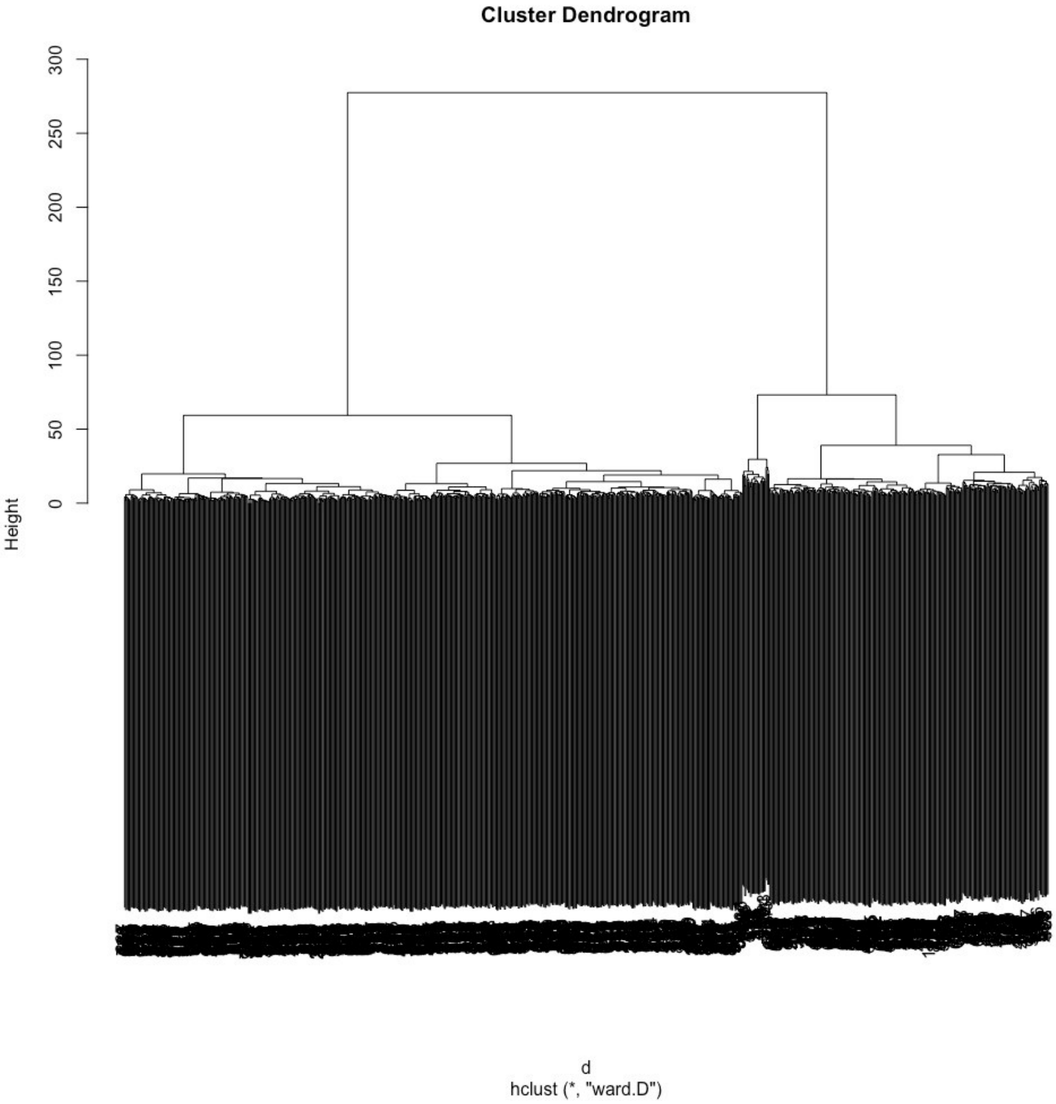
After the analysis we found out that our dataset had overall positive reviews . We also took into consideration the top 50 words depending on their counts and their sentiment value. The lexicon used in sentimental Analysis is get_nrc_sentiments which provided the outcome of the plot. The plot depicts that the positive reviews are more than the negative reviews. There are various types of emotions in which anger, anticipation, disgust,fear,sadness constitute towards the negativity of the review while joy, surprise, trust towards the positivity of the review.

Disadvantage: It is important to look at the data from the standpoint of time because sentiment changes over time; from the reviews in the database there are reviews which are very old and might contribute a negative perspective .

3.3 K-NN ALGORITHM

The idea in k -nearest-neighbors methods is to identify k records in the training dataset that are similar to a new record that we wish to classify. We then use these similar (neighboring) records to classify the new record into a class, assigning the new record to the predominant class among these neighbors. Denote the values of the predictors for this new record by x 1 , x 2 , . . . , x p . We look for records in our training data that are similar or "near" the record to be classified in the predictor space (i.e., records that have values close to x 1 , x 2 , . . . , x p). Then, based on the classes to which those proximate records belong, we assign a class to the record that we want to classify. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour.

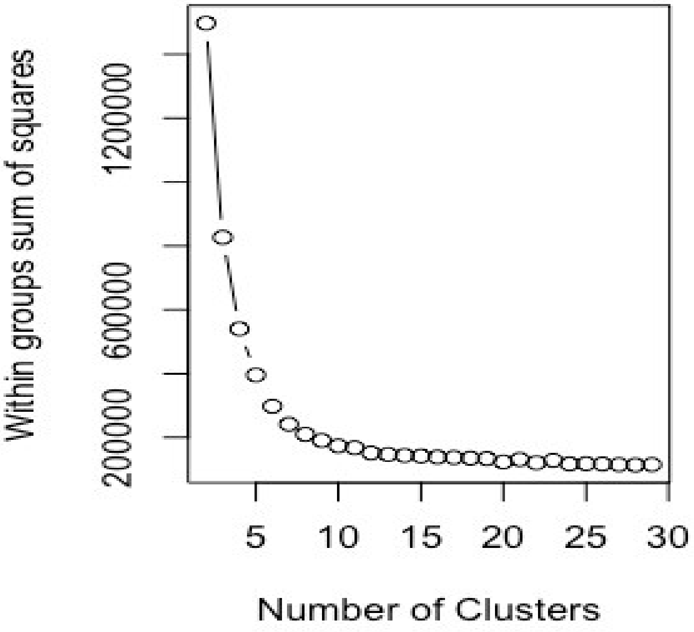




Disadvantage: In KNN the dimensions can be inter related and instead assumes they are independent (as it’s just calculating distance) and also if the data is not normalized distance can be biased towards a specific dimension.

3.4 CLUSTERING

K-means clustering is the most popular partitioning method. It requires to specify the number of clusters to extract. A plot of the within groups sum of squares by number of clusters extracted can help determine the appropriate number of clusters. There are a wide range of hierarchical clustering approaches. In k means clustering, we have to specify the number of clusters we want the data to be grouped into. The algorithm randomly assigns each observation to a cluster and finds the centroid of each cluster. Then, the algorithm iterates through Reassigning data points to the cluster whose centroid is closest and calculating new centroid of each cluster.These two steps are repeated till the within cluster variation cannot be reduced any further. The within cluster variation is calculated as the sum of the euclidean distance between the data points and their respective cluster centroids.



The within group sum of squares **wss (**within group variability) is used to obtain the scree plot above. It gives overview all possible clusters and within group sum of squares. We need to reduce within cluster variables as we can see in the plot the drop between clusters is very large. This scree plot explains that we need to less number of clusters because beyond that the gains are not very significant

Disadvantages: It assumes prior knowledge of the data and requires choosing the appropriate number of clusters in advance. The final results obtained is sensitive to the initial random selection of cluster centers because, for every different run of the algorithm on the same dataset, you may choose different set of initial centers. This may lead to different clustering results on different runs of the algorithm. It’s sensitive to outliers. If the data is rearranged, possible that there is a different solution every time there is a change in the ordering of data.

4. ANALYSIS

- Location of the Hotel is found out using the Demographic data which includes latitudes longitudes and Hotel Address.
- From the reviewer nationality and the average score of reviews the countries are found out in which people gave the most reviews.

From the correlation plot:

- With the increase in the total number of reviews, naturally the additional number of scoring increases. It shows a high positive correlation.
- The Average Score and Reviewers score are positivelyy related. However, the correlation is not conclusive enough. It shows a significantly low level of positive correlation which is indicative from the dataset.
- Total Negative Word Counts and the ReviewersScore are inversely proportional to some degree as they are negatively correlated with low correlation coefficient.
- Total Reviews reviewers have given and days since review do not show significant correlation with any other attributes. Their correlation coefficient is almost zero.
- The inferences suggest that only few attributes are correlated whereas others are not dependent on each other.

From Sentimental Analysis:

- There are more Positive Reviews than the Negative reviews according to the reviews data provided in the dataset.
- The positive word count is more which constitutes more to the positive reviews
- As the data is large to understand the clustering scree plot is used which depicts that less number of clusters are required to if not the using more number of cluster makes the gains not significant

5. RECOMMENDATIONS

- Collecting more demographic data will help in locating more countires inside the continent
- The data can be collected when a person is checking in through his e-mail or using his card which provides the information about the person.
- Using the reviews not more than a month helps in increasing the positive rating.
- Avoiding fake reviews filtering them can help in providing genuine reviews about the hotel and its environment.

6. SITES AND REFERENCES

Project:

https://www.kaggle.com/515k-hotel-reviews-data-in-europe/home

Book:

Data Mining for Business Analytics: Concepts, Techniques, and Applications with XLMiner 3rd Edition.

Additional Reference :

https://stackoverflow.com

7. **R-CODE**

```
hotel<-Hotel_Reviews
library(NLP)
library(caret)
library(tidyverse)
library(tidytext)
library(plyr)
library(dplyr)
library(sentimentr)
library(SnowballC)
library(tm)
library(RColorBrewer)
library(ROAuth)
library(wordcloud)
library(corpus)
library(ggrepel)
theme_set(theme_classic())

hotel.df <- hotel
str(hotel.df)
hotel.df <- hotel.df[1:5000, ]
hotel.df$reviews        =        paste        (hotel.df$Negative_Review,
hotel.df$Positive_Review)
set.seed(1207)
index <- 1:nrow(hotel.df)

# partition data into training and validation
training.index <- sample(index,trunc(length(index)*0.8))
training.df <- hotel.df$reviews[training.index]
validation.df <- hotel.df$reviews[-training.index]
---------------------------------------------
  #correaltion plot
  library(corrplot)
M <- cor(f[c(2,4,8,9,11,12,13,15)])
corrplot(M, method = "circle")

  # scoring or reviewer
 g       <-       ggplot(hotel.df[sample(nrow(hotel.df),       5000),
],aes(x=Reviewer_Score))       +       geom_histogram(binwidth       =
1)+theme_bw()+ggtitle('Distribution of reviewer socres')
plot(g)

avgscore_nation       <-       sqldf('SELECT       Reviewer_Nationality,
avg(Reviewer_Score) as avg_score from hotel group by Reviewer_Nationality
order by avg(Reviewer_Score) desc')
avgscore_nation[166,1]<-'UnKnown'
g                                                                      <-
ggplot(avgscore_nation[1:20,],aes(x=Reviewer_Nationality,y=avg_score))
+   geom_bar(stat   =   'identity')+theme_bw()+   theme(axis.text.x   =
element_text(angle = 45, hjust = 1)) + ggtitle('20 Nationalities with
highest average score')

g                                                                      <-
ggplot(avgscore_nation[207:227,],aes(x=Reviewer_Nationality,y=avg_score
))  +  geom_bar(stat  =  'identity')+theme_bw()+  theme(axis.text.x  =
element_text(angle = 45, hjust = 1)) + ggtitle('20 Nationalities with
lowest average score')
plot(g)

g        <-        ggplot(hotel.df[sample(nrow(hotel.df),        10000),
],aes(x=Total_Number_of_Reviews_Reviewer_Has_Given,y=Reviewer_Score))  +
geom_point()+theme_bw()+geom_smooth(method = "lm")+ggtitle('Correlation
between score and review frequency')
plot(g)


View(hotel.df)
#google map

summary(hotel.df)
library(leaflet)
library(leaflet.extras)
library(tidyr)
library(scales)
library(ggplot2)

hotel.names = hotel.df %>%
  select(Hotel_Name,    Hotel_Address,    lat,    lng,    Average_Score,
Total_Number_of_Reviews,
         Review_Total_Positive_Word_Counts,
Review_Total_Negative_Word_Counts) %>%
  #Remove the 17 records without geo coordinates
  filter(lat != 0 & lng != 0) %>%
  group_by(Hotel_Name,   Hotel_Address,   lat,   lng,   Average_Score,
Total_Number_of_Reviews) %>%
  summarise(Tot_Pos_Words = sum(Review_Total_Positive_Word_Counts),
            Tot_Neg_Words = sum(Review_Total_Negative_Word_Counts),
            Total_Words = sum(Tot_Pos_Words + Tot_Neg_Words),
            Pos_Word_Rate = percent(Tot_Pos_Words/Total_Words),
            Neg_Word_Rate = percent(Tot_Neg_Words/Total_Words))
points <- cbind(hotel.names$lng,hotel.names$lat)
leaflet() %>%
  addProviderTiles('OpenStreetMap.Mapnik',
                   options = providerTileOptions(noWrap = TRUE)) %>%
  addMarkers(data = points,
             popup = paste0("<strong>Hotel: </strong>",
                            hotel.names$Hotel_Name,
                            "<br><strong>Address: </strong>",
                            hotel.names$Hotel_Address,
                            "<br><strong>Average Score: </strong>",
                            hotel.names$Average_Score,
                            "<br><strong>Number of Reviews: </strong>",
                            hotel.names$Total_Number_of_Reviews,
                            "<br><strong>Percent Positive Review Words:
</strong>",
                            hotel.names$Pos_Word_Rate),
             clusterOptions = markerClusterOptions())

colSums(sapply(hotel.df, is.na))
hotel.df%>%select(Average_Score,Hotel_Address)%>%distinct(Average_Score
,Hotel_Address)%>%ggplot(aes(x=Average_Score))+
  geom_histogram(color='blue',fill='blue',alpha=0.3,bins=30)+
  xlab("Average Review Score")+ylab("Counts")
```

```
#introduce corpus
corpus<-Corpus(VectorSource(hotel.df$reviews))
#text cleaning
View(corpus)
#convert the text to lower case
corpus <- tm_map(corpus,content_transformer(tolower))
inspect(corpus[1:20])
#remove nuymbers
corpus<- tm_map(corpus,removeNumbers)
inspect(corpus[1:20])
#remove english common stopwords
corpus<- tm_map(corpus,removeWords,stopwords("english"))
inspect(corpus[1:20])
#remove punctuation
corpus<-tm_map(corpus,removePunctuation)
inspect(corpus[1:20])
#remove extra whitespaces
corpus<-tm_map(corpus,stripWhitespace)
inspect(corpus[1:20])


#load library
library(SnowballC)
#Stem document
corpus <- tm_map(corpus,stemDocument)
writeLines(as.character(corpus[[30]]))
dtm <- DocumentTermMatrix(corpus)
dtm
inspect(dtm[1:2,1000:1005])
freq <- colSums(as.matrix(dtm))
length(freq)
ord <- order(freq,decreasing=TRUE)
freq[head(ord)]
#inspect least frequently occurring terms
freq[tail(ord)]
dtmr <-DocumentTermMatrix(corpus, control=list(wordLengths=c(4, 20),
                                               bounds = list(global =
c(3,27))))
dtmr
freqr <- colSums(as.matrix(dtmr))
#length should be total number of terms
length(freqr)
#create sort order (asc)
ordr <- order(freqr,decreasing=TRUE)
#inspect most frequently occurring terms
freqr[head(ordr)]
#inspect least frequently occurring terms
freqr[tail(ordr)]
findFreqTerms(dtmr,lowfreq=20)
wf=data.frame(term=names(freqr),occurrences=freqr)

# worldcloud
wordcloud(names(freq), freq = freq, max.words=1000,
          random.order = FALSE,
          min.freq =50,colors= rainbow(50),
          rot.per= 0.7)
```

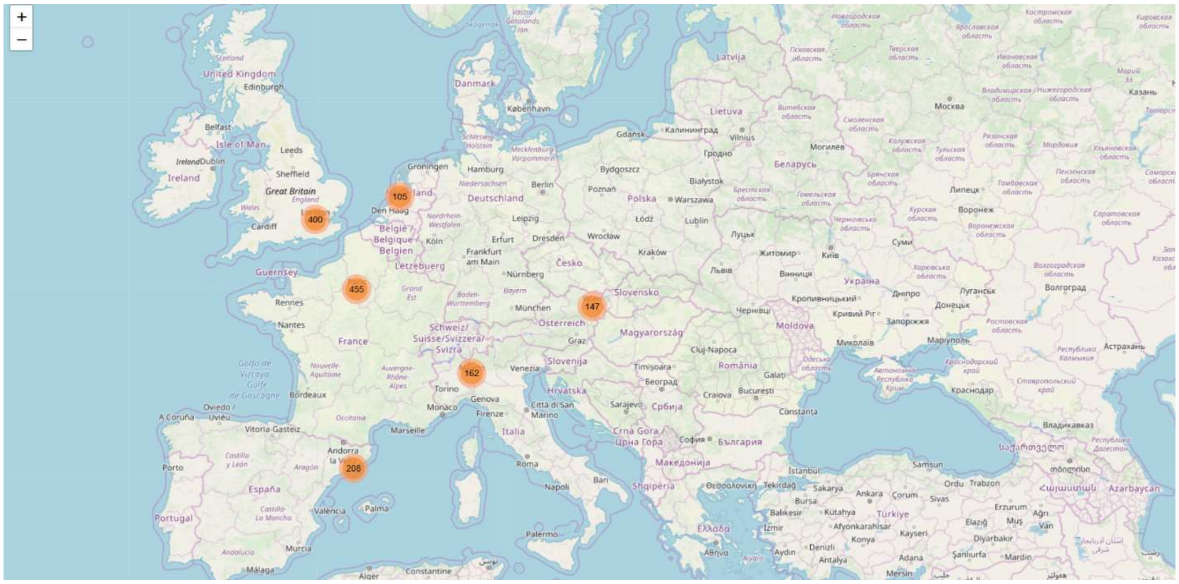
```
library(ggplot2)
p <- ggplot(subset(wf, freqr>20), aes(term, occurrences))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))
p


------------------------------------------------------------------------
---------------------------------

  #sentimental analysis
  library(syuzhet)
library(lubridate)
review <- iconv (corpus)
s <- get_nrc_sentiment(review)
barplot(colSums(s), las=2, col= rainbow(10), ylab= 'Count', main =
'sentiment analysis of Hotel Review')
------------------------------------------------------------------------
---------------------------------------
  #clustering
  v <- as.matrix(dtm)
d<- dist(v)
#run hierarchical clustering using Ward,Äôs method
groups <- hclust(d,method="ward.D")
#plot dendogram, use hang to ensure that labels fall below tree
plot(groups, hang=1)
rect.hclust(groups,2)
#k means algorithm, 2 clusters, 100 starting configurations
kfit <- kmeans(d, 10, nstart=100)
#plot ,Äì need library cluster
library(cluster)
clusplot(as.matrix(d), kfit$cluster, color=T, shade=T, labels=2,
lines=0)
#kmeans ,Äì determine the optimum number of clusters (elbow method)
#look for ,Äúelbow,Äù in plot of summed intra-cluster distances
(withinss) as fn of k
wss <- 2:29
for (i in 2:29) wss[i] <- sum(kmeans(d,centers=i,nstart=25)$withinss)
plot(2:29, wss[2:29], type="b", xlab="Number of Clusters",ylab="Within
groups sum of squares")
```

8. REPORT &RESULT

```
#google map

summary(hotel.df)
library(leaflet)
library(leaflet.extras)
library(tidyr)
library(scales)
library(ggplot2)

hotel.names = hotel.df %>%
  select(Hotel_Name, Hotel_Address, lat, lng, Average_Score,
Total_Number_of_Reviews,
         Review_Total_Positive_Word_Counts,
Review_Total_Negative_Word_Counts) %>%
  #Remove the 17 records without geo coordinates
  filter(lat != 0 & lng != 0) %>%
  group_by(Hotel_Name, Hotel_Address, lat, lng, Average_Score,
Total_Number_of_Reviews) %>%
  summarise(Tot_Pos_Words = sum(Review_Total_Positive_Word_Counts),
            Tot_Neg_Words = sum(Review_Total_Negative_Word_Counts),
            Total_Words = sum(Tot_Pos_Words + Tot_Neg_Words),
            Pos_Word_Rate = percent(Tot_Pos_Words/Total_Words),
            Neg_Word_Rate = percent(Tot_Neg_Words/Total_Words))
points <- cbind(hotel.names$lng,hotel.names$lat)
leaflet() %>%
  addProviderTiles('OpenStreetMap.Mapnik',
                   options = providerTileOptions(noWrap = TRUE)) %>%
  addMarkers(data = points,
             popup = paste0("<strong>Hotel: </strong>",
                            hotel.names$Hotel_Name,
                            "<br><strong>Address: </strong>",
                            hotel.names$Hotel_Address,
                            "<br><strong>Average Score: </strong>",
                            hotel.names$Average_Score,
                            "<br><strong>Number of Reviews: </strong>",
                            hotel.names$Total_Number_of_Reviews,
                            "<br><strong>Percent Positive Review Words:
</strong>",
                            hotel.names$Pos_Word_Rate),
             clusterOptions = markerClusterOptions())

colSums(sapply(hotel.df, is.na))
hotel.df%>%select(Average_Score,Hotel_Address)%>%distinct(Average_Score
,Hotel_Address)%>%ggplot(aes(x=Average_Score))+
  geom_histogram(color='blue',fill='blue',alpha=0.3,bins=30)+
  xlab("Average Review Score")+ylab("Counts")
```

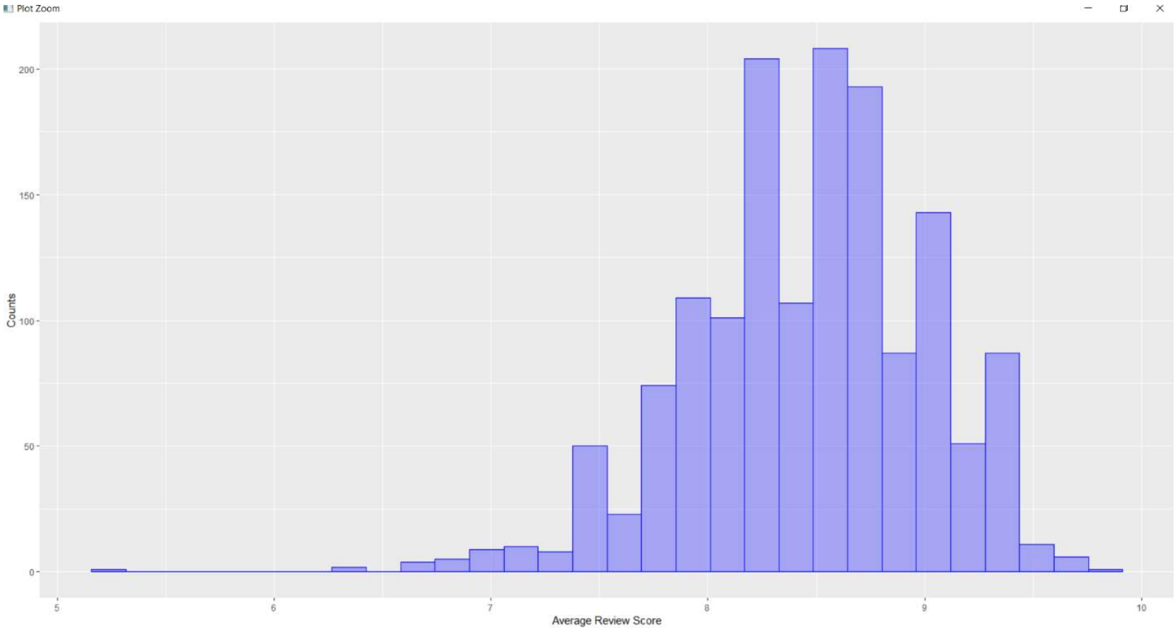
<u>Output and Inferences:</u>

The below Map shows the location of 1477 Hotels that were part of the dataset based on the Latitude and Longitude score.



The count of Hotels in London, Paris and Amsterdam are more for which the reviews have been taken.

A simple graph of Average Review Score and Count is plotted.



Around 8.5 is the Average Review score that the most users have given to most of the hotels.

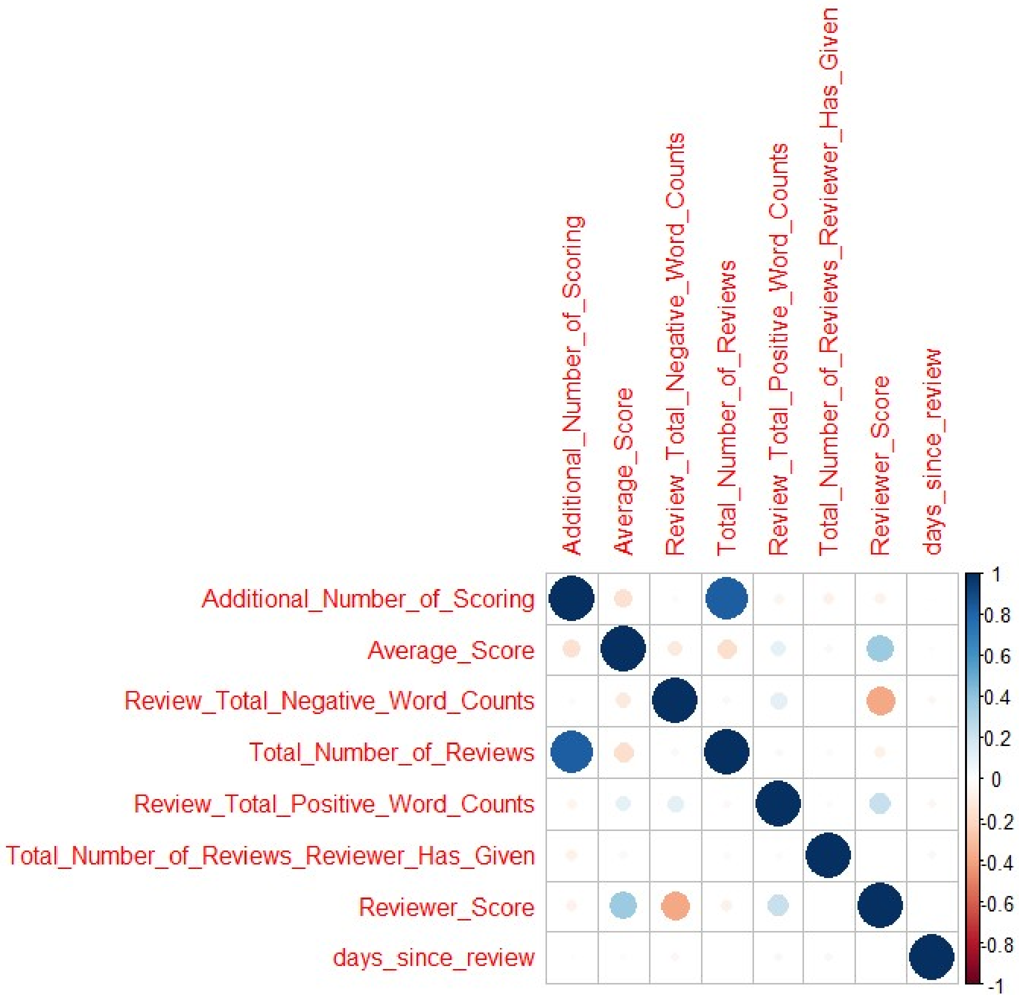
9.4 is the highest score and 6.9 is the least score.

#CORRELATION

Code

```
#correaltion plot
  library(corrplot)
M <- cor(f[c(2,4,8,9,11,12,13,15)])
corrplot(M, method = "circle")
```

Output

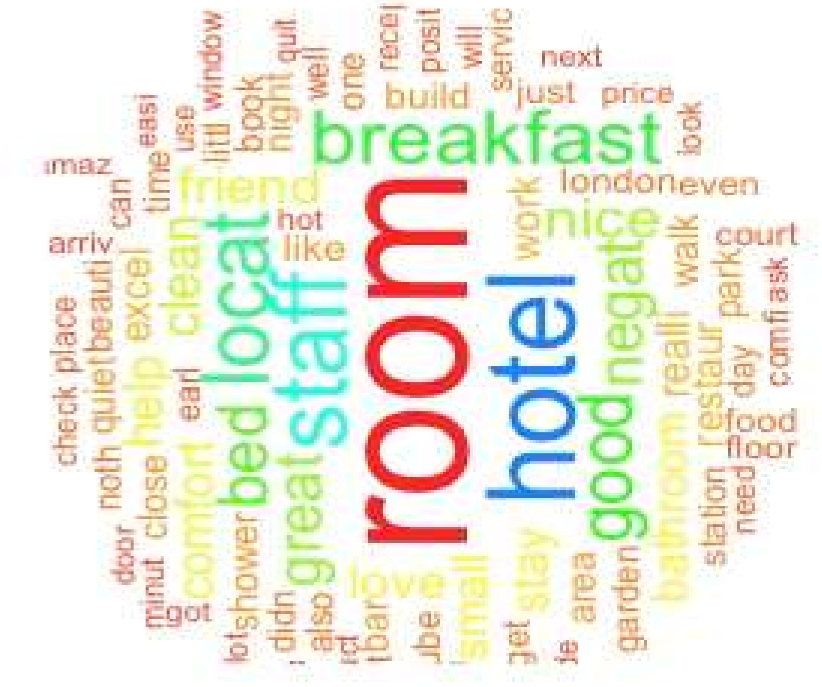


Taken into consideration the following variables for the correlation plot:

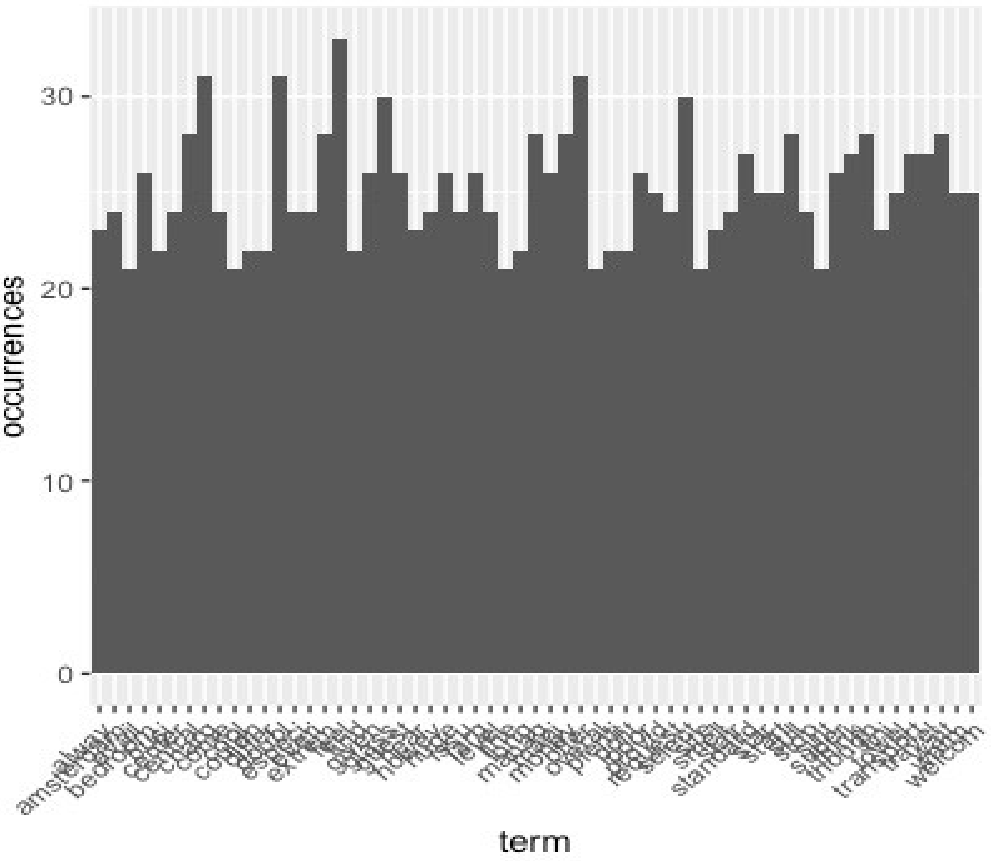
1. Additional_Number_of_Scoring.
2. Average_Score.
3. Review_Total_Negative_Word_Counts.
4. Total_Number_of_Reviews.
5. Review_Total_Positive_Word_Counts.
6. Total_Number_Of_Reviews_Reviewer_Has_Given.
7. Reviewer_Score.
8. Days_since_review.

#WORD CLOUD -& SENTIMENTAL ANALYSIS

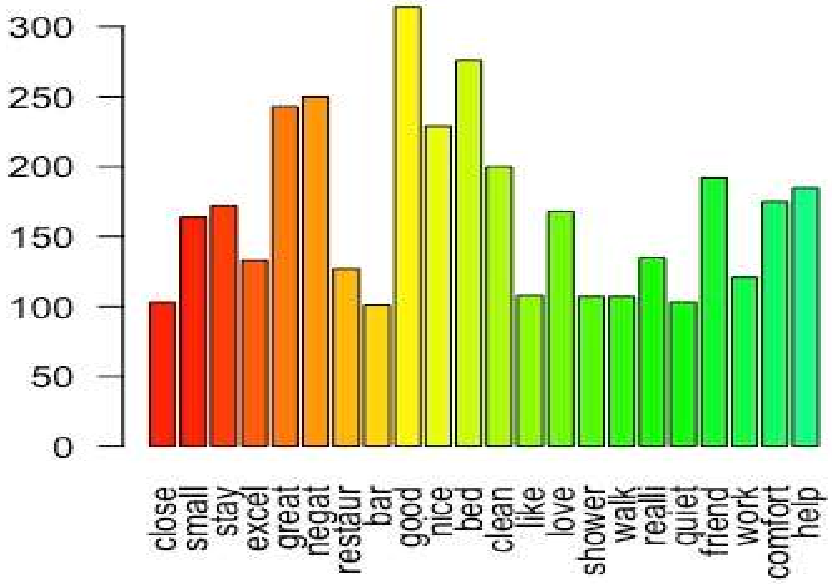
```
#load library
library(SnowballC)
#Stem document
corpus <- tm_map(corpus,stemDocument)
writeLines(as.character(corpus[[30]]))
dtm <- DocumentTermMatrix(corpus)
dtm
inspect(dtm[1:2,1000:1005])
freq <- colSums(as.matrix(dtm))
length(freq)
ord <- order(freq,decreasing=TRUE)
freq[head(ord)]
#inspect least frequently occurring terms
freq[tail(ord)]
dtmr <-DocumentTermMatrix(corpus, control=list(wordLengths=c(4, 20),
                                               bounds  =  list(global  =
c(3,27))))
dtmr
freqr <- colSums(as.matrix(dtmr))
#length should be total number of terms
length(freqr)
#create sort order (asc)
ordr <- order(freqr,decreasing=TRUE)
#inspect most frequently occurring terms
freqr[head(ordr)]
#inspect least frequently occurring terms
freqr[tail(ordr)]
findFreqTerms(dtmr,lowfreq=20)
wf=data.frame(term=names(freqr),occurrences=freqr)

# worldcloud
wordcloud(names(freq), freq = freq, max.words=1000,
          random.order = FALSE,
          min.freq =50,colors= rainbow(50),
          rot.per= 0.7)

library(ggplot2)
p <- ggplot(subset(wf, freqr>20), aes(term, occurrences))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))
p


-----------------------------------------------------------------------
---------------------------------
```



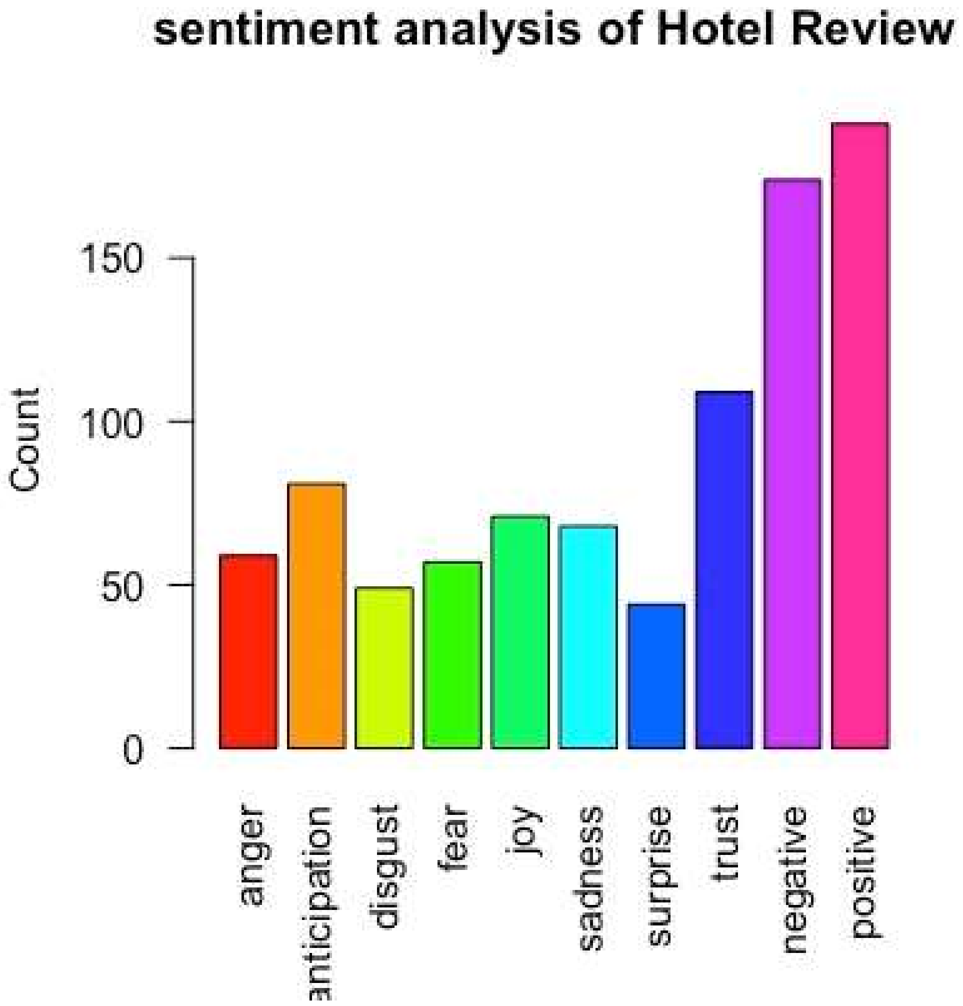
```
#sentimental analysis
  library(syuzhet)
library(lubridate)
review <- iconv (corpus)
s <- get_nrc_sentiment(review)
barplot(colSums(s), las=2, col= rainbow(10), ylab= 'Count', main =
'sentiment analysis of Hotel Review')
```



occurrences
30
20
10
0
term

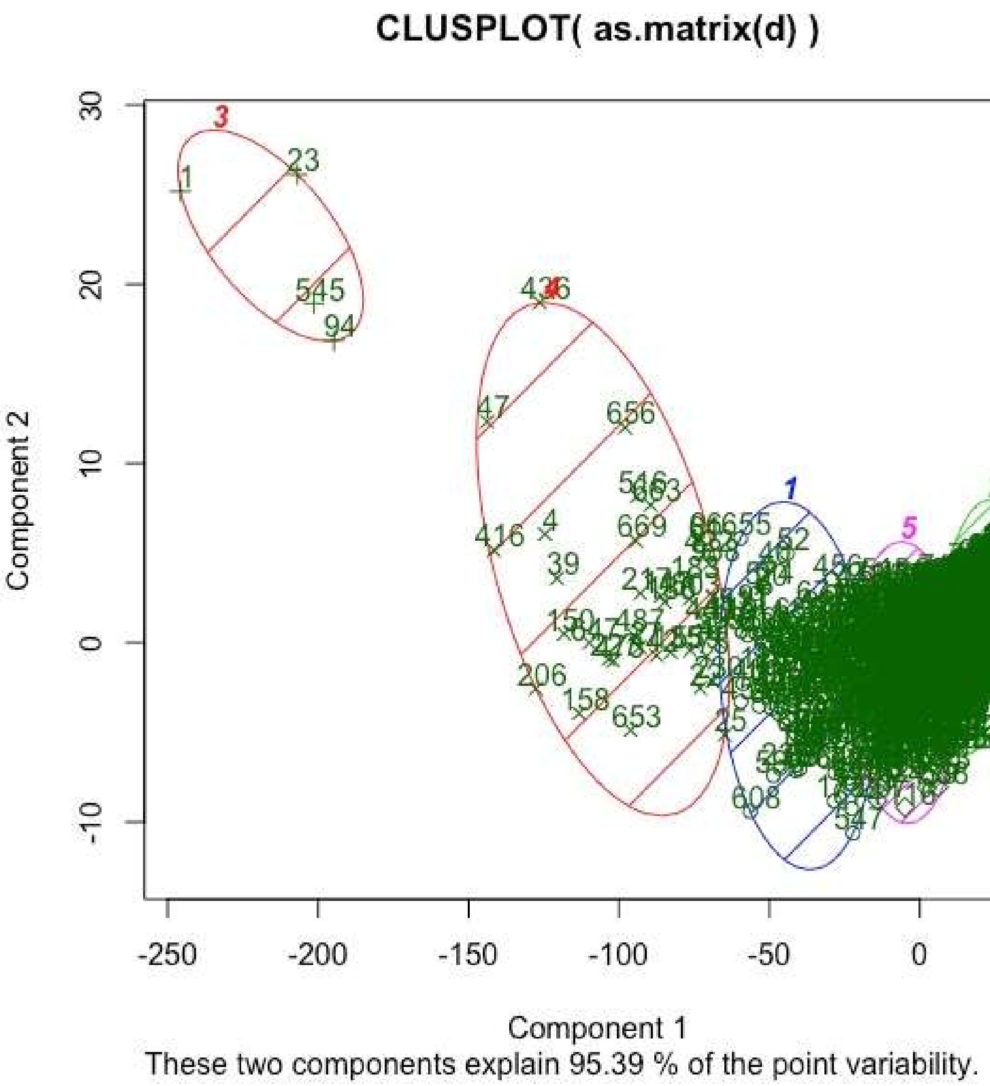


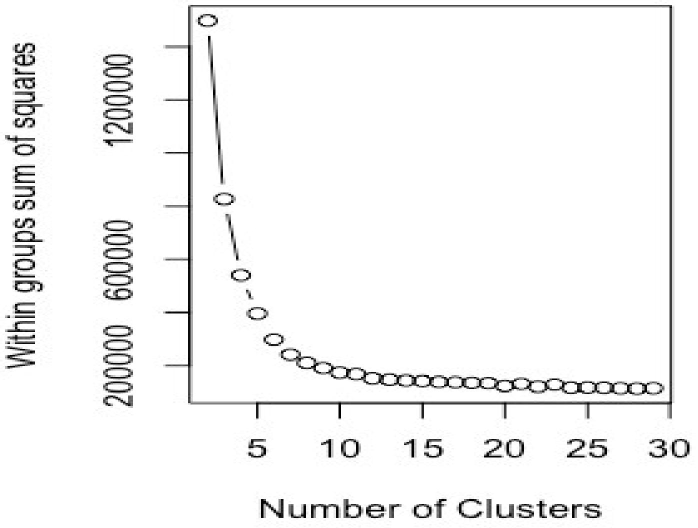
300
250
200
150
100
50
0
close
small
stay
excel
great
negat
restaur
bar
good
nice
bed
clean
like
love
shower
walk
realli
quiet
friend
work
comfort
help

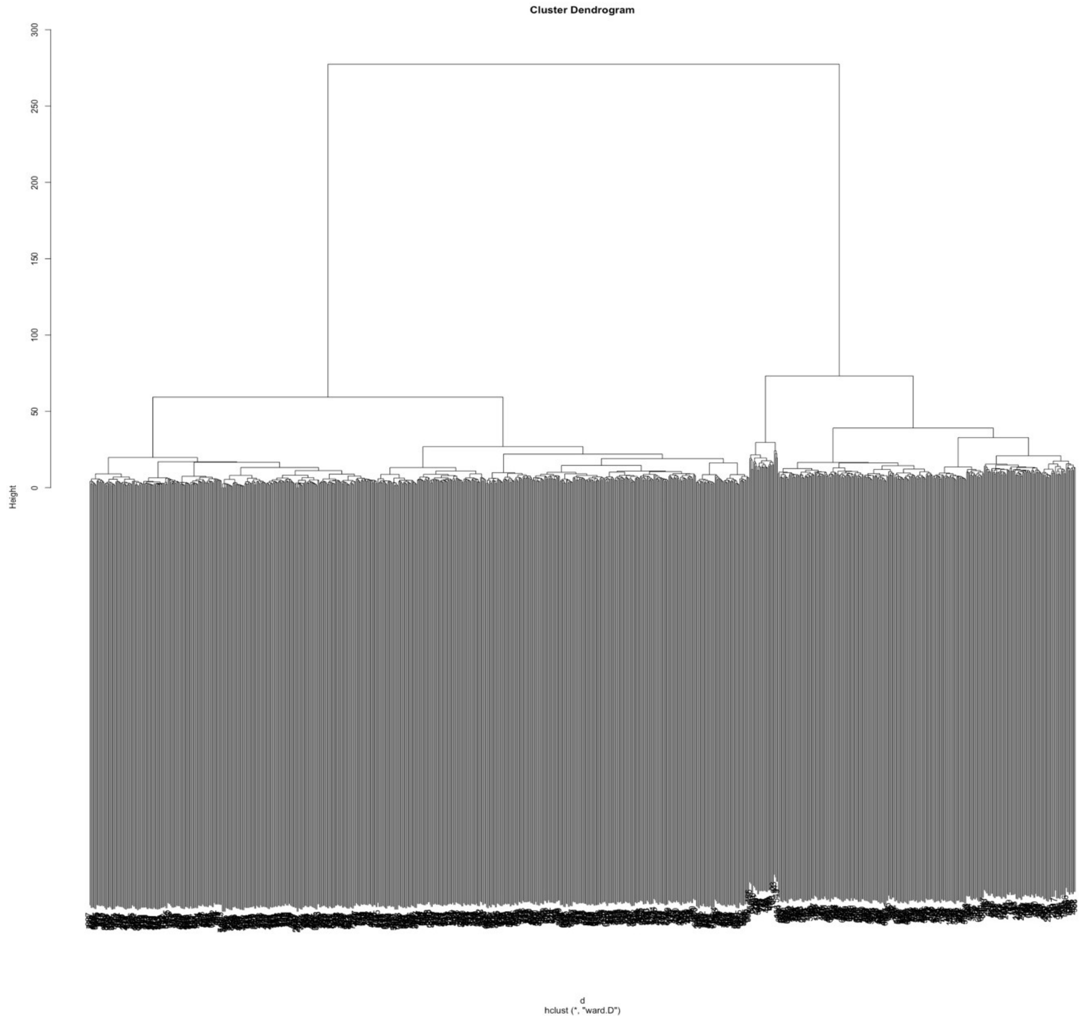


```
#clustering
  v <- as.matrix(dtm)
d<- dist(v)
#run hierarchical clustering using Ward,Äôs method
groups <- hclust(d,method="ward.D")
#plot dendogram, use hang to ensure that labels fall below tree
plot(groups, hang=1)
rect.hclust(groups,2)
#k means algorithm, 2 clusters, 100 starting configurations
kfit <- kmeans(d, 10, nstart=100)
#plot ,Äì need library cluster
library(cluster)
clusplot(as.matrix(d),   kfit$cluster,   color=T,   shade=T,   labels=2,
lines=0)
#kmeans ,Äì determine the optimum number of clusters (elbow method)
#look  for  ,Äúelbow,Äù  in  plot  of  summed  intra-cluster  distances
(withinss) as fn of k
wss <- 2:29
```

```
for (i in 2:29) wss[i] <- sum(kmeans(d,centers=i,nstart=25)$withinss)
plot(2:29, wss[2:29], type="b", xlab="Number of Clusters",ylab="Within
groups sum of squares")
```




Within groups sum of squares
200000
600000
1200000
5
10
15
20
25
30
Number of Clusters


Cluster Dendrogram
Height
0
50
100
150
200
250
300
d
hclust (*, "ward.D")

INDIVIDUAL REPORT

BUAN 6356

GROUP 9

TEAM MEMBER: **CHINMAY KARANDIKAR**

NET ID : **CSK180002**

I have had the liberty to lead many projects during my undergraduate stint in my home country. I have also handled a couple of projects at my workplace. However, the language 'R' is new to me. I decided to take a back seat and enjoy the luxury of taking orders rather than giving them. It was a new learning curve which was going to be instrumental towards shaping my career. Working with a group is not new to me, however working with every group is different. We started with a different dataset and three weeks before the deadline, we decided to switch it owing to our hunger for greater and richer analysis. I commend the courage of my group members to take such a defining decision at such a short notice.

No one in our team had prior experience in R. So, all we could end up doing on first few of our meetings was to look at datasets and select the one which we found interesting. While working on a project this is not the way to go about things. One fine weekend we sat down and started looking at datasets from an entirely different outlook. A good project needs great analysis. A great analysis needs a dataset which can give immense opportunities to identify different trends and patterns, which beholds a lion's share in the success of the project. Possessed with the same idea, we finally decided on a dataset. This dataset required a lot of cleaning, but it had brighter results ahead.

Different individuals have different ideas. Respecting each idea, hearing one out and then deciding on something substantial together are some of the essentials for a successful group project. Most of our group meetings took place in the library. We used to decide on the things which were to be done on the given day. Each person would then voluntarily pick a segment of their own liking and started working on it. Everybody used to reach the pre-decided place on time. People who could not manage to be there would complete the required work from home.

We spent almost two weeks working on the sentimental analysis. I would be lying if I were to say it wasn't tough. It was extremely tough. We had an output but not the one which we desired. Everyone from the group tried working on it, but to no avail. Until finally, Dharani broke the deadlock and gave us the much-desired output. Data cleaning was also a task. We spent about five days to get it done. The clustering algorithm was also a big challenge for us. Owing to the size of the dataset, we were getting many clusters in the dendogram. It did look extremely messy. We worked on it and eventually settled on the best looking dendogram which we have included in our project report. Sentimental analysis is something which is difficult, but we could realize it extremely well. We could conclude that the dataset has a higher number of positive reviews than the negative reviews.

I would be lying, if I were to say it was all merry throughout. No, it was not. There were differences of opinions, there were a few upset faces, few frowning as well, but, we handled the situations adeptly and dealt with the differences maturely. At the end of the day, all of us almost always managed to be on the same page. Obviously, 'In the end, it doesn't even matter.'

Our dataset had to cleaned from scratch. It was a tedious task, but Dharani managed it well. I was responsible for cleaning the 'Days_since_review' column, which I needed to take into consideration while designing the correlation plot. Our dataset is dataset which shows little or almost no correlation. I was advised to not go ahead with the correlation plot, but I decided against it. Upon realization, I found some very interesting and noteworthy relationships among variables. I have mentioned these in the project report. Akash performed the word cloud and played a major part in the sentimental analysis. Deeksha realized the basic visualization charts. Exploratory Data Analysis (EDA) was performed by Hardil. Dharani has written major parts of the report and has also designed the kin and clustering algorithms.