

## IMDb sentimental Analysis (Shyam Gandhi)

### Extracting file from source.

```
file<-read.csv("C:/Users/Shyam Gandhi/Desktop/Traintrial.csv", header = T,  
sep = ",")
```

### Extracting the comment text

```
file_text<-file$text
```

### Convert all text to lower case

```
file_text<-tolower(file_text)
```

### Removing extra text between the words

```
file_text<-gsub("\\s+", " ",file_text)
```

### For removing extra space and tab between text

```
file_text<-gsub("[ |\\t]{2,}", " ",file_text)
```

### For removing any space at beggining of text

```
file_text<-gsub("^ ", "",file_text)
```

### For removing any space at end of text

```
file_text<-gsub(" $", "",file_text)
```

### For removing any punctuation

```
file_text<-gsub("[[:punct:]]", "",file_text)
```

### For removing predifined stopwords

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(stopwords)
```

```
##
## Attaching package: 'stopwords'

## The following object is masked from 'package:tm':
##
##      stopwords

file_text<-removeWords(file_text, stopwords("en"))
```

## For remvng my own made stopwords

```
file_text<-removeWords(file_text, c("film","movie","films","movies"))
```

## For removing all numbers

```
file_text<-removeNumbers(file_text)
file_text<-gsub("[ |\\t]{2,}", " ",file_text)
```

## For lemmatizing the string

```
library(textstem)

## Loading required package: koRpus.lang.en
## Loading required package: koRpus
## Loading required package: sylly

## For information on available language packages for 'koRpus', run
##
##   available.koRpus.lang()
##
## and see ?install.koRpus.lang()

file_text<-lemmatize_strings(file_text, language = "porter")
```

#Making corpus of text

```
file_corpus<-Corpus(VectorSource(file_text))
```

## Creating term frequency matrix

```
dtm<-TermDocumentMatrix(file_corpus)
m<-as.matrix(dtm)
v<-sort(rowSums(m), decreasing = T)
d<-data.frame(word=names(v), freq=v)
head(d,10)

##              word freq
## good          good  144
## one           one   95
```

##	like	like	78
##	make	make	78
##	get	get	74
##	see	see	71
##	watch	watch	53
##	great	great	53
##	time	time	51
##	character	character	51

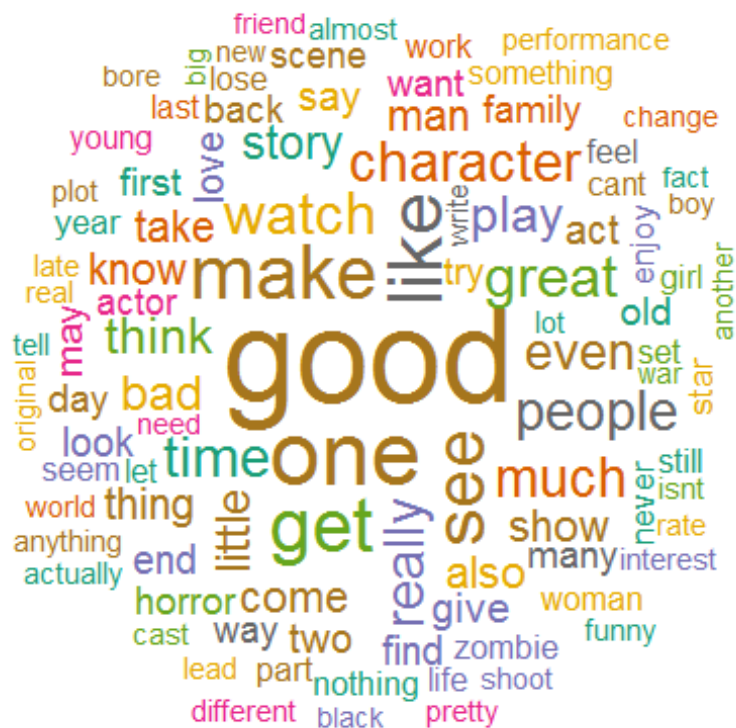
## #Creating WordCloud to demonstrate term frequency

```
library(wordcloud)

## Loading required package: RColorBrewer

library(RColorBrewer)

set.seed(200)
wordcloud(words = d$word, freq = d$freq, min.freq = 1, max.words = 100,
random.order = F, random.color = T, colors = brewer.pal(8, "Dark2"))
```



## Ranking Sentiment

```
library(syuzhet)

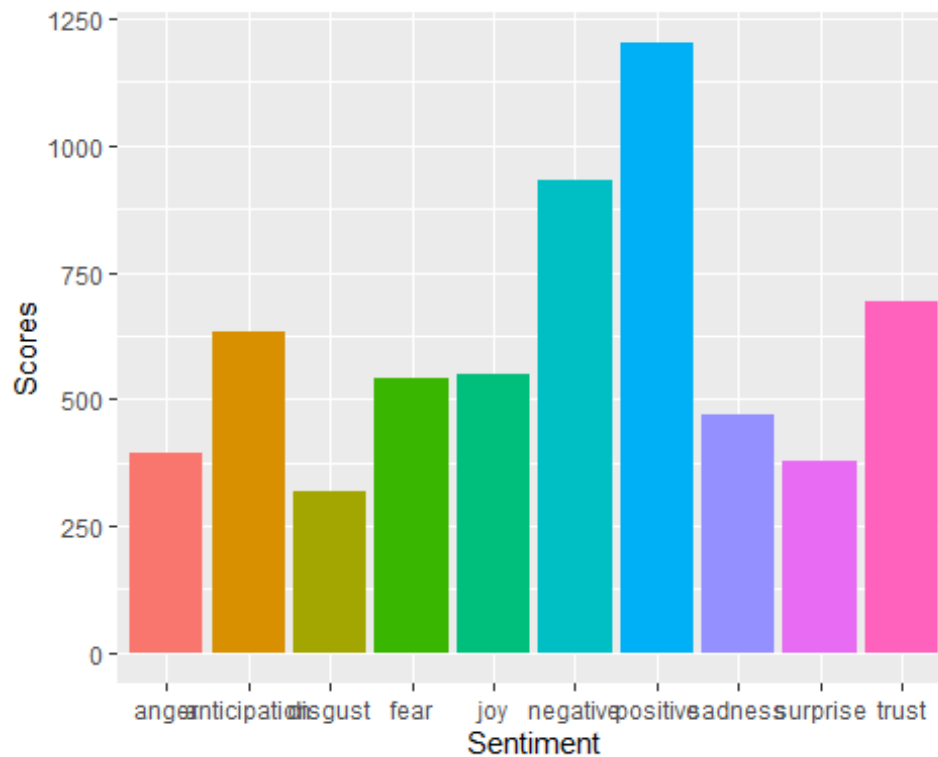
mysentiment<-get_nrc_sentiment(file_text)
sentiment_score<-data.frame(colSums(mysentiment[,]))
```

```
names(sentiment_score)<-"Score"  
sentiment_score<-cbind("sentiment"=rownames(sentiment_score),sentiment_score)  
rownames(sentiment_score)<-NULL  
sentiment_score
```

```
##      sentiment Score  
## 1      anger   393  
## 2 anticipation 634  
## 3    disgust  319  
## 4      fear   541  
## 5      joy    548  
## 6    sadness  471  
## 7    surprise 378  
## 8      trust  691  
## 9    negative 933  
## 10   positive 1203
```

#Ploting sentiment

```
library(ggplot2)  
  
##  
## Attaching package: 'ggplot2'  
  
## The following object is masked from 'package:NLP':  
##  
##      annotate  
  
ggplot(data=sentiment_score, aes(x=sentiment,  
y=Score))+geom_bar(aes(fill=sentiment),stat =  
"identity")+theme(legend.position = "none")+xlab("Sentiment")+ylab("Scores")
```

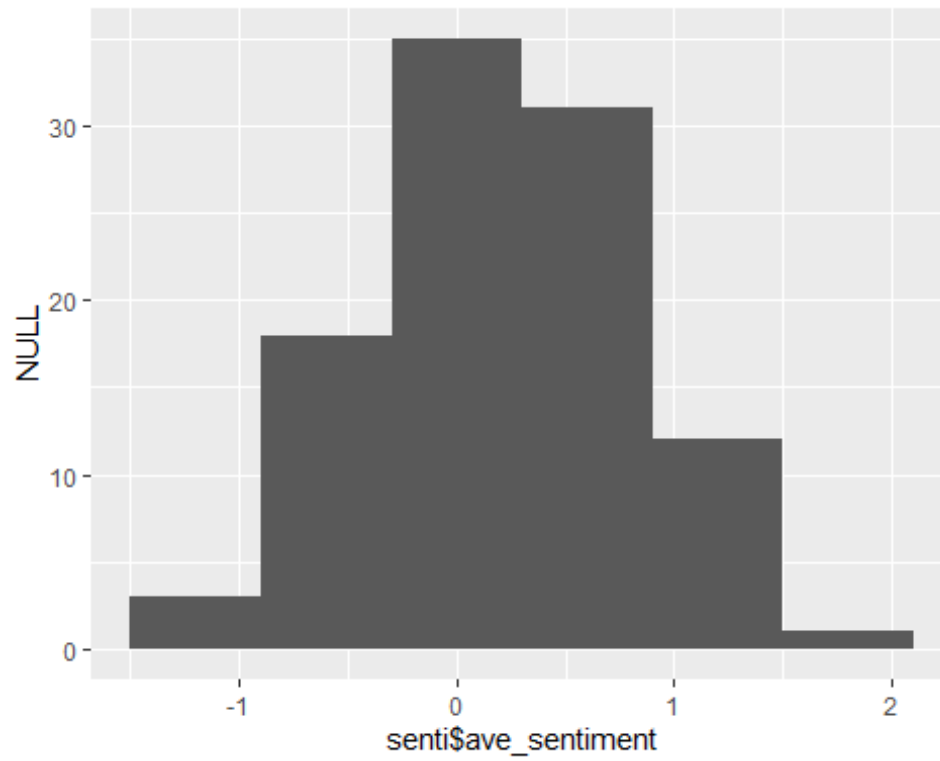


## Assigning score to each individual text

```
library(sentimentr)

##
## Attaching package: 'sentimentr'
##
## The following object is masked from 'package:syuzhet':
##
##   get_sentences

senti<-sentiment_by(file_text)
senti$ave_sentiment<-round(senti$ave_sentiment,3)
qplot(senti$ave_sentiment,geom = "histogram", binwidth =0.6)
```



```
sentiterms<-extract_sentiment_terms(file_text)
sentiterms$rating<-senti$ave_sentiment
```