Sentiment Analysis of Drugs and Medicine Reviews

Submitted in partial fulfillment of the requirements for the

R Programming (Mini Project) Lab

by

1. Sumedh Ghavat - 20
2. Parth Kodnani - 34
3. Harshita Singh - 66

Under the guidance of

Mrs. Jayashree Hajgude



Department of Information Technology

Vivekanand Education Society’s Institute of Technology

2020-2021

# Table of Contents

[Problem Statement](#_pazgb816gdyj) 3

[Dataset](#_b2jtksrbwgnu) 4

[Extracting Data From Dataset](#_95xfx0yr8tme) 5

[Exploratory Data Analysis and Visualization](#_fbtmagwb4xv1) 6

[Data Preprocessing](#_sxbtpwiwyjj0) 17

[Sentiment Analysis](#_79z0sdeci3zg) 26

[Conclusion](#_s31wah6knkp0) 33

# 

# Problem Statement

Drug utilization review (DUR) is defined as an authorized, structured, ongoing review of prescribing, dispensing and use of medication. DUR encompasses a drug review against predetermined criteria that results in changes to drug therapy when these criteria are not met. It involves a comprehensive review of patients' prescription and medication data before, during and after dispensing to ensure appropriate medication decision-making and positive patient outcomes. As a quality assurance measure, DUR programs provide corrective action, prescriber feedback and further evaluations.

To examine the reviews regarding various drugs and medicines and their conditions and to analyse the sentiments and emotions in those reviews as positive or negative. Finally, visualise the data like the rating of the drugs, their useful count and overall sentiments as per the data analysis done above.

# 

# Dataset

***Dataset :*** Drug Review Dataset

***Source :*** UCI Machine Learning Repository

***Link :*** [Drug Review Dataset (Drugs.com) Data Set](https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29#)

***Description :*** The dataset provides patient reviews on specific drugs along with related conditions and a 10 star patient rating reflecting overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites.

***Dataset Exploration (Attributes) :*** The dataset has 6 attributes :

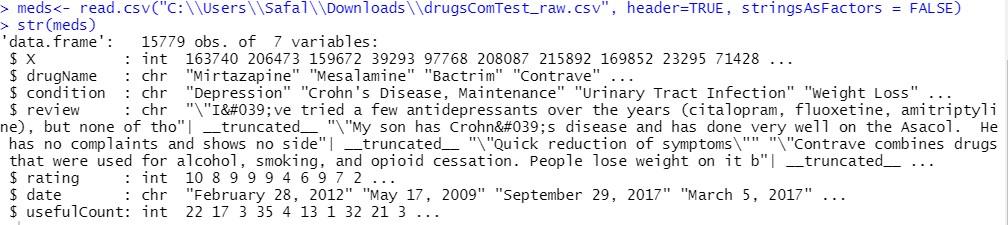
1. *drugName (categorical):* name of drug
2. *condition (categorical):* name of condition
3. *review (text):* patient review
4. *rating (numerical):* 10 star patient rating
5. *date (date):* date of review entry
6. *usefulCount (numerical):* number of users who found review useful

# 

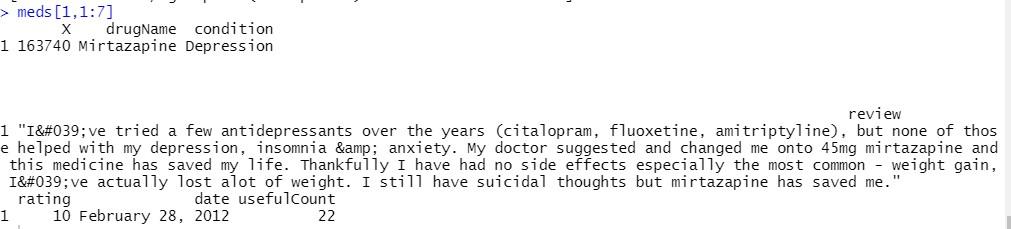
# Extracting Data From Dataset

To extract the data from the dataset we make use of data frames. First we create a data frame named ‘meds’ by reading the csv file of the actual dataset using the following code snippet.

* meds <- read.csv("C:\\Users\\Safal\\Downloads\\drugsComTest\_raw.csv", he ader=TRUE, stringsAsFactors = FALSE)

Then we understand the attributes and some of the rows of the data frame as shown in the following two images.

***Figure 3.1 Compact Display of the Data Frame meds***

***Figure 3.2 Row 1 of Data Frame meds***

The dimension of the data frame is 15779 rows and 7 columns which can be seen in the following code snippet.

* dim(meds)

[1] 15779 7

# Exploratory Data Analysis and Visualization

Exploratory Data Analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. Primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

The packages and libraries used here are ggplot, tidyverse, DataExplorer, plot3D. Tidyverse is a collection of essential R packages like ggplot and tidyr which is used for data analysis, transformation and visualisation. The DataExplorer package automates Data Exploration and it helps in visualising different variables with typical graphical techniques. First, we mention these libraries in the code as mentioned below.

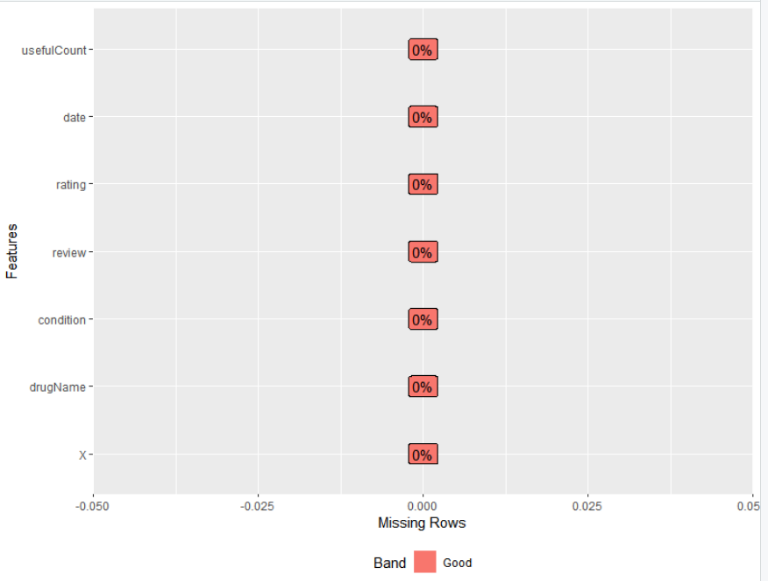
> library(tidyverse)

> library(DataExplorer)

> library(plot3D)

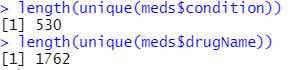
We then perform exploratory data analysis on the data frame ‘meds’ as created above. We first check for missing values in our dataset by using the plot\_missing function by DataExplorer package. There are no missing values in our dataset.

> plot\_missing(meds)

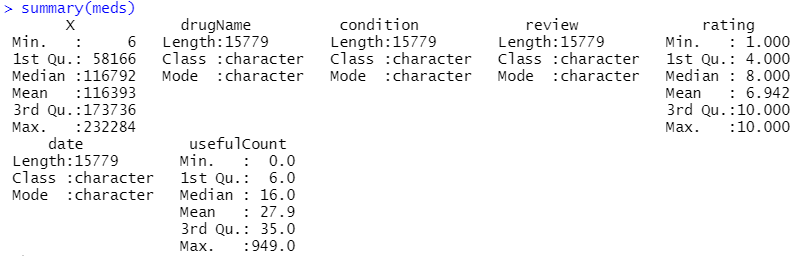


***Figure 4.1 Checking for missing values in the datas***

We find out that out of the 15779 entries of drug names and conditions, there are a total 1762 unique drugs and 530 unique conditions using the unique() function in R as shown below.



Then we plot the summary of all attributes which includes the mean, median, min, max, Q1, Q3 and IQR as shown below.



***Figure 4.2 Summary of the dataset attributes***

We then find out the mean for rating and useful count of each unique drug.

> uniqueruc= meds %>%

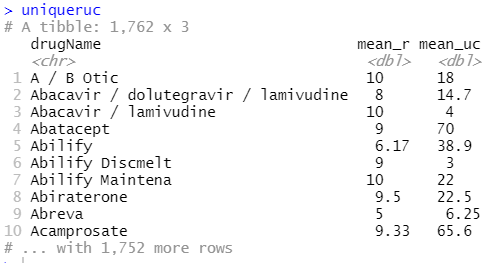
+ group\_by(drugName) %>%

+ summarise(mean\_r = mean(rating), mean\_uc = mean(usefulCount))

> meds %>% summarise(mean\_r = mean(rating), mean\_uc = mean(usefulCount))

mean\_r mean\_uc

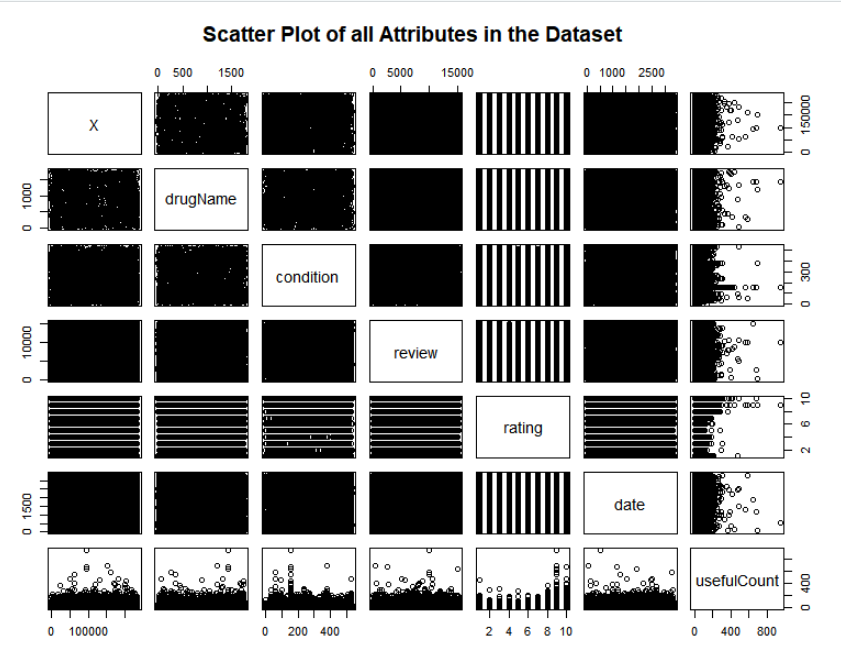
1 6.942455 27.90044



***Figure 4.3 Mean and Useful Count for each Unique Drug***

We then plot a scatterplot for all the attributes in our dataset as follows.

> plot(meds, main="Scatter Plot of all Attributes in the Dataset ")



***Figure 4.4 Scatterplot of all attributes***

We then plot a bar graph for 25 most useful drugs and useful count of some drugs as per their condition.

> mostusefulcount= uniqueruc[order(-uniqueruc$mean\_uc),]

> mud=mostusefulcount$mean\_uc[1:25]

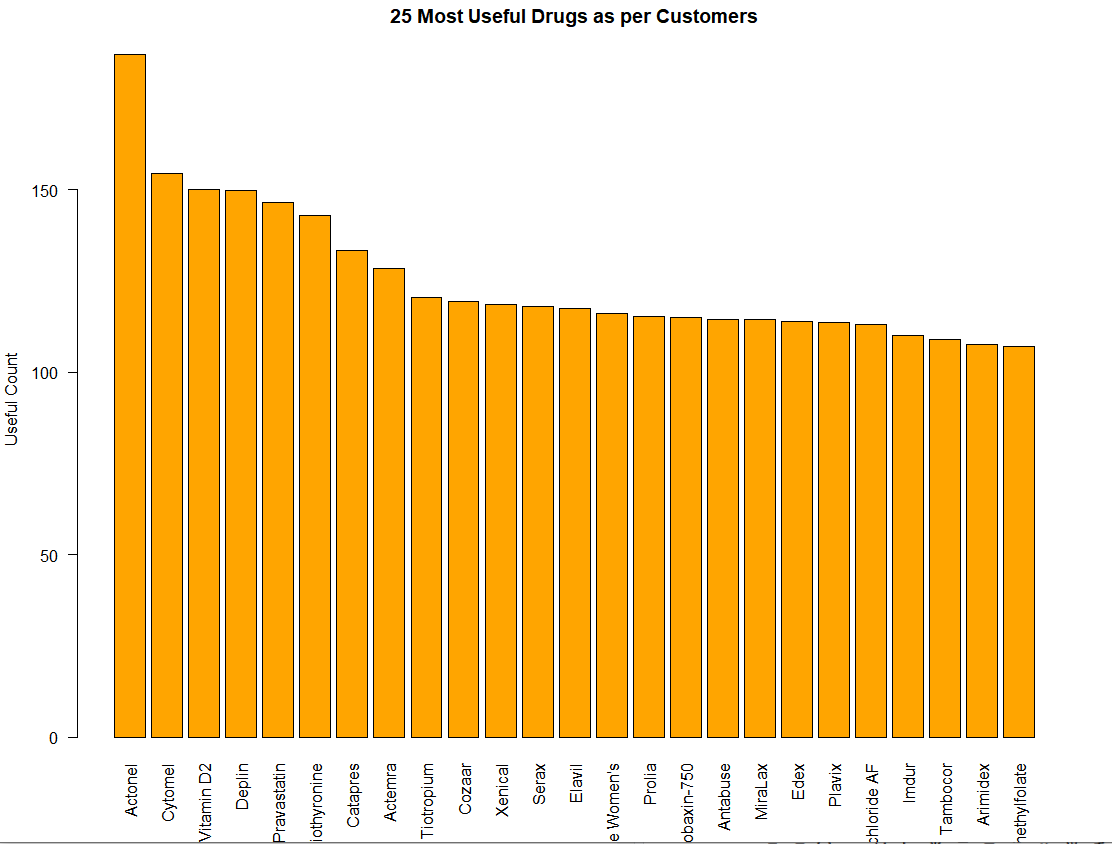
> mudn=mostusefulcount$drugName[1:25]

> barplot(mud,names.arg=mudn,las=2, ylab="Useful Count",main="25 Most Useful Drugs as per Customers",col="Orange")

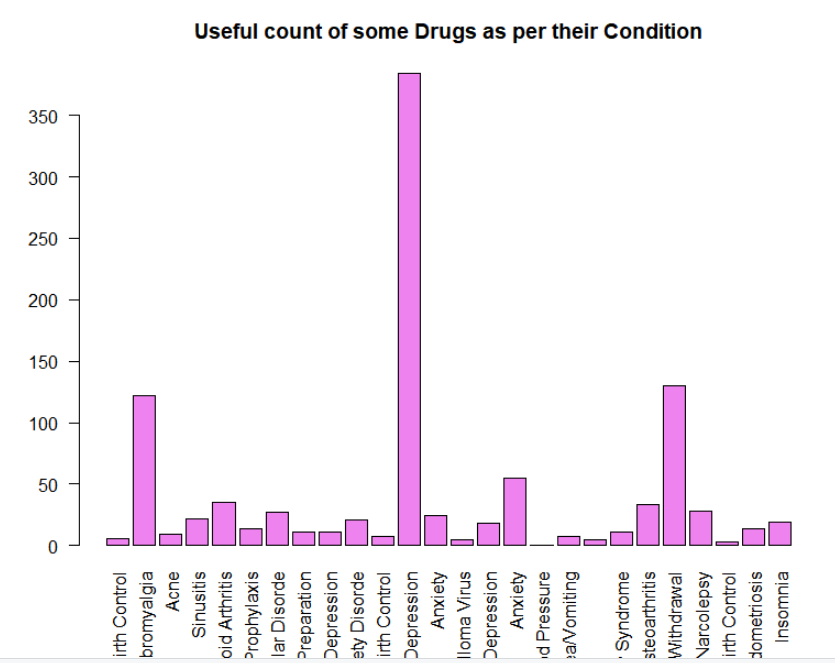
> confirst25=meds$condition[1000:1025]

> ucfirst25=meds$usefulCount[1000:1025]

> barplot(ucfirst25,names.arg=confirst25,las=2,main="Useful count of some Drugs as per their Condition",col="violet")



***Figure 4.5 Bar graph 25 most useful drugs***



***Figure 4.6 Bar Graph of Drugs according to condition***

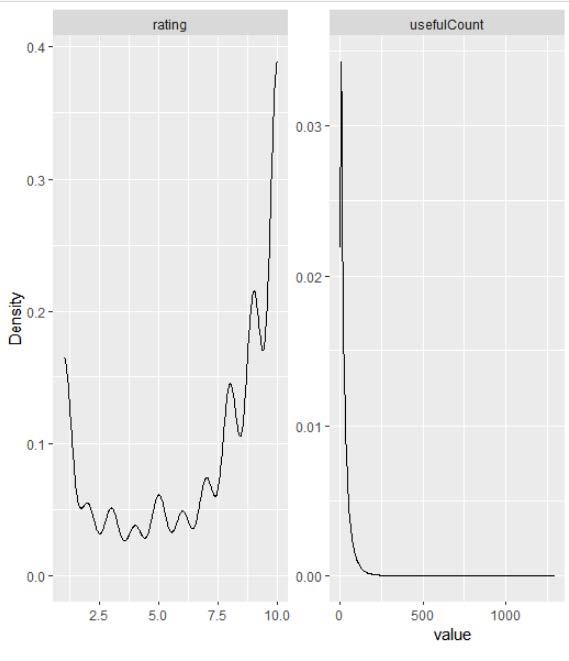
Finally, under Exploratory data analysis we perform various plots like Density plot for rating and useful count, box plot for drug ratings, histogram for drug ratings and their count, correlation plot for 3 numeric attributes (review id, rating and useful count and their correlation).

> plot\_density(meds)

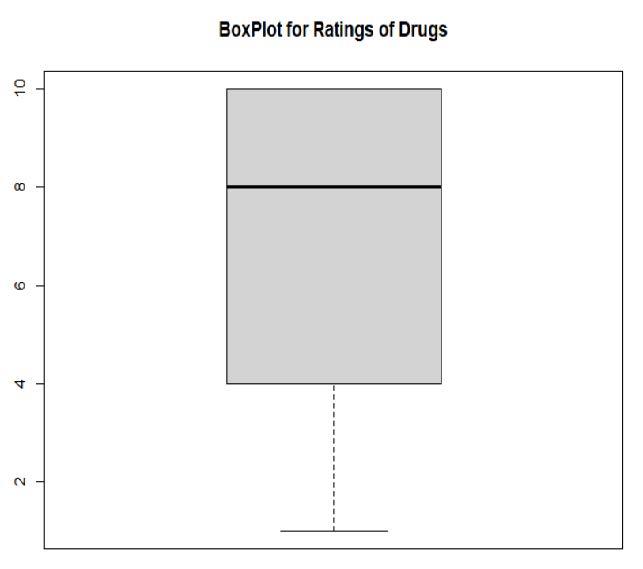
> boxplot(meds$rating, main="BoxPlot for Ratings of Drugs")

> ggplot(meds, aes(x=rating)) +ggtitle("Histogram for Drug Ratings") + geom\_histogram( col="red",fill="green", alpha = .4, binwidth = 3)

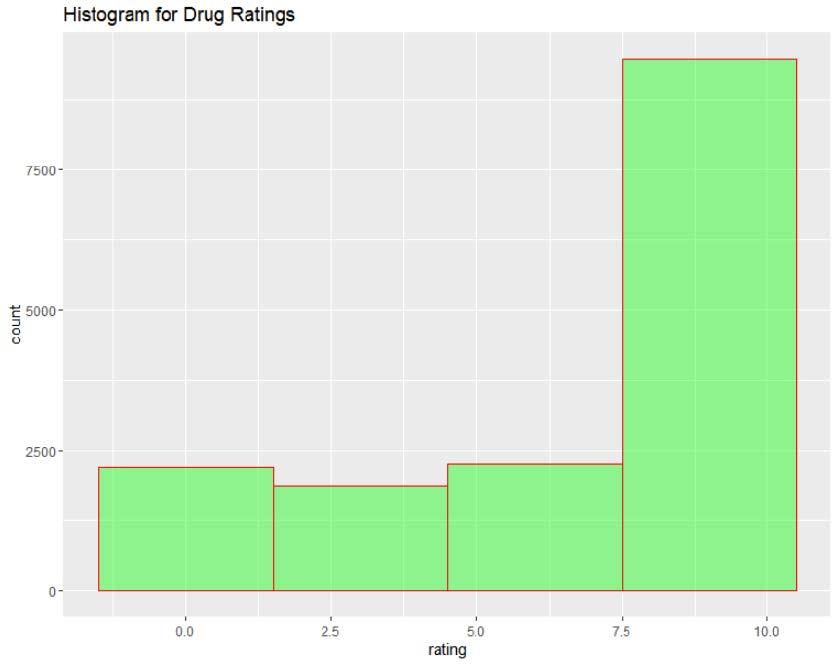
> plot\_correlation(meds, type = 'continuous','Review.Date')



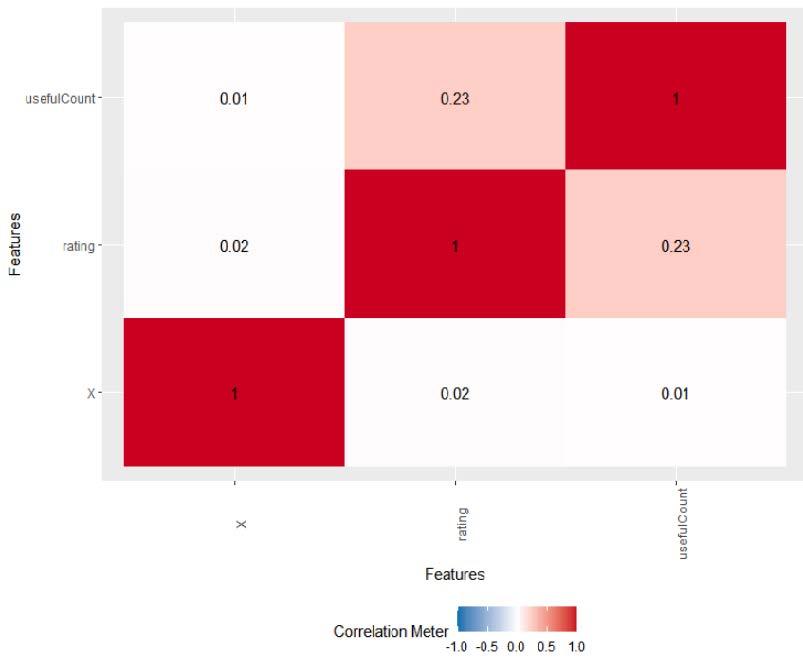
***Figure 4.7 Density plot of drug ratings and useful count***



***Figure 4.8 Box plot of drug ratings***

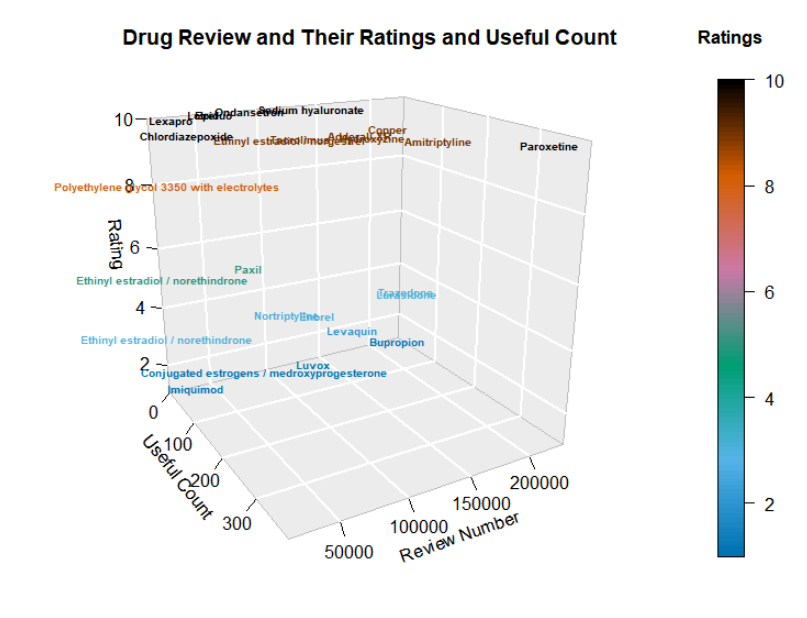


***Figure 4.9 Histogram of drug ratings***



***Figure 4.10 Correlation Plot***

At last, we try a 3D plot of a 3D view of the three numeric attributes (the user rating, the useful count and the review id) by using the text3D plot function under the plot3D package as follows.



***Figure 4.11 3D Plot of numeric attributes of some drugs***

# Data Preprocessing

Data pre-processing and data analysis is an important step under text mining by which we get clean text data and also, we can find out common tokens and most frequent words.

The libraries and packages that we have used here are: tm, wordcloud, ggplot2, dplyr, tidytext. Tm which stands for text mining is a very useful package in Data preprocessing and data analytics and it provides various functions in tm\_map like tolower, removePunctuation, removeNumbers, remove Stopwords and Stemming. It also provides functions to find word association and correlation and the term document matrix of each word in various docs (reviews). Wordcloud package helps in forming the word cloud of the most frequent words in the data. The dplyr package is used in data manipulation with functions like mutate and filter. The tidytext package is used to form tokens and ngrams(bigrams, trigrams etc).

Here we have performed various tasks like forming tokens and trigrams out of preprocessed data, finding association between words, forming the term document matrix and some visualisation based on the most common words as discussed below.

Code:

library(tm)

library(wordcloud)

library(syuzhet)

# Loading the Dataset

meds<- read.csv("C:\\Users\\Safal\\Downloads\\drugsComTest\_raw.csv", header=TRUE, stringsAsFactors = FALSE)

str(meds)

# Proper encoding format for the dataset

corpus <- conv(meds$review)

corpus <- Corpus(VectorSource(corpus)) inspect(corpus[1:5])

# Data preprocessing

corpus <- tm\_map(corpus, tolower)

corpus <- tm\_map(corpus, removePunctuation)

corpus <- tm\_map(corpus, removeNumbers)

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

corpus <- tm\_map(corpus, stripWhitespace)

corpus <- tm\_map(corpus, stemDocument) inspect(corpus[1:5])

reviews\_final <- corpus

# Forming the Term Document Matrix length(reviews\_final)

tdm <- TermDocumentMatrix(reviews\_final[1:"10000"])

tdm <- as.matrix(tdm)

tdm[12:39,1:20]

#5 most common words

tdm\_v <- sort(rowSums(tdm),decreasing=TRUE)

tdm\_d <- data.frame(word = names(tdm\_v),freq=tdm\_v) # Display the top 5 most frequent words

head(tdm\_d, 5)

barplot(tdm\_d[1:5,]$freq, las = 2, names.arg = tdm\_d[1:5,]$word, col ="yellow", main ="Top 5 most frequent words", ylab = "Word frequencies")

#Association

tdms <- TermDocumentMatrix(reviews\_final[1:"10000"])

findAssocs(tdms, terms = ("effect","fever","birth","depress","cough"), corlimit = 0.25)

#findAssocs(tdms, terms = findFreqTerms(tdms, lowfreq = 1000), corlimit = 0.25)

#Plotting words with frequency above 500 w <- rowSums(tdm)

w <- subset(w, w>=500)

barplot(w,las=2, col="green", ylab="Count", main="Bar Graph of occurrences of frequent words")

#Forming the word cloud

w<-sort(rowSums(tdm),decreasing=TRUE) set.seed(2000)

wordcloud(words=names(w), freq=w, max.words=70, random.order=T, min.freq=5, colors=brewer.pal(25,"Dark2"), scale=c(3,0.3))

#Forming Tokens

tokens <- meds %>%

unnest\_tokens(output = word, input = reviews) tokens

cleaned\_tokens <- tokens %>% anti\_join(get\_stopwords())

nums <- cleaned\_tokens %>%

filter(str\_detect(word, "^[0-9]")) %>% select(word) %>% unique()

Nums

#Unique Tokens

cleaned\_tokens <- cleaned\_tokens %>% anti\_join(nums, by = "word")

length(unique(cleaned\_tokens$word))

#Histogram

cleaned\_tokens %>%

count(word, sort = T) %>% rename(word\_freq = n) %>% ggplot(aes(x=word\_freq)) +

geom\_histogram(aes(y=..count..), color="black", fill="blue", alpha=0.3, binwidth = 0.2) + scale\_x\_continuous(breaks=c(0:5,10,100,500,10e3), trans="log1p", expand=c(0,0)) + scale\_y\_continuous(breaks=c(0,100,1000,5e3,10e3,5e4,10e4,4e4), expand=c(0,0)) + theme\_bw()

#Removing Rare Words from Tokens

rare <- cleaned\_tokens %>% count(word) %>%

filter(n<10) %>% select(word) %>% unique()

cleaned\_tokens <- cleaned\_tokens %>% filter(!word %in% rare$word) length(unique(cleaned\_tokens$word)) first50tokens=c(cleaned\_tokens$word[1:50]) first50tokens

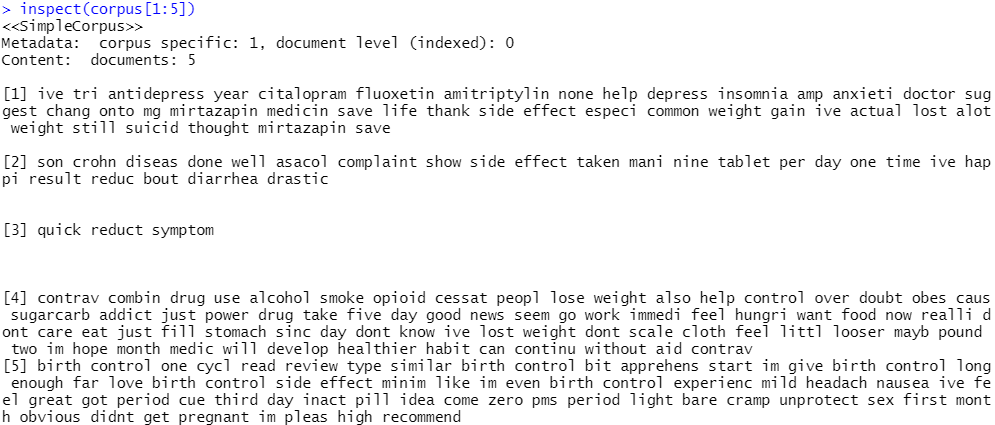
#Forming Trigrams

trigrams <- meds %>%

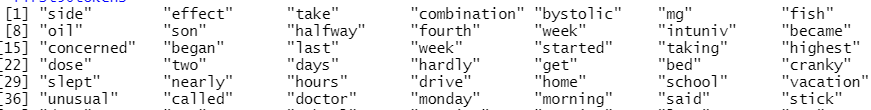
unnest\_tokens(trigram, reviews, token = "ngrams", n = 3) trigrams %>% select(trigram) trigrams %>%

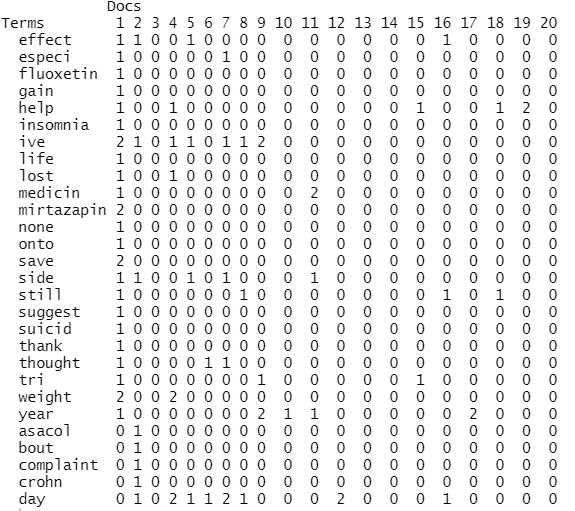
count(trigram, sort = TRUE)

Output:

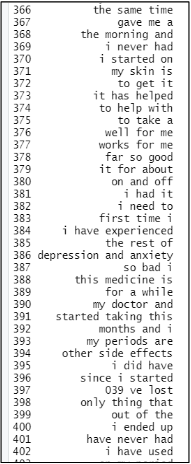
****

***Figure 5.1 Snapshot of encoded corpus reviews of first 5 rows***

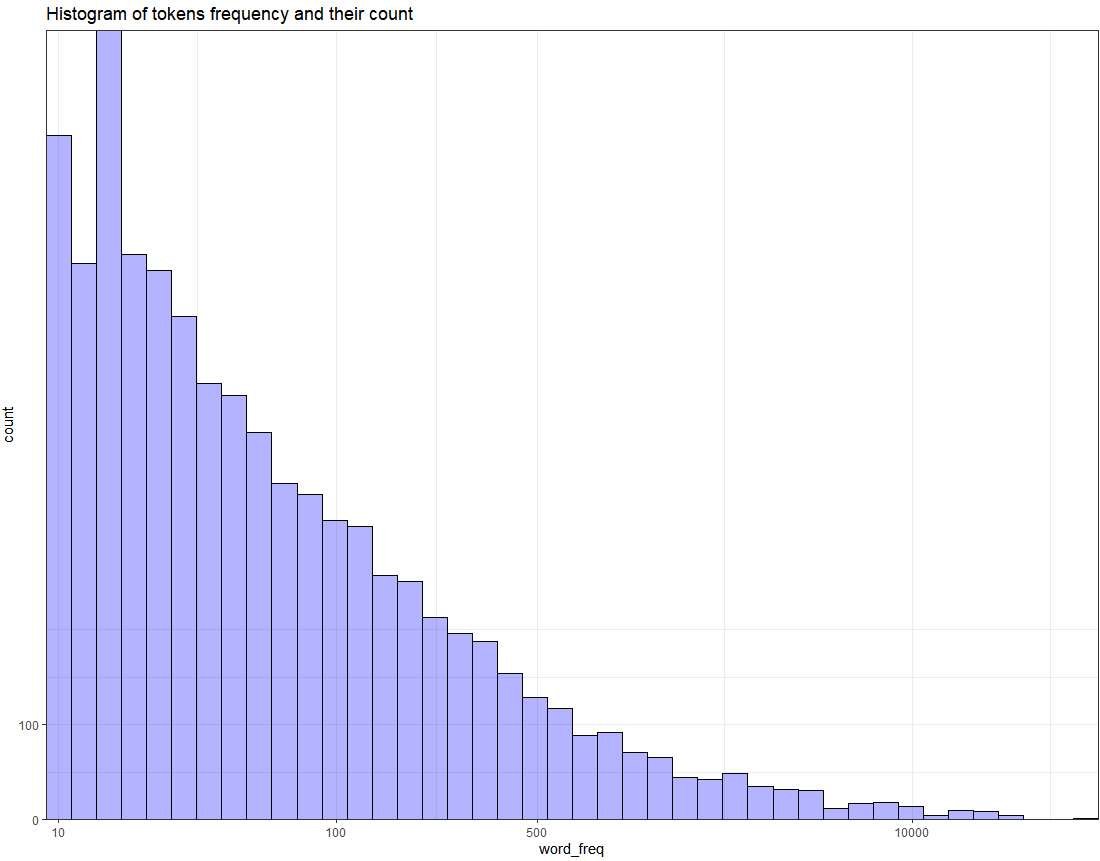
***Figure 5.2 Example of some cleaned tokens***



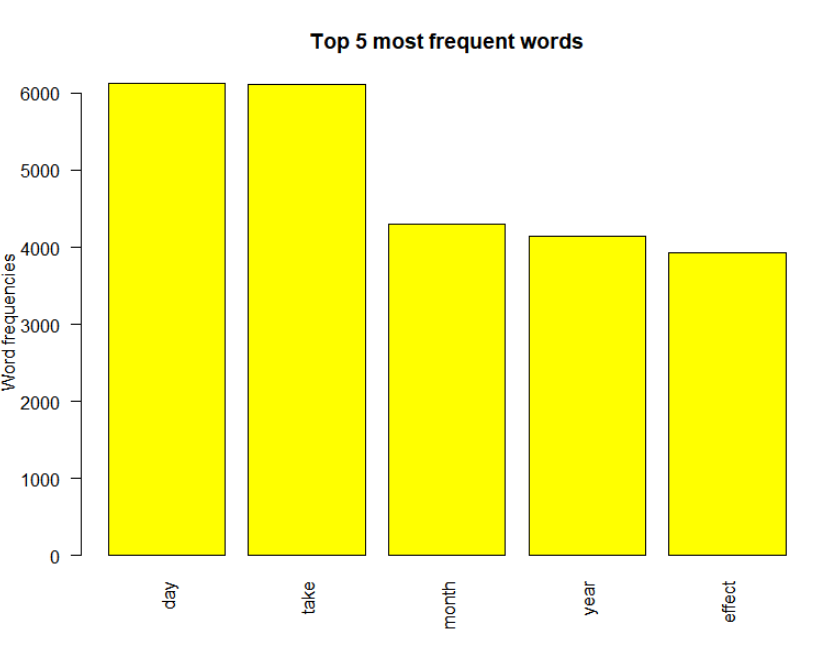
***Figure 5.3 Term Document Matrix for 20 reviews***



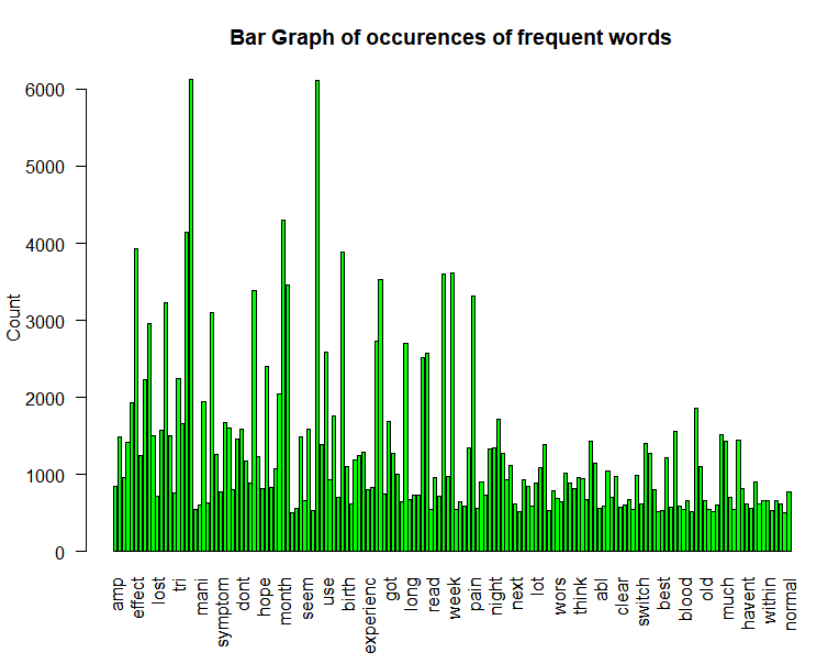
***Figure 5.4 Snapshot of some Trigram***



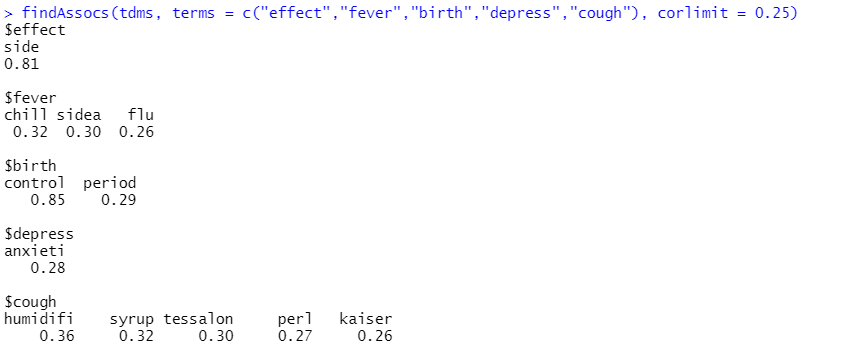
***Figure 5.5 Histogram of frequent words and their count***



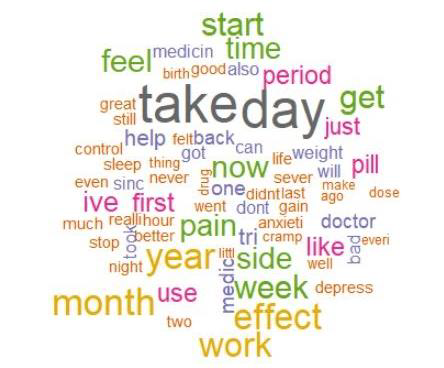
***Figure 5.6 Bar graph of 5 most common words and their count***

******

***Figure 5.7 Bar graph of words with frequency more than 500***



***Figure 5.8 Association of some words based on correlation***



***Figure 5.9 Word Cloud of most common word***

# Sentiment Analysis

Sentiment analysis is a text mining technique that uses machine learning and natural language processing (nlp) to automatically analyze text for the sentiment of the writer (positive, negative, neutral, and beyond). Aspect-based sentiment analysis organizes text like customer feedback or product reviews and then mines text for sentiment so you can see which categories are positive or negative.

The libraries and packages used here are sentiment and syuzhet. The sentimentr package calculates text polarity sentiment at the sentence level and optionally aggregates by rows or grouping variable(s). The syuzhet package is used for the extraction of sentiment and sentiment-based plot arcs from text and it also gives 8 NRC Emoticon Lexicons like trust, anger, joy, etc and 2 sentiments as positive and negative.

In sentiment analysis we try to find sentiments by word i.e how much does each word contribute to the sentiments and also the sentiments based on overall reviews and their score based on 8 emotions like anger, trust, fear, joy, etc and positive and negative scores along with a final score. We then merge this final score column to the actual dataset to understand the score for each review and their drug name.

## **Code:**

# Method 1

# Sentiments for each review

sentiment\_data <- iconv(meds$review)

s <- get\_nrc\_sentiment(sentiment\_data) s[1:10,]

#Calculating the score based on sentiments

s$score <- s$positive-s$negative

s[1:10,]

#Final score excel sheet

write.csv(x=s, file="C:\\Users\\Safal\\Downloads\\FinalScores.csv")

review\_score <- colSums(s[,]) print(review\_score)

#Bar plot for scores

barplot(colSums(s), las=2, col=rainbow(10), ylab='Count', main='Sentiment')

total <- colSums(ab[2]) total

pospercent=round(((colSums(s[10])/total)\*100), digits=2) negpercent=round(((colSums(s[9])/total)\*100), digits=2) postring= paste(c("Positive (",pospercent,") %"), collapse="") negstring= paste(c("Negative (",negpercent,") %"), collapse="")

ab <- data.frame(group=c(postring,negstring), value=c(colSums(s[10]),colSums(s[9])))

# Method 2

sentiment\_data <- iconv(meds$review)

sentiment <- analyzeSentiment(sentiment\_data)

# Extract dictionary-based sentiment according to the QDAP dictionary

sentiment$SentimentQDAP

# View sentiment direction (i.e. positive, neutral and negative)

convertToDirection(sentiment$SentimentQDAP)

#Pie Chart for reviews and overall sentiments

library(ggplot2)

bp <- ggplot(ab, aes(x="", y=value, fill=group)) + geom\_bar(width=1, stat="identity")

pie <- bp + coord\_polar("y", start=0)

pie

get\_sentiments("nrc")

#Sentiment by words

sentiment\_reviews = cleaned\_tokens %>%

left\_join(get\_sentiments("nrc")) %>%

rename(nrc = sentiment) %>%

left\_join(get\_sentiments("bing")) %>%

rename(bing = sentiment) %>%

left\_join(get\_sentiments("afinn")) %>%

rename(afinn = value) rmarkdown::paged\_table(sentiment\_reviews)

#Polarity of word counts

polarity\_word\_counts <- sentiment\_reviews %>%

filter(!is.na(bing)) %>%

count(word, bing, sort = TRUE)

polarity\_word\_counts

#Visual Sentiments

polarity\_word\_counts %>%

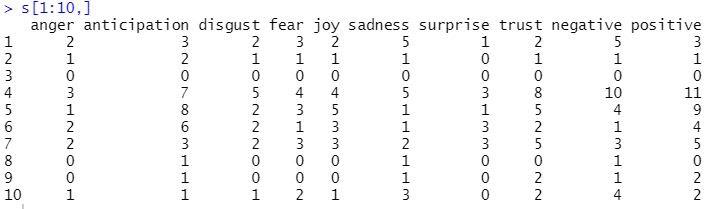
filter(n > 3000) %>%

mutate(n = ifelse(bing == "negative", -n, n)) %>%

mutate(word = reorder(word, n)) %>%

ggplot(aes(word, n, fill = bing)) + geom\_col() + coord\_flip()+scale\_y\_discrete(labels = abbreviate) + labs(y = "Contribution to sentiment")

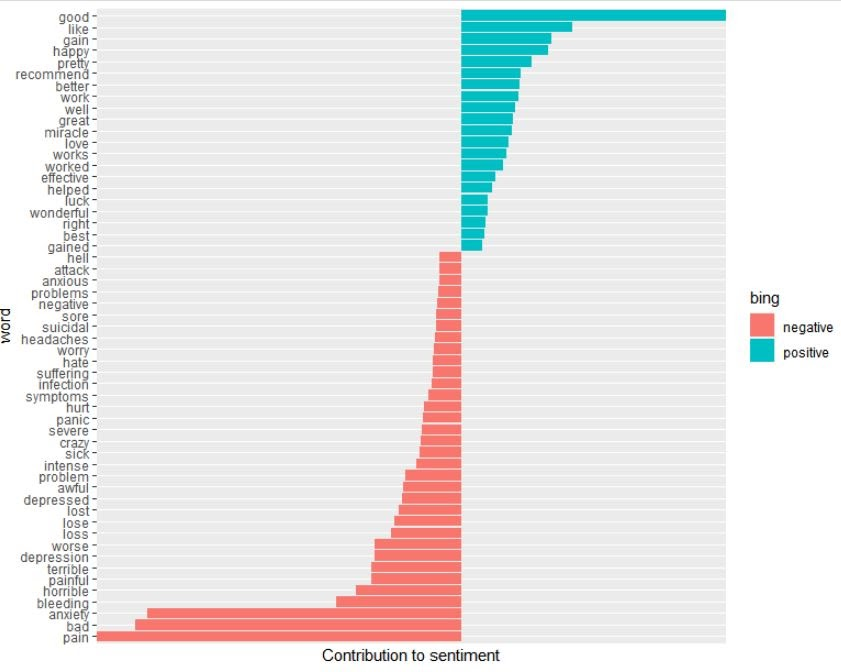
## **Output:**

****

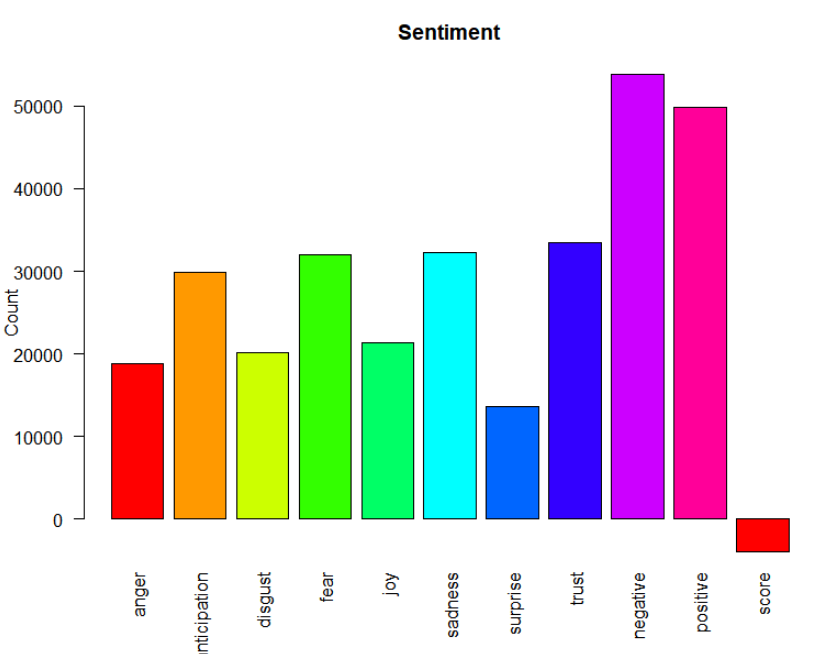
***Figure 6.1 Emotion and Sentiment Scores for 10 reviews***

******

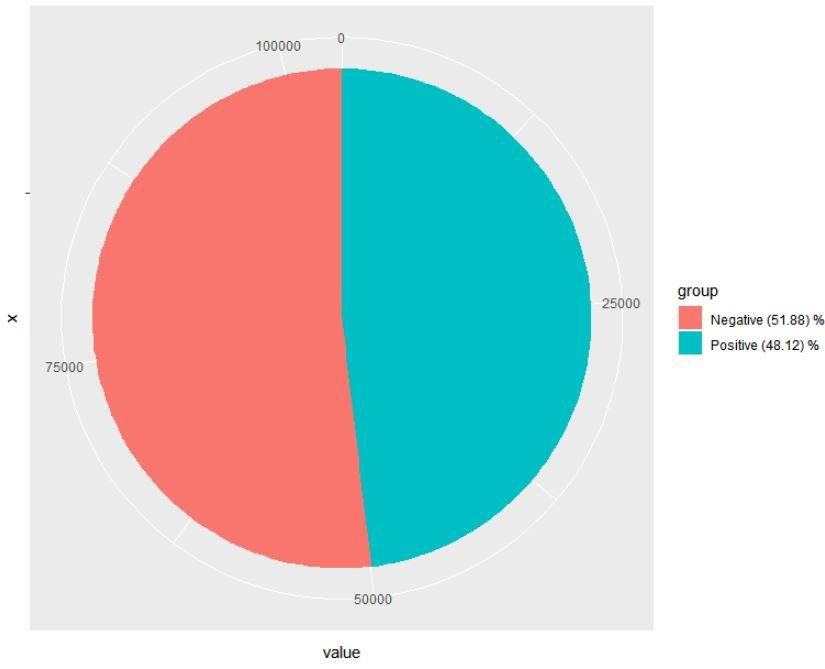
***Figure 6.2 Total Emotion and Sentiment Scores sum***

******

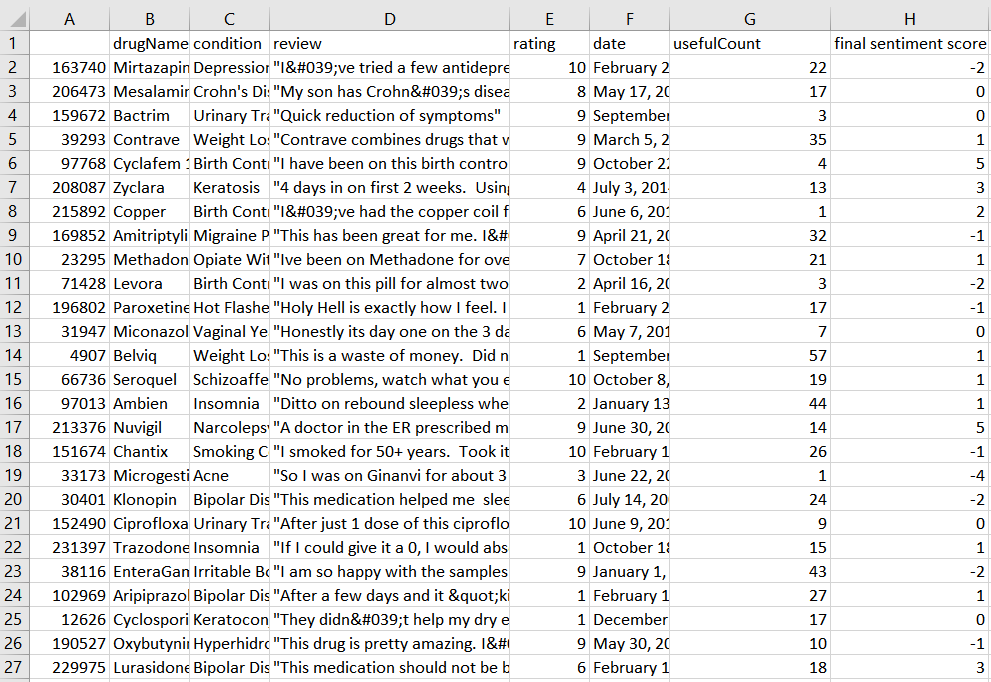
***Figure 6.3 Word wise contribution to the sentiments***



***Figure 6.4 Bar Graph of scores of 8 emotions and positive, negative and overall score***

******

***Figure 6.5 Pie Chart Indicating Total Percentage of Positive and Negative Emotions***



***Figure 6.6 Final Merged Excel Sheet of drug names, reviews and their sentiment scores.***

# Conclusion

As per the exploratory data analysis, the attributes rating and useful count are correlated to each other and the mean rating is around 8 for all the drugs.

In Data Analysis we understand the frequent words like takes, months etc which occur frequently in the dataset and their associated words.

As per the sentiment analysis done on the overall reviews under the dataset, the percentage of negative reviews is around 52% whereas the percentage of positive reviews is 48%. This indicates that more reviews in the dataset are on the negative side than on the positive side in the dataset.

The emotions like sadness, trust and fear are mostly found in the sentiments of the reviews whereas positive emotions like joy and surprise are low in number. This indicates that mostly people are disappointed with the drugs overall.

The final sentiment scores have been merged with the reviews in a new excel sheet. This makes it easy to analyse which particular drug with a condition and its review has what score. Scores in negative sign indicates that the drug is not good, between 0-2 indicates that it is neutral as such and scores above 3 indicate that the review is on the positive side.