Financial_News_Sentimental_Analysis

JungHwan Park

3/5/2022

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tm)
```

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

```
library(NLP)
library(proxy)
```

```
##
## Attaching package: 'proxy'
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
library(RTextTools)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(fpc)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(cluster)
library(tm)
library(stringi)
library(wordcloud2)
library(dplyr)
library(tidyr)
library(stringr)
library(tidytext)
library(topicmodels)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(readxl)
library(caTools)
library(readr)
library(tidytext)
data <- read_csv("all-data.csv")
```

```
## Registered S3 method overwritten by 'cli':
##   method     from
##   print.tree tree
## Rows: 4846 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Sentiment, text
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Data cleaning

```
data_pos <- data %>% filter(Sentiment == "positive")
data_neg <- data %>% filter(Sentiment == "negative")
data_neu <- data %>% filter(Sentiment == "neutral")

cat(data_pos$text, file = "data_pos.txt")
cat(data_neg$text, file = "data_neg.txt")
cat(data_neu$text, file = "data_neu.txt")

#texts_names<-list.files("text")
#texts_names<-paste0("text/",texts_names)
#texts<-lapply(texts_names,readLines)

cname_stat418 = "C:/Users/JUNGHWAN PARK/Downloads/Financial_News"
data_dtm_Corp <-VCorpus(DirSource(cname_stat418))
#inspect(amz_all_dtm_Corp)

data_dtm_Corp[[1]]$meta$id <- "positive"
data_dtm_Corp[[2]]$meta$id <- "negative"
data_dtm_Corp[[3]]$meta$id <- "neutral"
```

```
f <- function (x) {gsub("[^[:alnum:]]", ' ', x)}   # function for erasing alphanumeric character
s
data_dtm_Corp <- tm_map(data_dtm_Corp,f)
```

- Erased non-alphanumeric characters.

```
#toSpace<-content_transformer(function(x, pattern) gsub(pattern, " ", x))
data_dtm_Corp<-tm_map(data_dtm_Corp, tolower)
#inspect(data_dtm_Corp)
```

- to lower cases

```
data_dtm_Corp<-tm_map(data_dtm_Corp, removeNumbers)
#inspect(data_dtm_Corp)
```

- Removed numbers

```
data_dtm_Corp<-tm_map(data_dtm_Corp, removePunctuation)
#inspect(data_dtm_Corp)
```

- Remove punctuations.

```
data_dtm_Corp <- tm_map(data_dtm_Corp, removeWords, stopwords("english"))
#inspect(data_dtm_Corp)
```

- Remove English Stopwords.

```
data_dtm_Corp <- tm_map(data_dtm_Corp, stripWhitespace)
#inspect(data_dtm_Corp)
```

- Strip whitespace

```
data_dtm_Corp<-tm_map(data_dtm_Corp, stemDocument)
#inspect(data_dtm_Corp)
```

- Stemming

```
data_dtm_Corp <- tm_map(data_dtm_Corp, PlainTextDocument)
#inspect(data_dtm_Corp)
```

```
library(tm)
data_dtm_Corp[[1]]$meta$id <- "positive"
data_dtm_Corp[[2]]$meta$id <- "negative"
data_dtm_Corp[[3]]$meta$id <- "neutral"

data_dtm <-DocumentTermMatrix(data_dtm_Corp)
#head(t(as.matrix(data_dtm)))
inspect(data_dtm)
```

```
## <<DocumentTermMatrix (documents: 3, terms: 6521)>>
## Non-/sparse entries: 9884/9679
## Sparsity           : 49%
## Maximal term length: 22
## Weighting          : term frequency (tf)
## Sample             :
##           Terms
## Docs       compani eur finnish million oper said sale share will year
##   negative     592 318     220     193  248  238  197   307  595  114
##   neutral      273 597     203     171  215  230  204    84  186  231
##   positive     108 400     102      78  116   77  102    50   69  104
```

- Explore DTM – Frequent terms

```
findFreqTerms(data_dtm, lowfreq= 400, highfreq= Inf)
```

```
##  [1] "compani" "eur"     "finnish" "million" "net"     "oper"    "profit" 
##  [8] "said"    "sale"    "share"   "will"    "year"
```

– Remove sparse terms

```
data_dtm <- removeSparseTerms(data_dtm, 0.9)
inspect(data_dtm)
```

```
## <<DocumentTermMatrix (documents: 3, terms: 6521)>>
## Non-/sparse entries: 9884/9679
## Sparsity           : 49%
## Maximal term length: 22
## Weighting          : term frequency (tf)
## Sample             :
##           Terms
## Docs       compani eur finnish million oper said sale share will year
##   negative     592 318     220     193  248  238  197   307  595  114
##   neutral      273 597     203     171  215  230  204    84  186  231
##   positive     108 400     102      78  116   77  102    50   69  104
```

```
library(Rgraphviz)
```

```
## Loading required package: graph
```

```
## Loading required package: BiocGenerics
```

```
## 
## Attaching package: 'BiocGenerics'
```

```
## The following object is masked from 'package:NLP':
## 
##     annotation
```

```
## The following objects are masked from 'package:dplyr':
## 
##     combine, intersect, setdiff, union
```

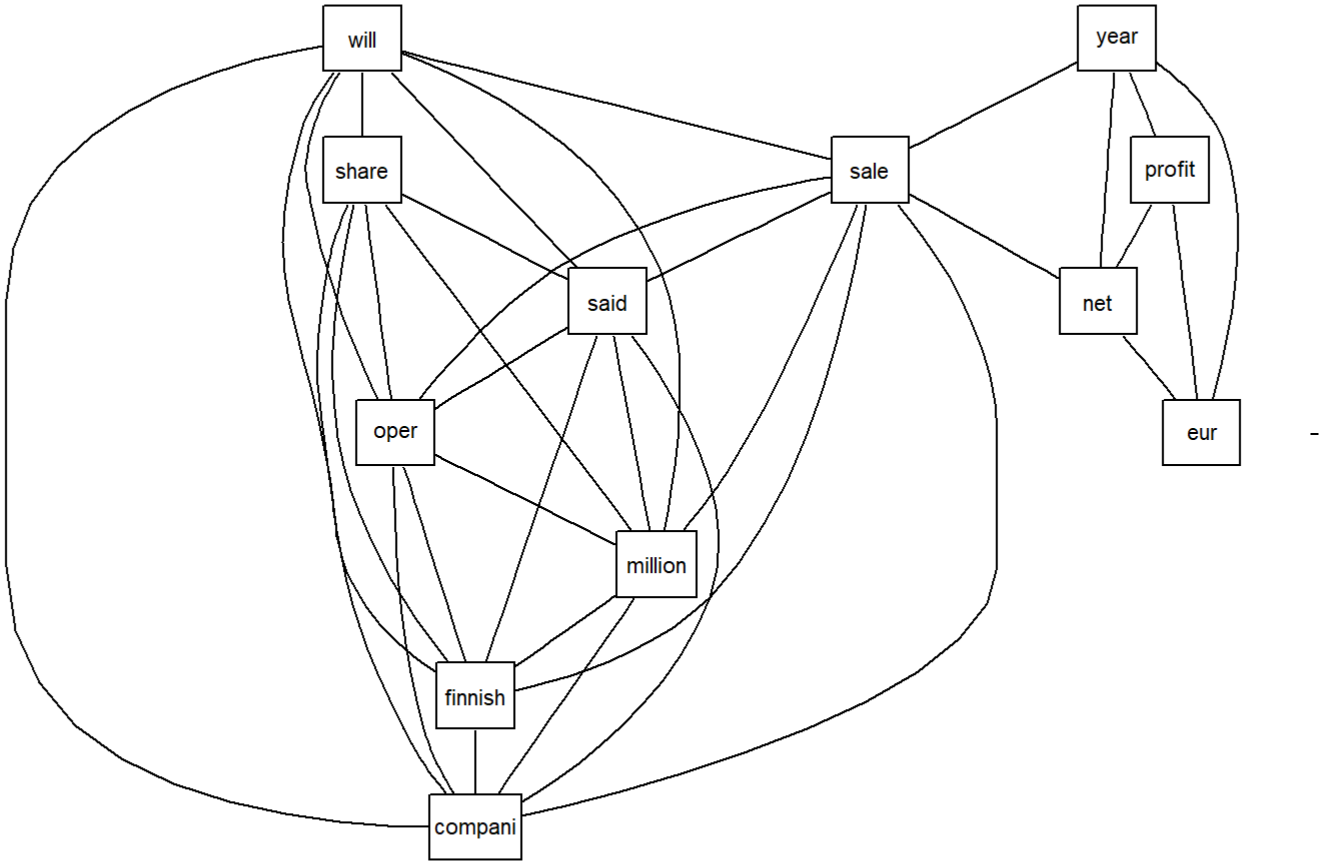
```
## The following objects are masked from 'package:stats':
## 
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

```
##
## Attaching package: 'graph'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```
## Loading required package: grid
```

```
plot(data_dtm,terms=findFreqTerms(data_dtm, lowfreq=400),corThreshold=0.6)
```



tibble for the text

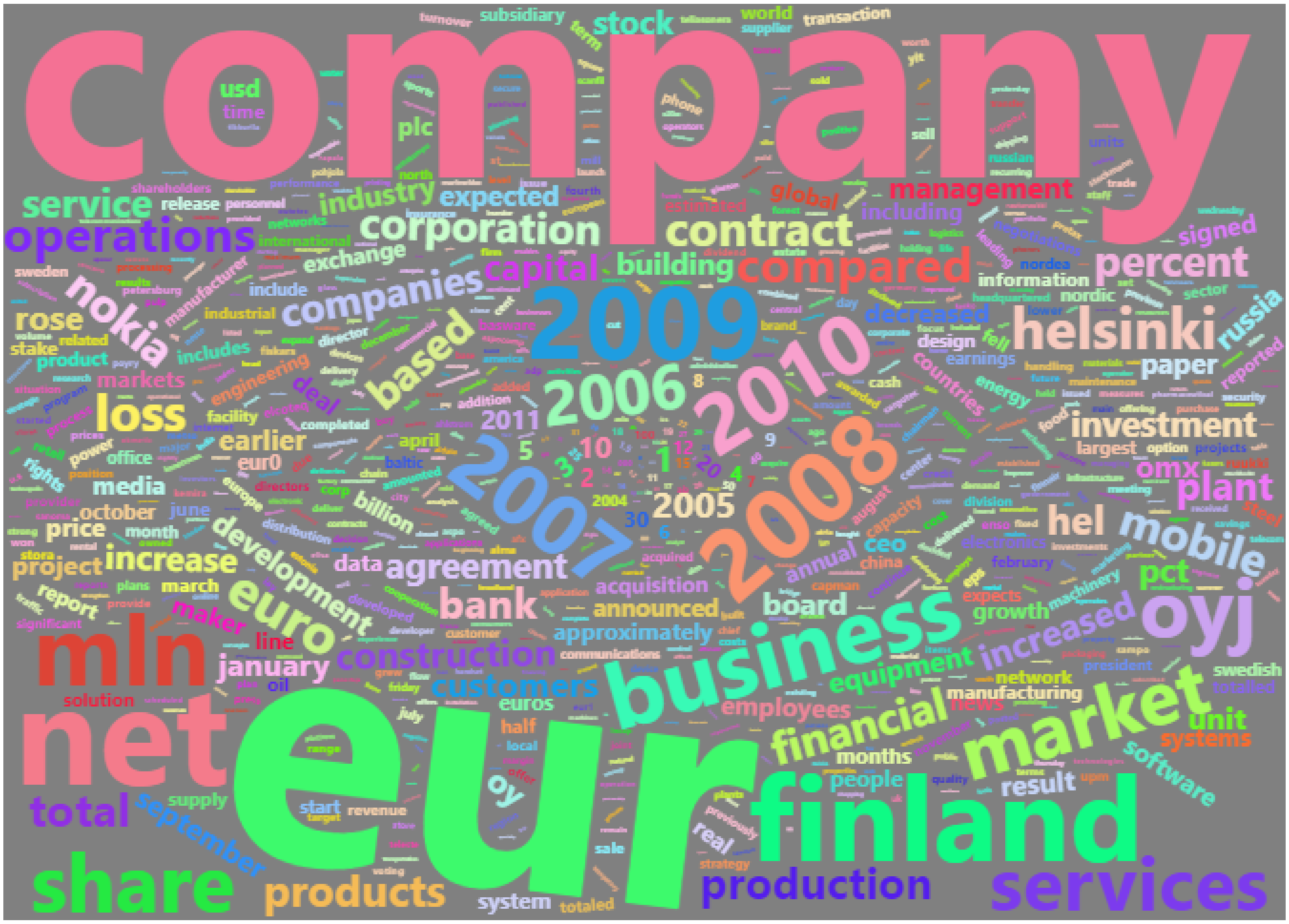
```
#data_df = tibble(id=data$Sentiment,line=1:length(data_dtm_Corp[[1]]$content),text = amz_all_sectors_Corp$content$content)

#gandhi_df_1 = tibble(id=corpus_All[[1]]$meta$id,line=1:length(corpus_All[[1]]$content),text = corpus_All[[

#all_sectors_df


corpus_All<- data$text %>% VectorSource() %>% VCorpus()


corpus_All <- tm_map(corpus_All,f)

corpus_All<-tm_map(corpus_All, tolower)

corpus_All<-tm_map(corpus_All, removeNumbers)

corpus_All<-tm_map(corpus_All, removePunctuation)

corpus_All <- tm_map(corpus_All, removeWords, stopwords("english"))

corpus_All <- tm_map(corpus_All, stripWhitespace)

corpus_All<-tm_map(corpus_All, stemDocument)

corpus_All <- tm_map(corpus_All, PlainTextDocument)

#inspect(corpus_All)
data_df = tibble(id=data$Sentiment,line=1:length(data$text),text = data$text)
data_df_unnest <- data_df %>% unnest_tokens(word,text)
```

- WordCloud for all text

```
data_td <- data_df_unnest %>% anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
t.data<- data.frame(data_td %>% count(word))

wordcloud2(t.data, minSize = 2,color = "random-light", backgroundColor = "grey")
```



- WordCloud for positive text

```
t.positive <- data.frame(data_td %>% filter(id == "positive") %>% count(word))

wordcloud2(t.positive, minSize = 20,color = "random-light", backgroundColor = "grey")
```

- WordCloud for negative text

```
t.negative <- data.frame(data_td %>% filter(id == "negative") %>% count(word))

wordcloud2(t.negative, minSize = 20,color = "random-light", backgroundColor = "grey")
```

- WordCloud for neutral text

```
t.neutral <- data.frame(data_td %>% filter(id == "neutral") %>% count(word))

wordcloud2(t.neutral, minSize = 20,color = "random-light", backgroundColor = "grey")
```

Sentiment Analysis

– Load sentiments

```
library(syuzhet)
```

```
##
## Attaching package: 'syuzhet'
```

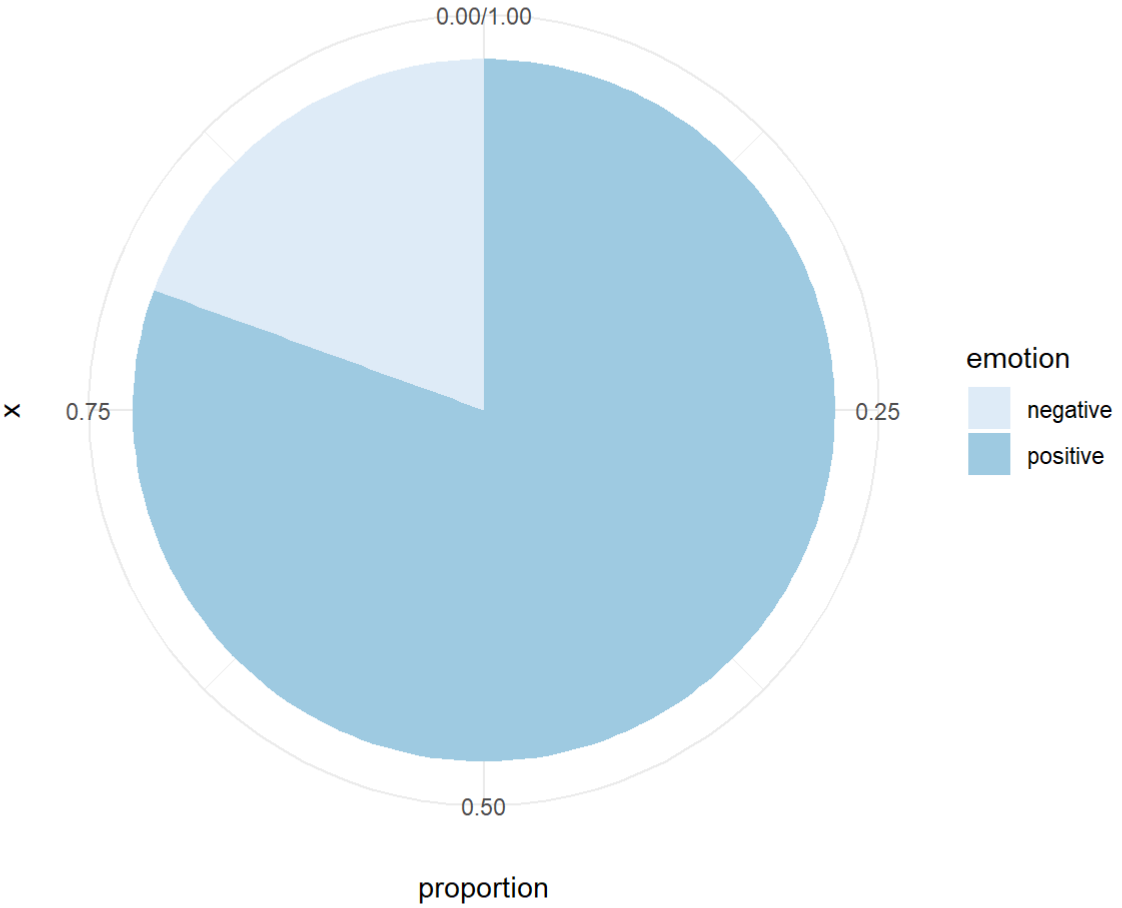
```
## The following object is masked from 'package:scales':
##
##     rescale
```

```
bing <- get_sentiments("bing")
afinn <- get_sentiments("afinn")
nrc_positive <- get_sentiments("nrc") %>% filter(sentiment == "positive")
nrc_negative <- get_sentiments("nrc") %>% filter(sentiment == "negative")
nrc_surprise <- get_sentiments("nrc") %>% filter(sentiment == "surprise")
nrc_anticipation <- get_sentiments("nrc") %>% filter(sentiment == "anticipation")
nrc_disgust <- get_sentiments("nrc") %>% filter(sentiment == "disgust")
nrc_fear <- get_sentiments("nrc") %>% filter(sentiment == "fear")
nrc_sadness <- get_sentiments("nrc") %>% filter(sentiment == "sadness")
nrc_anger <- get_sentiments("nrc") %>% filter(sentiment == "anger")
nrc_joy <- get_sentiments("nrc") %>% filter(sentiment == "joy")
nrc_trust <- get_sentiments("nrc") %>% filter(sentiment == "trust")

nrc_data <- get_nrc_sentiment(data_td$word)
head(nrc_data)
```

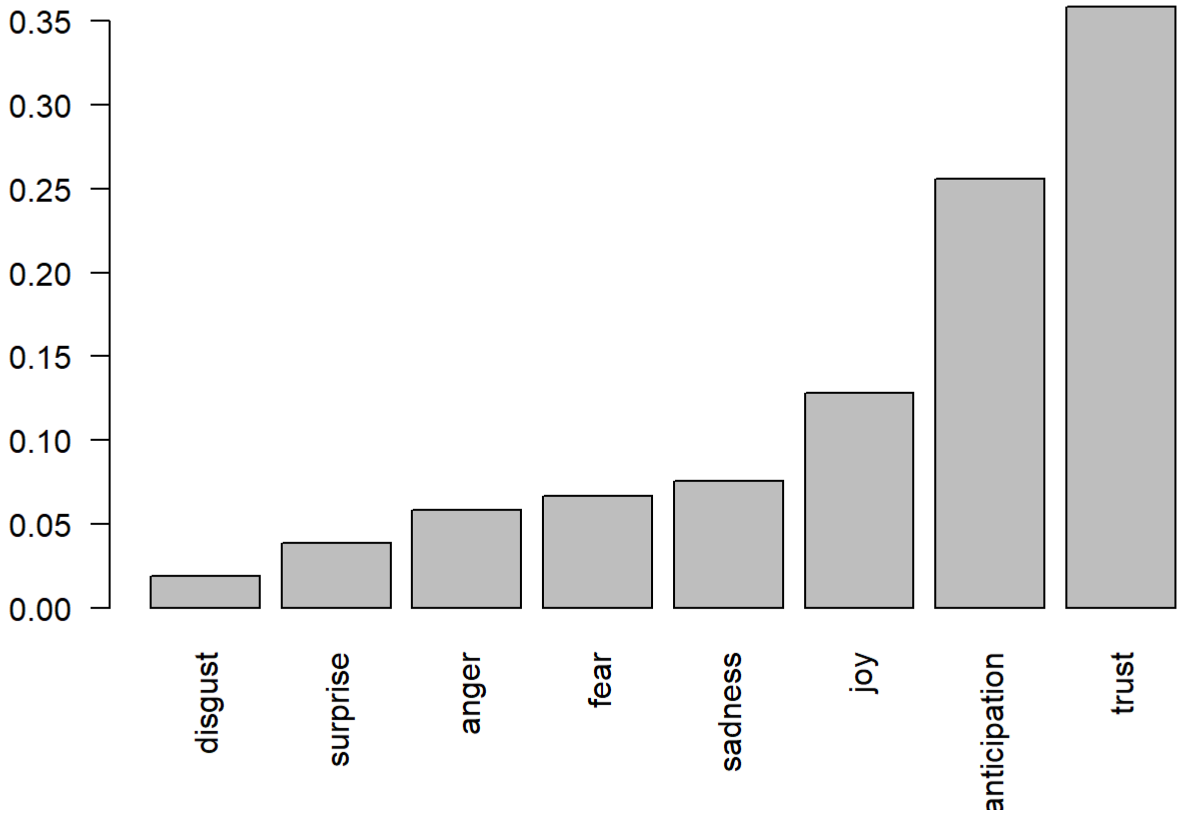
	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	negative ▶
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0

6 rows | 1-10 of 11 columns

```
#library(ggplot2)
pie_graph_data <- data.frame(emotion = names(sort(colSums(prop.table(nrc_data[, 9:10])))), proportion = colSums(prop.table(nrc_data[, 9:10])))
ggplot(pie_graph_data, aes(x="", y=proportion, fill=emotion))+geom_bar(width = 1, stat = "identity")+coord_polar("y", start=0)+scale_fill_brewer(palette="Blues")+theme_minimal()
```



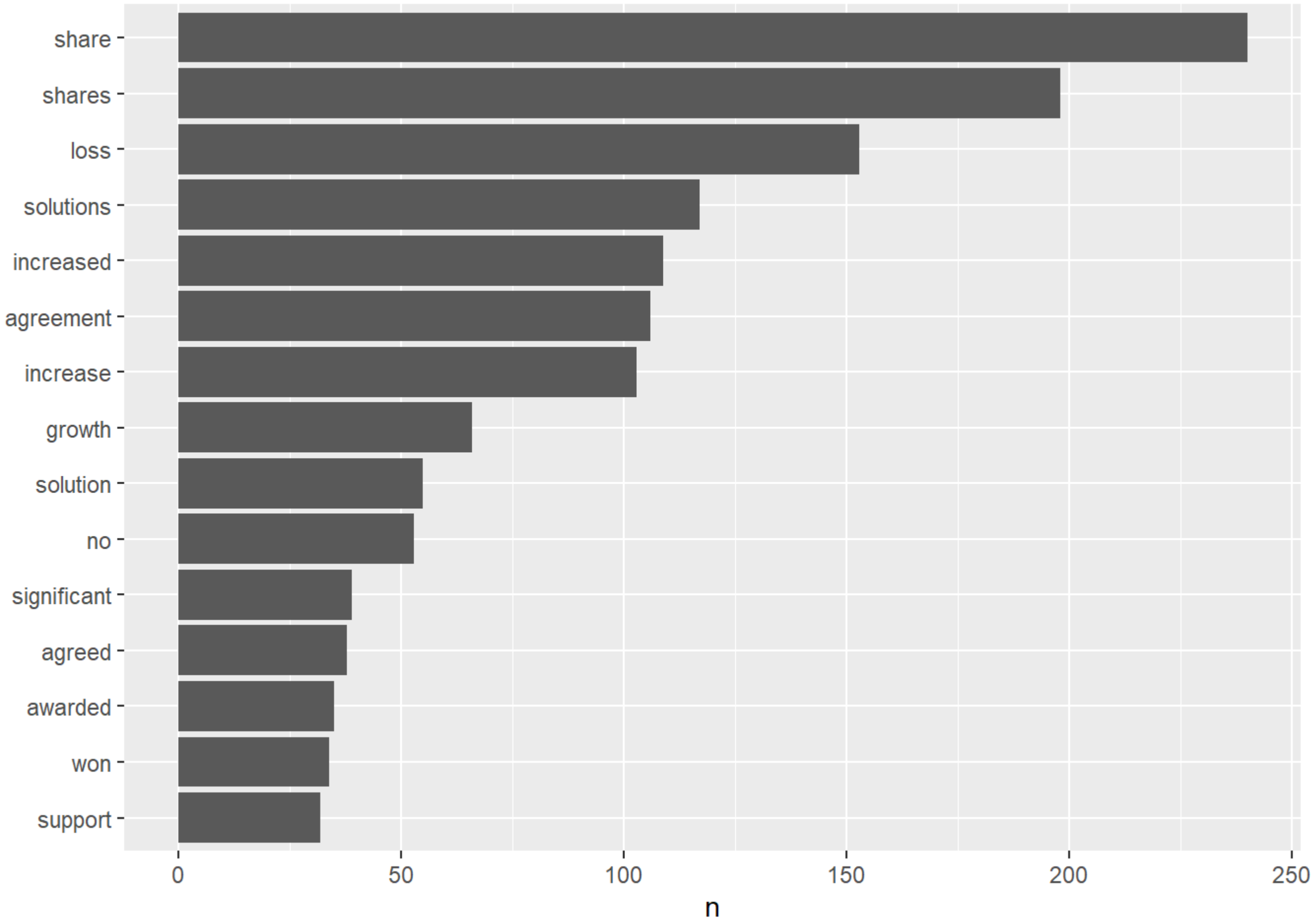
```
barplot<-data.frame(emotion = sort(colSums(prop.table(nrc_data[,1:8]))))
barplot(barplot$emotion,names =row.names(barplot),las=2)
```



```
TRY1_data<- data_df %>% unnest_tokens(word, text) %>% inner_join(afinn)%>% count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

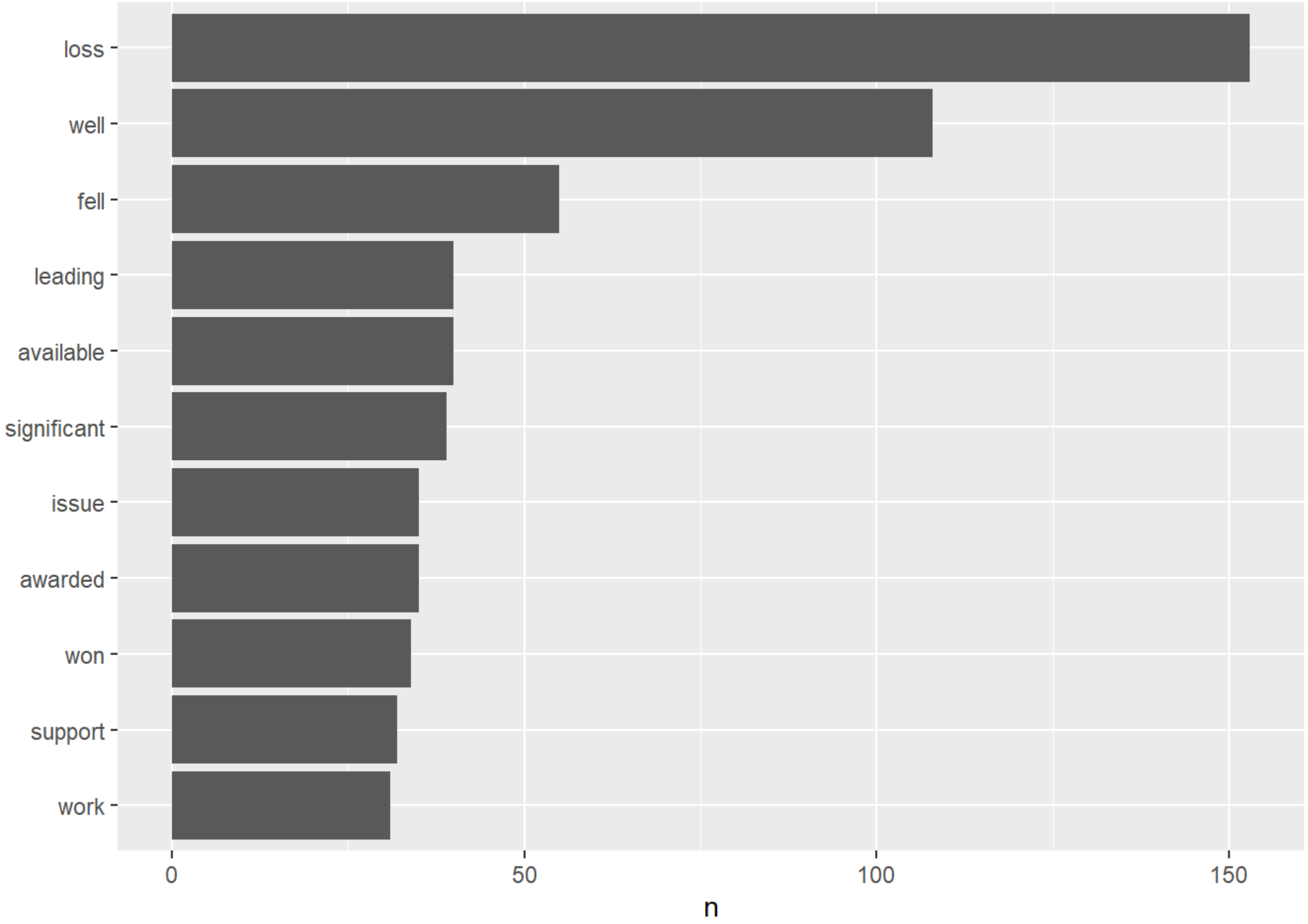
```
TRY1_data %>% filter(n > 30) %>% mutate(word = reorder(word, n)) %>% ggplot(aes(word, n)) + geom_col() + xlab(NULL) + coord_flip()
```



```
TRY2_data <- data_df %>% unnest_tokens(word, text) %>% inner_join(bing) %>% count(word, sort = T
RUE)
```

```
## Joining, by = "word"
```

```
TRY2_data %>% filter(n > 30) %>% mutate(word = reorder(word, n)) %>% ggplot(aes(word, n)) + geom
_col() + xlab(NULL) + coord_flip()
```

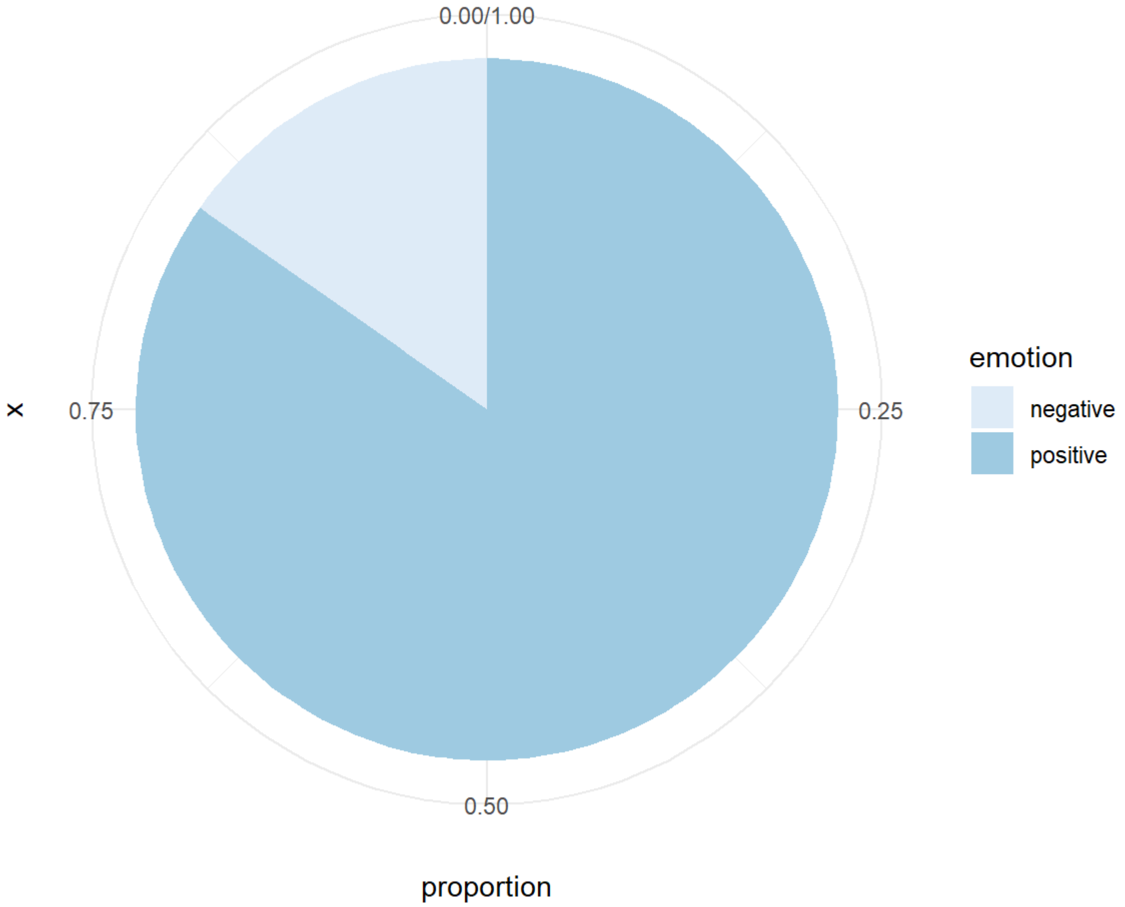


- Sentimental analysis by Positive, negative, neutral

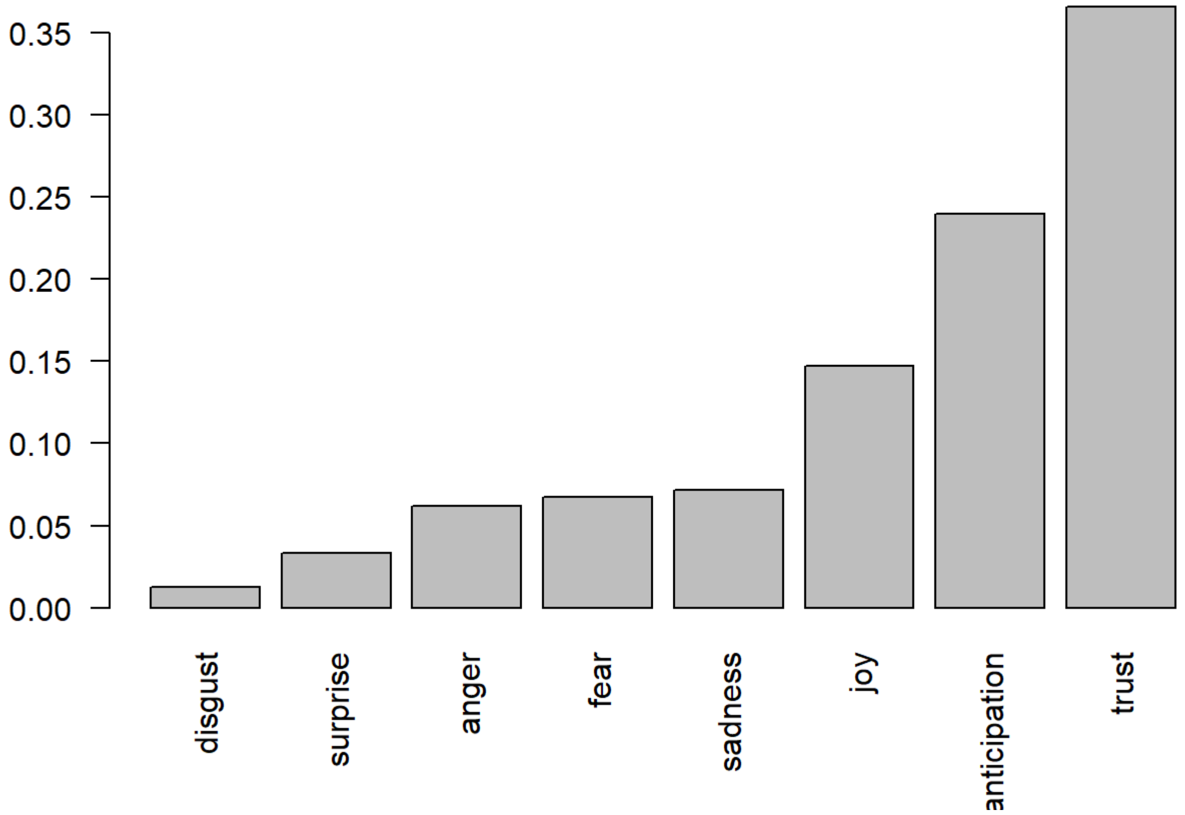
```
data_positive_td <- data_td %>% filter(id=="positive")
data_negative_td <- data_td %>% filter(id=="negative")
data_neutral_td <- data_td %>% filter(id=="neutral")

nrc_positive_data <- get_nrc_sentiment(data_positive_td$word)
nrc_negative_data <- get_nrc_sentiment(data_negative_td$word)
nrc_neutral_data <- get_nrc_sentiment(data_neutral_td$word)
```

1. Positive data composition

```
#library(ggplot2)
pie_graph_positive_data <- data.frame(emotion = names(sort(colSums(prop.table(nrc_positive_data
[, 9:10])))), proportion = colSums(prop.table(nrc_positive_data[, 9:10])))
ggplot(pie_graph_positive_data, aes(x="", y=proportion, fill=emotion))+geom_bar(width = 1, stat
 = "identity")+coord_polar("y", start=0)+scale_fill_brewer(palette="Blues")+theme_minimal()
```



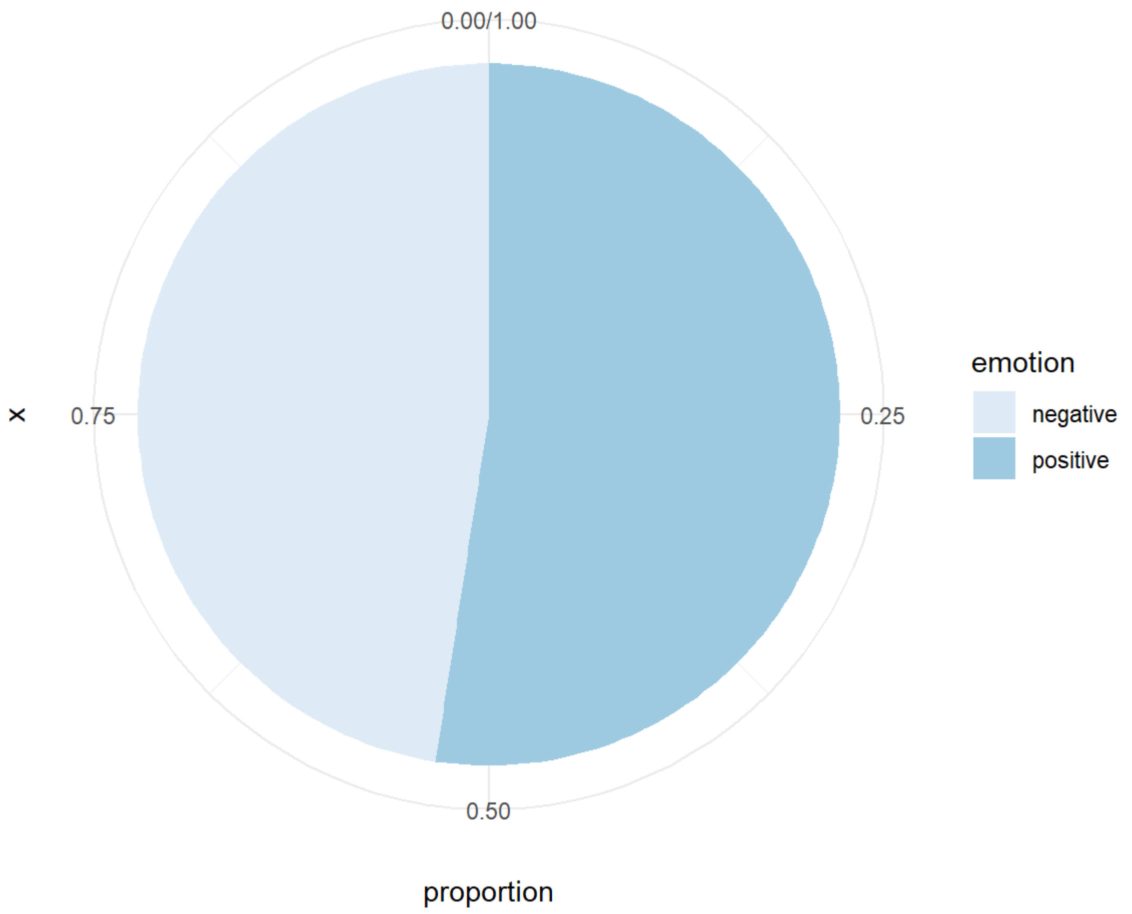
```
barplot_positive <-data.frame(emotion = sort(colSums(prop.table(nrc_positive_data[,1:8]))))
barplot(barplot_positive$emotion,names =row.names(barplot),las=2)
```



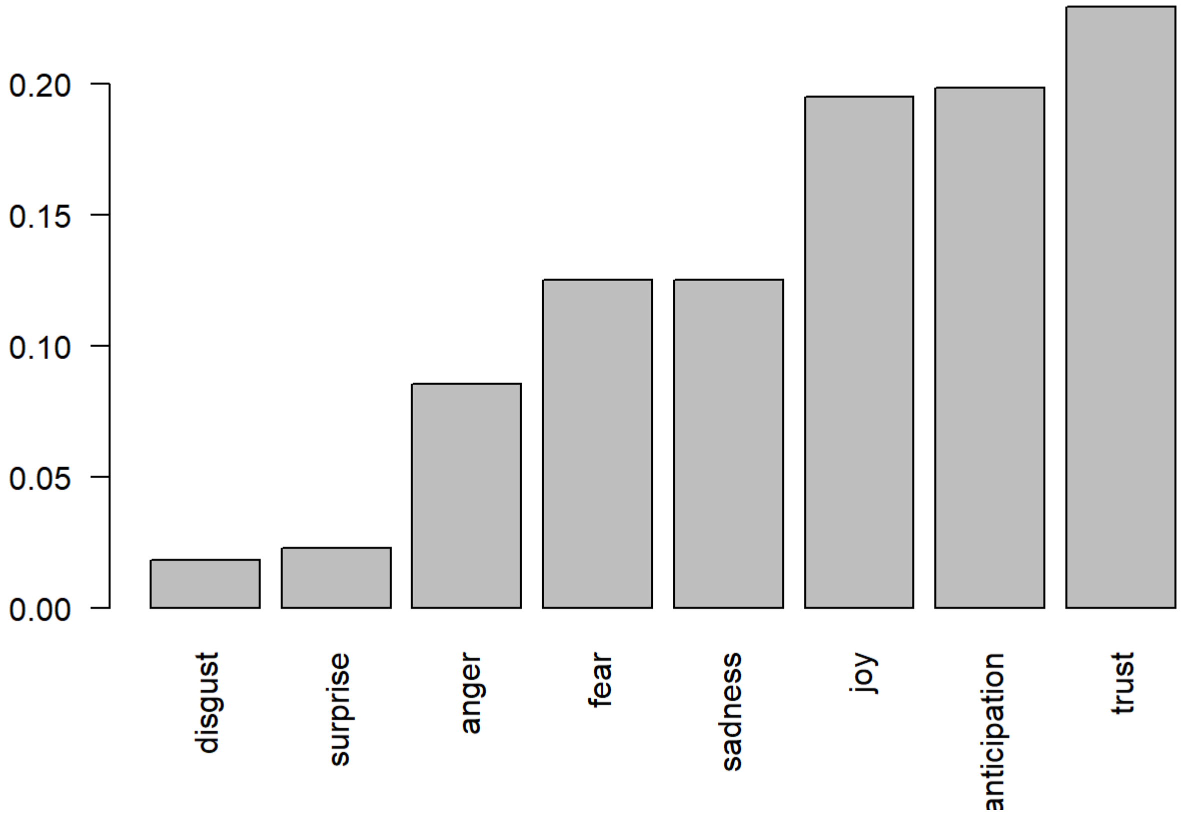
2.

Negative data composition

```
#library(ggplot2)
pie_graph_negative_data <- data.frame(emotion = names(sort(colSums(prop.table(nrc_negative_data[, 9:10])))), proportion = colSums(prop.table(nrc_negative_data[, 9:10])))
ggplot(pie_graph_negative_data, aes(x="", y=proportion, fill=emotion))+geom_bar(width = 1, stat = "identity")+coord_polar("y", start=0)+scale_fill_brewer(palette="Blues")+theme_minimal()
```



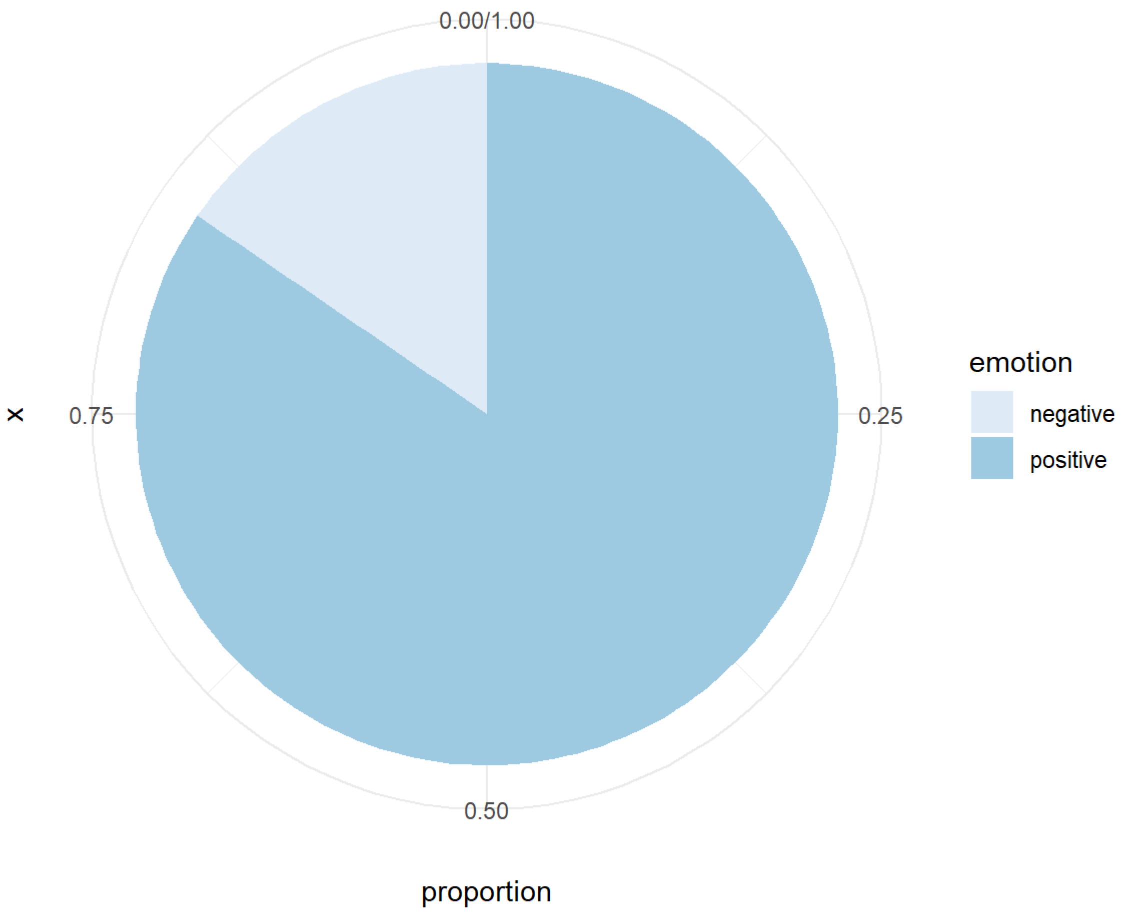
```
barplot_negative <-data.frame(emotion = sort(colSums(prop.table(nrc_negative_data[,1:8]))))
barplot(barplot_negative$emotion,names =row.names(barplot),las=2)
```



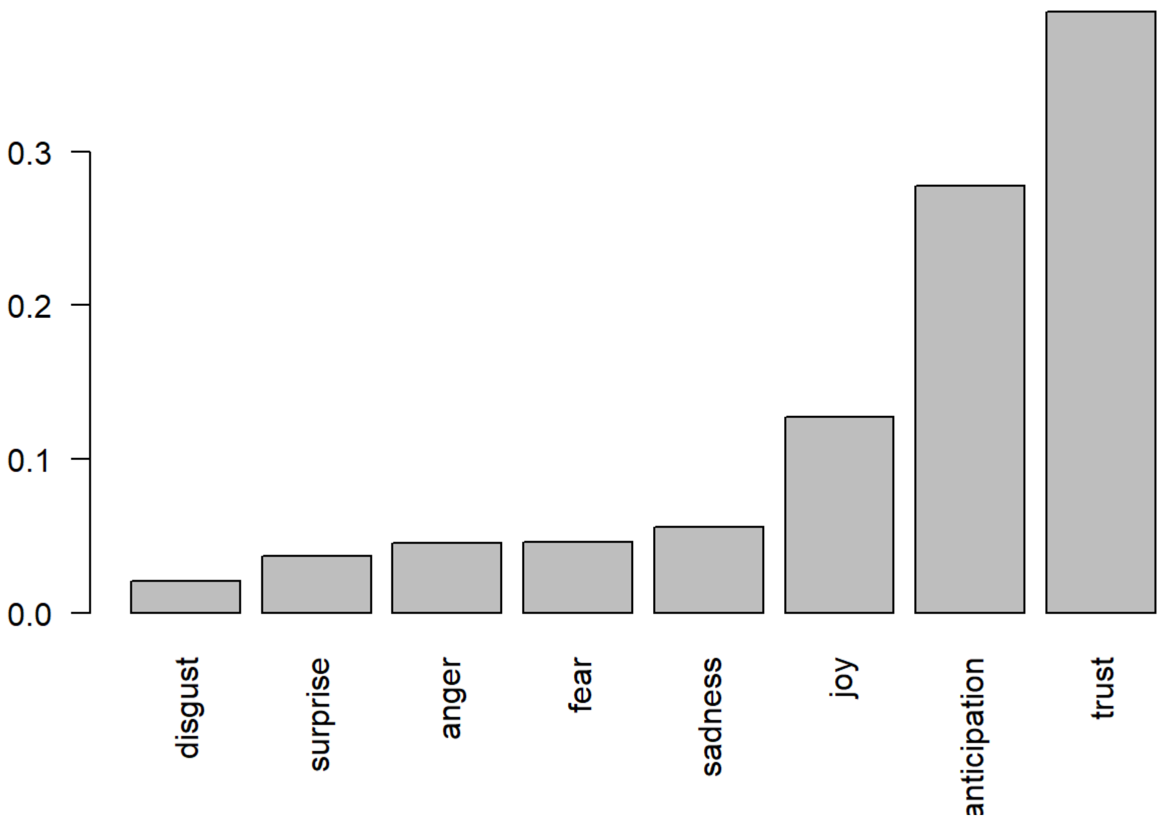
3.

Neutral data composition

```
#library(ggplot2)
pie_graph_neutral_data <- data.frame(emotion = names(sort(colSums(prop.table(nrc_neutral_data[,
9:10])))), proportion = colSums(prop.table(nrc_neutral_data[, 9:10])))
ggplot(pie_graph_neutral_data, aes(x="", y=proportion, fill=emotion))+geom_bar(width = 1, stat =
"identity")+coord_polar("y", start=0)+scale_fill_brewer(palette="Blues")+theme_minimal()
```



```
barplot_neutral <-data.frame(emotion = sort(colSums(prop.table(nrc_neutral_data[,1:8]))))
barplot(barplot_neutral$emotion,names =row.names(barplot),las=2)
```


0.3
0.2
0.1
0.0
disgust
surprise
anger
fear
sadness
joy
anticipation
trust