QUESTION 5: -

QUESTION 5: - TEXT ANALYTICS – SENTIMENT ANALYTICS USING R STUDIO AND WORDCLOUD

Word clouds can be a useful visualisation to quickly view the most used words in a block of text. R has the wordcloud function in the wordcloud package that can be used to quickly generate these plots. Sentiment analysis is an important part of the emotion computing and wordcloud is a fancy way of text visualization. Combining the two, we can reveal and display people's attitude and perspectives through their comments or articles.

CODE (R-STUDIO) : -

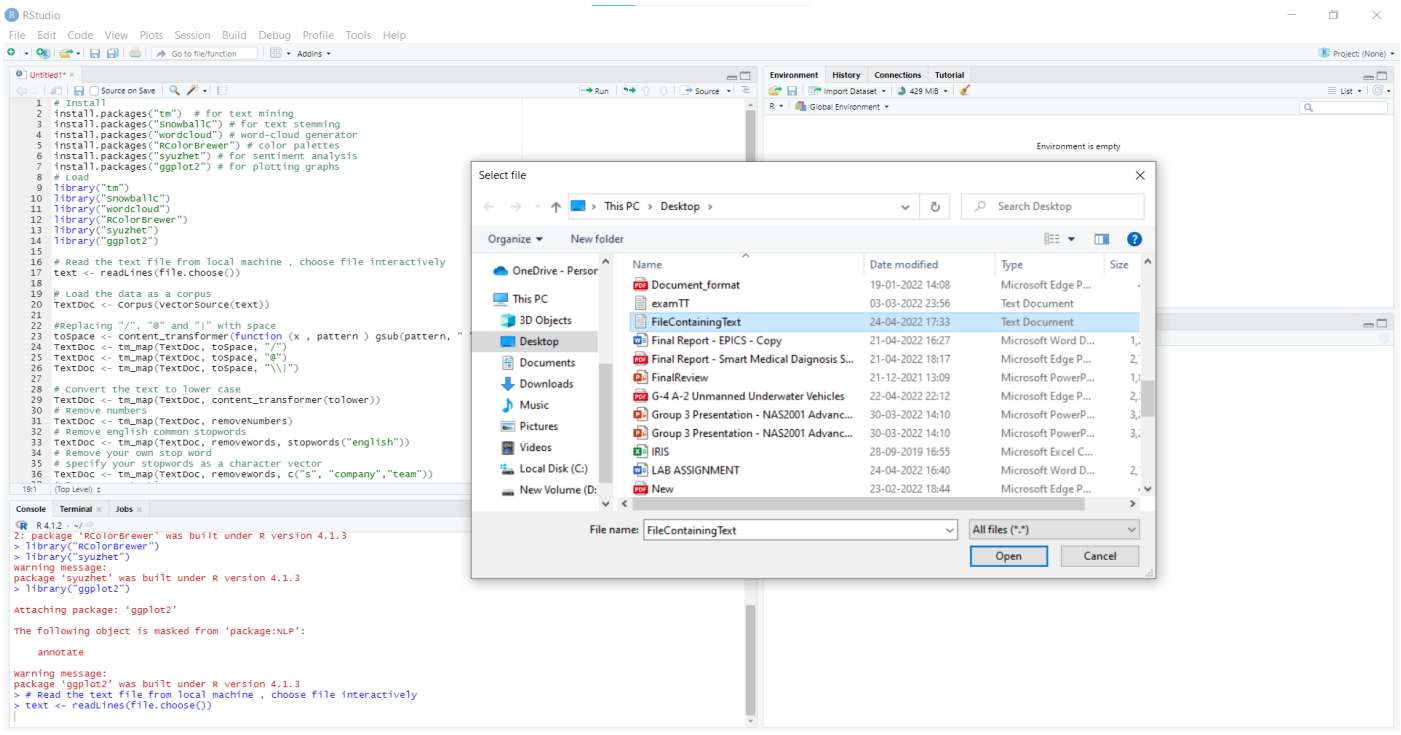
```
# Install
install.packages("tm")  # for text mining
install.packages("SnowballC") # for text stemming
install.packages("wordcloud") # word-cloud generator
install.packages("RColorBrewer") # color palettes
install.packages("syuzhet") # for sentiment analysis
install.packages("ggplot2") # for plotting graphs
# Load
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("syuzhet")
library("ggplot2")
# Read the text file from local machine , choose file interactively
text <- readLines(file.choose())
# Load the data as a corpus
TextDoc <- Corpus(VectorSource(text))
#Replacing "/", "@" and "|" with space
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc <- tm_map(TextDoc, toSpace, "/")
TextDoc <- tm_map(TextDoc, toSpace, "@")
TextDoc <- tm_map(TextDoc, toSpace, "\\|")
# Convert the text to lower case
TextDoc <- tm_map(TextDoc, content_transformer(tolower))
# Remove numbers
TextDoc <- tm_map(TextDoc, removeNumbers)
# Remove english common stopwords
TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
# Remove your own stop word
# specify your stopwords as a character vector
TextDoc <- tm_map(TextDoc, removeWords, c("s", "company","team"))
# Remove punctuations
TextDoc <- tm_map(TextDoc, removePunctuation)
# Eliminate extra white spaces
```

```
TextDoc <- tm_map(TextDoc, stripWhitespace)
# Text stemming - which reduces words to their root form
TextDoc <- tm_map(TextDoc, stemDocument)
# Build a term-document matrix
TextDoc_dtm <- TermDocumentMatrix(TextDoc)
dtm_m <- as.matrix(TextDoc_dtm)
# Sort by descearing value of frequency
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
# Display the top 5 most frequent words
head(dtm_d, 5)
# Plot the most frequent words
barplot(dtm_d[1:5,]$freq, las = 2, names.arg = dtm_d[1:5,]$word,
        col ="lightgreen", main ="Top 5 most frequent words",
        ylab = "Word frequencies")
#generate word cloud
set.seed(1234)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 5,
          max.words=100, random.order=FALSE, rot.per=0.40,
          colors=brewer.pal(8, "Dark2"))
# Word Association :
# Find associations
findAssocs(TextDoc_dtm, terms = c("good","work","health"), corlimit = 0.25)
# Find associations for words that occur at least 50 times
findAssocs(TextDoc_dtm, terms = findFreqTerms(TextDoc_dtm, lowfreq = 50), corlimit = 0.25)
# possibly creat a heat map ?
# regular sentiment score using get_sentiment() function and method of your choice
# please note that different methods have different scales
syuzhet_vector <- get_sentiment(text, method="syuzhet")
# see the first 10 elements of the vector
head(syuzhet_vector,10)
# see median value of vector elements
# median(syuzhet_vector)
summary(syuzhet_vector)
# bing
bing_vector <- get_sentiment(text, method="bing")
head(bing_vector)
summary(bing_vector)
#afinn
afinn_vector <- get_sentiment(text, method="afinn")
head(afinn_vector)
summary(afinn_vector)
#nrc
nrc_vector <- get_sentiment(text, method="nrc")
head(nrc_vector)
median(nrc_vector)
#compare the first row of each vector using sign function
rbind(
  sign(head(syuzhet_vector)),
  sign(head(bing_vector)),
```

```
sign(head(afinn_vector))
)
# head(d,10) - just to see top 10 lines
# run nrc sentiment analysis to return data frame with each row classified as one of the following
# emotions, rather than a score :
# anger, anticipation, disgust, fear, joy, sadness, surprise, trust
# and if the sentiment is positive or negative
d<-get_nrc_sentiment(text)
head (d,10)
#transpose
td<-data.frame(t(d))
#The function rowSums computes column sums across rows for each level of a grouping variable.
td_new <- data.frame(rowSums(td[2:253]))
#Transformation and cleaning
names(td_new)[1] <- "count"
td_new <- cbind("sentiment" = rownames(td_new), td_new)
rownames(td_new) <- NULL
td_new2<-td_new[1:8,]
#Plot 1 - count of words associated with each sentiment
quickplot(sentiment, data=td_new2, weight=count,
geom="bar",fill=sentiment,ylab="count")+ggtitle("Survey sentiments")
#Plot 2 - count of words associated with each sentiment, expressed as a percentage
barplot(
  sort(colSums(prop.table(d[, 1:8]))),
  horiz = TRUE,
  cex.names = 0.7,
  las = 1,
  main = "Emotions in Text", xlab="Percentage"
)
```

OUTPUTS

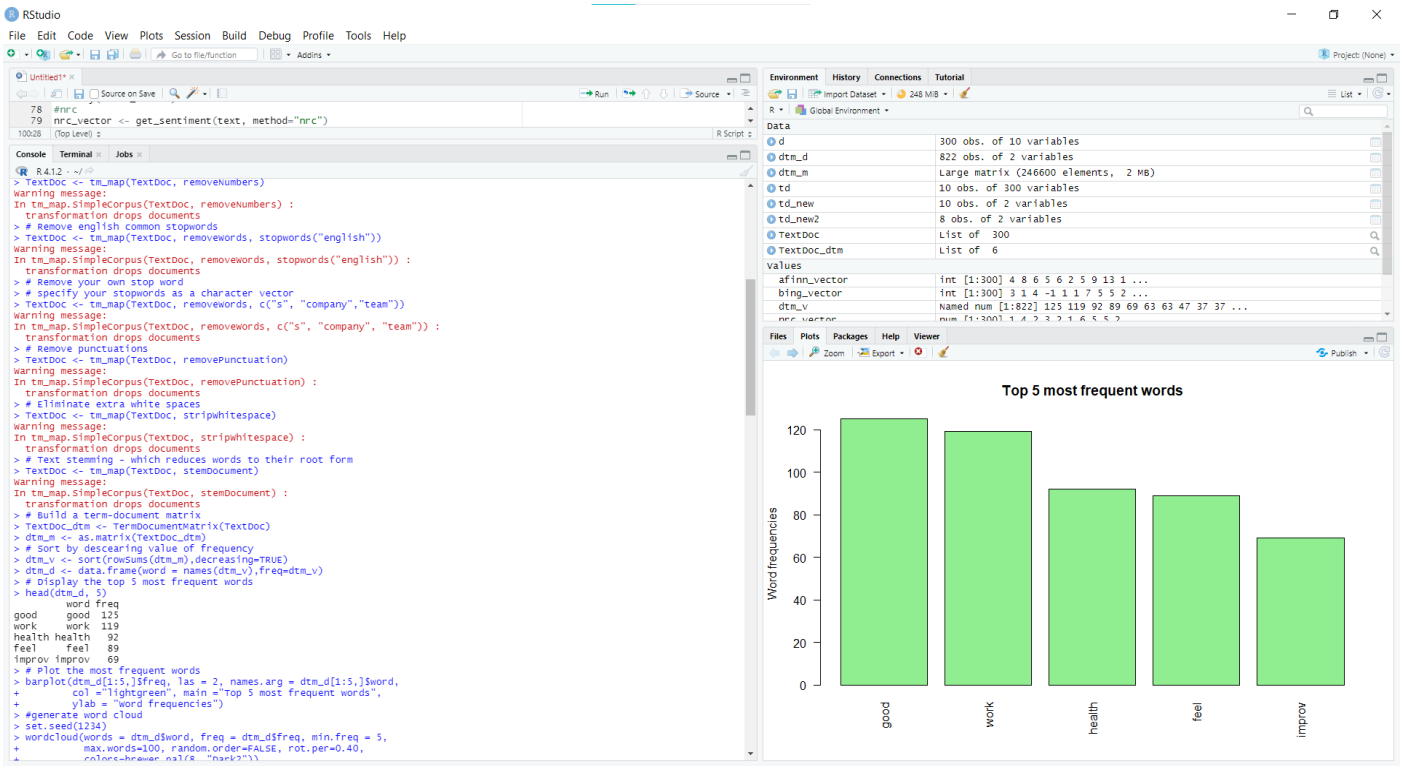
Selecting the text file and loading it to the variable in R Studio: -



After cleaning the text data, the next step is to count the occurrence of each word, to identify popular or trending topics. Using the function TermDocumentMatrix() from the text mining package, you can build a Document Matrix – a table containing the frequency of words.

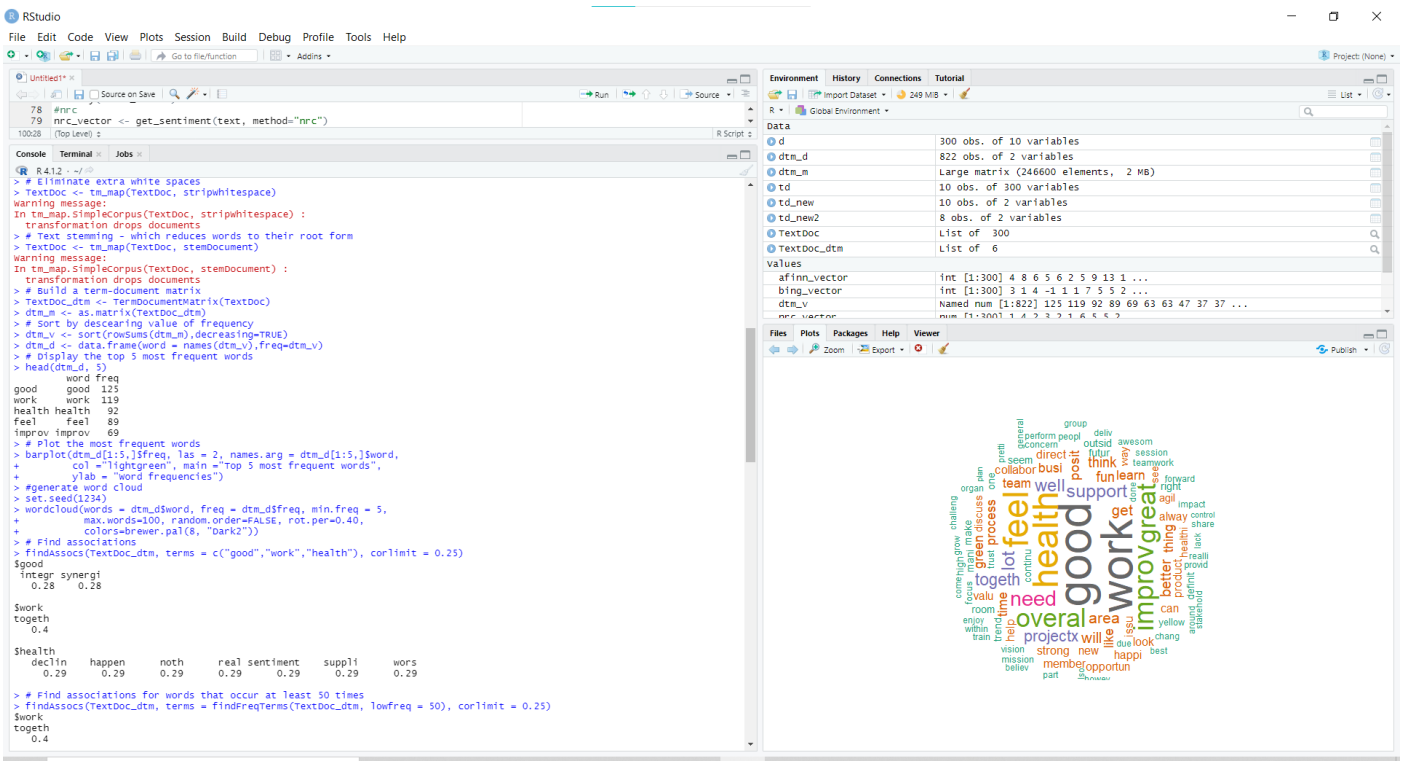
```
> TextDoc <- tm_map(TextDoc, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(TextDoc, removeNumbers) :
  transformation drops documents
> # Remove english common stopwords
> TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(TextDoc, removeWords, stopwords("english")) :
  transformation drops documents
> # Remove your own stop word
> # specify your stopwords as a character vector
> TextDoc <- tm_map(TextDoc, removeWords, c("s", "company","team"))
Warning message:
In tm_map.SimpleCorpus(TextDoc, removeWords, c("s", "company", "team")) :
  transformation drops documents
> # Remove punctuations
> TextDoc <- tm_map(TextDoc, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(TextDoc, removePunctuation) :
  transformation drops documents
> # Eliminate extra white spaces
> TextDoc <- tm_map(TextDoc, stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(TextDoc, stripWhitespace) :
  transformation drops documents
> # Text stemming - which reduces words to their root form
> TextDoc <- tm_map(TextDoc, stemDocument)
Warning message:
In tm_map.SimpleCorpus(TextDoc, stemDocument) :
  transformation drops documents
> # Build a term-document matrix
> TextDoc_dtm <- TermDocumentMatrix(TextDoc)
> dtm_m <- as.matrix(TextDoc_dtm)
> # Sort by descearing value of frequency
> dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
> dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
> # Display the top 5 most frequent words
> head(dtm_d, 5)
          word freq
good      good  125
work      work  119
health health   92
feel      feel   89
improv improv   69
```

Top 5 Frequent Words: -



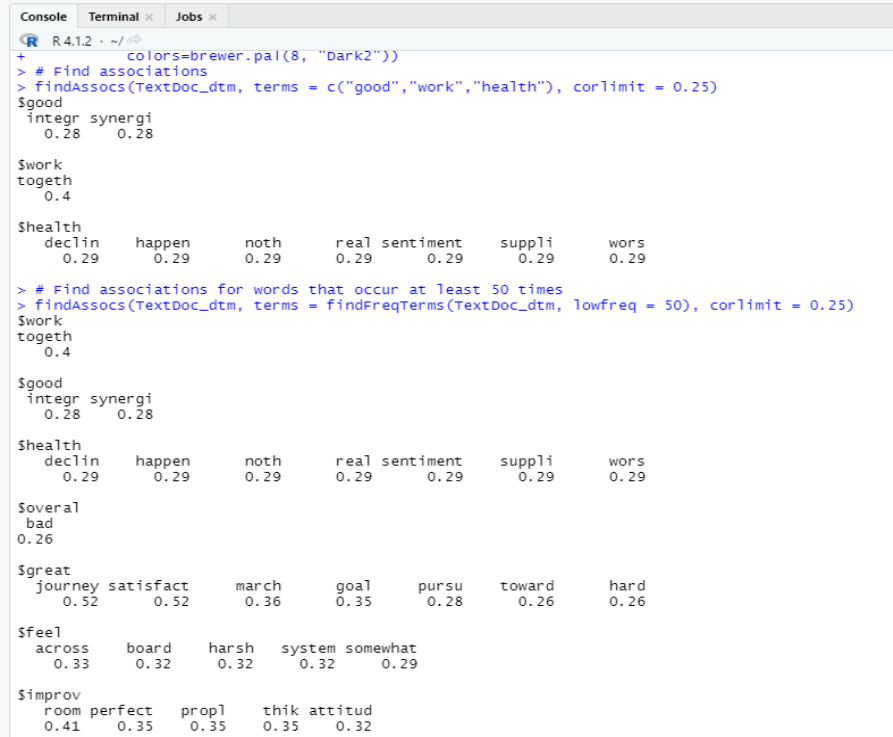
Generate the Word Cloud

A word cloud is one of the most popular ways to visualize and analyze qualitative data. It's an image composed of keywords found within a body of text, where the size of each word indicates its frequency in that body of text. Use the word frequency data frame (table) created previously to generate the word cloud.



Word Association

Correlation is a statistical technique that can demonstrate whether, and how strongly, pairs of variables are related. This technique can be used effectively to analyze which words occur most often in association with the most frequently occurring words in the survey responses, which helps to see the context around these words.



Sentiment Scores

Sentiments can be classified as positive, neutral or negative. They can also be represented on a numeric scale, to better express the degree of positive or negative strength of the sentiment contained in a body of text.

```
  0.41    0.35    0.35    0.35    0.32

> # regular sentiment score using get_sentiment() function and method of your choice
> # please note that different methods have different scales
> syuzhet_vector <- get_sentiment(text, method="syuzhet")
> # see the first 10 elements of the vector
> head(syuzhet_vector,10)
 [1] 2.60 4.65 2.55 1.05 1.00 0.25 6.60 3.90 4.40 1.20
> # see median value of vector elements
> # median(syuzhet_vector)
> summary(syuzhet_vector)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.450   0.900   1.600   1.883   2.650   9.000
```

Please note the scale of sentiment scores generated by:

- **bing** – binary scale with -1 indicating negative and +1 indicating positive sentiment
- **afinn** – integer scale ranging from -5 to +5

The summary statistics of `bing` and `afinn` vectors also show that the `Median` value of Sentiment scores is above 0 and can be interpreted as the overall average sentiment across the all the responses is positive.

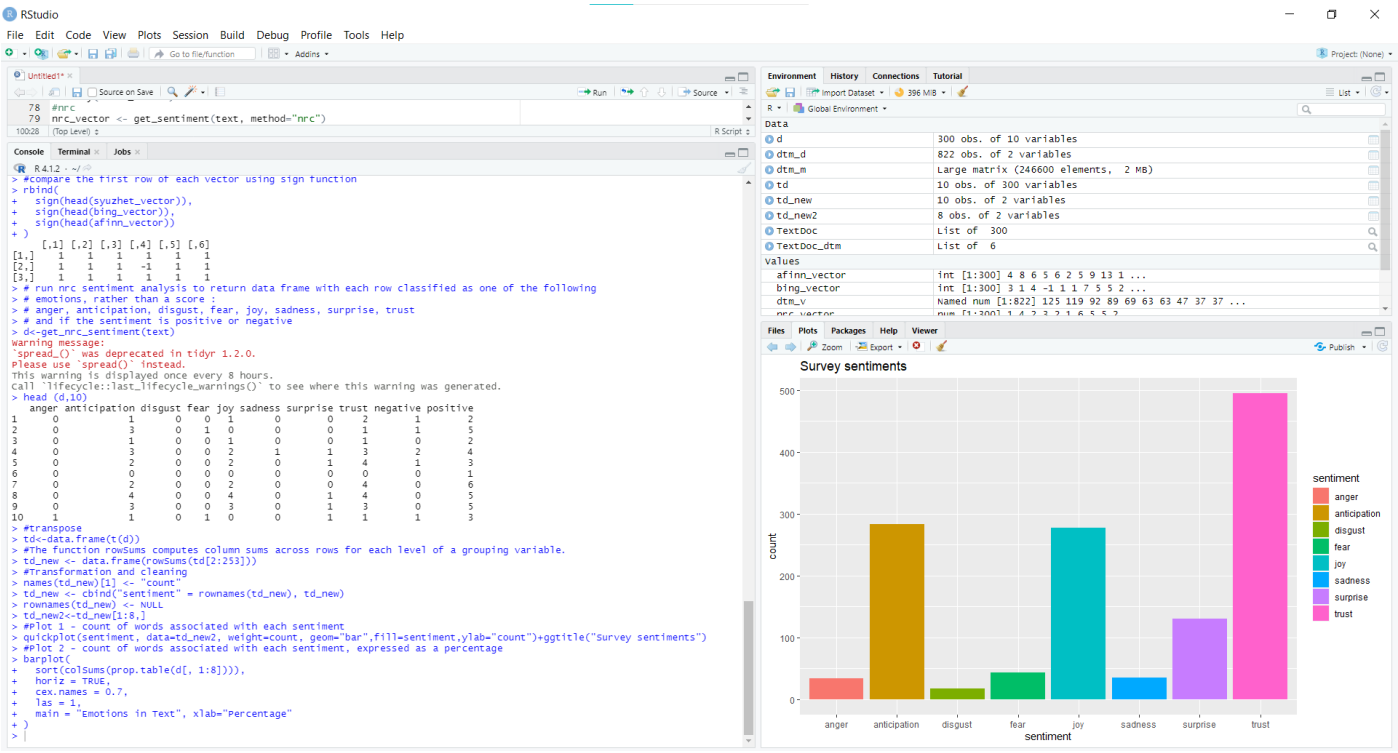
```
> # bing
> bing_vector <- get_sentiment(text, method="bing")
> head(bing_vector)
[1]  3  1  4 -1  1  1
> summary(bing_vector)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -3.000   1.000   2.000   2.007   3.000   9.000
> #afinn
> afinn_vector <- get_sentiment(text, method="afinn")
> head(afinn_vector)
[1] 4 8 6 5 6 2
> summary(afinn_vector)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -6.00    2.00    4.00    4.42    7.00   18.00
> #nrc
> nrc_vector <- get_sentiment(text, method="nrc")
> head(nrc_vector)
[1] 1 4 2 3 2 1
> median(nrc_vector)
[1] 2
> #compare the first row of each vector using sign function
> rbind(
+   sign(head(syuzhet_vector)),
+   sign(head(bing_vector)),
+   sign(head(afinn_vector))
+ )
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    1    1    1
[2,]    1    1    1   -1    1    1
[3,]    1    1    1    1    1    1
```

Emotion Classification

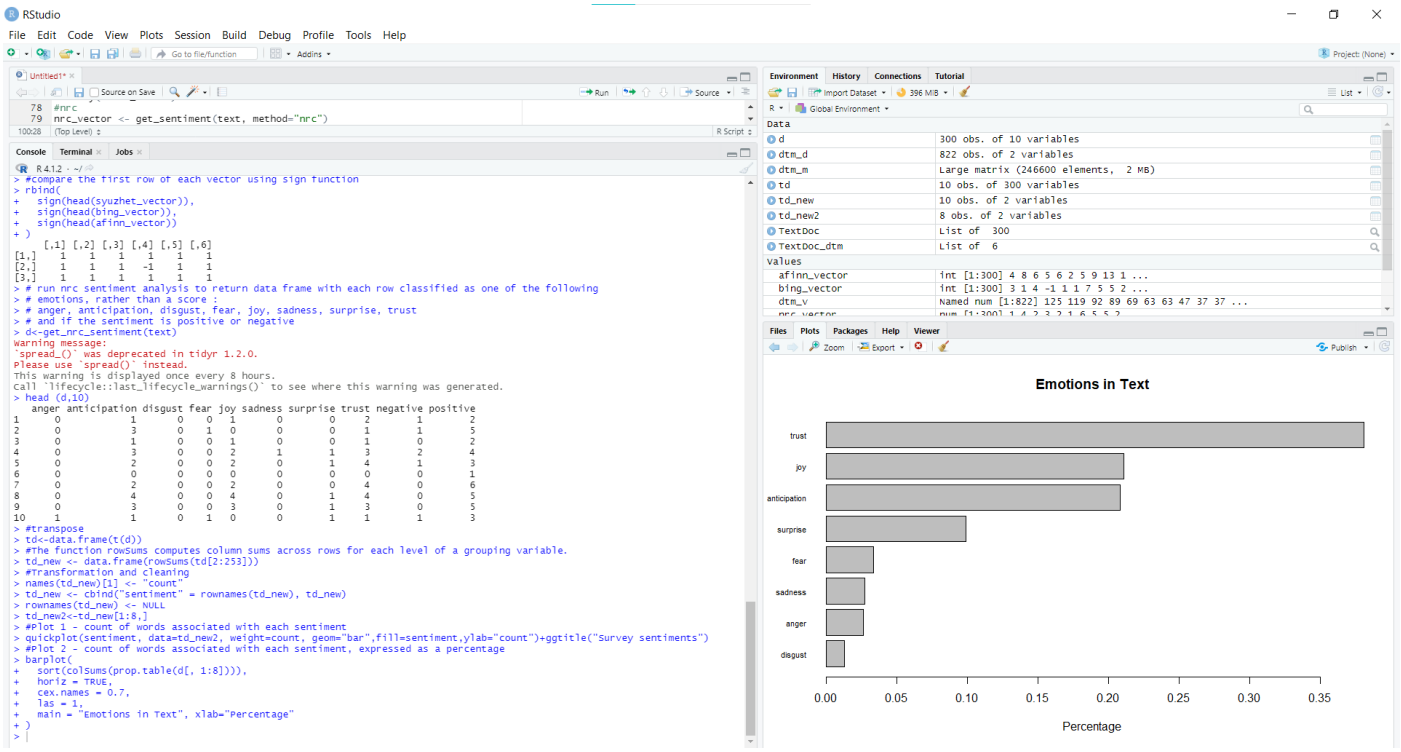
Emotion classification is built on the NRC Word-Emotion Association Lexicon (aka EmoLex). The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing

```
> # run nrc sentiment analysis to return data frame with each row classified as one of the following
> # emotions, rather than a score :
> # anger, anticipation, disgust, fear, joy, sadness, surprise, trust
> # and if the sentiment is positive or negative
> d<-get_nrc_sentiment(text)
Warning message:
`spread_()` was deprecated in tidyr 1.2.0.
Please use `spread()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> head (d,10)
   anger anticipation disgust fear joy sadness surprise trust negative positive
1      0            1       0    0   1       0        0     2        1        2
2      0            3       0    1   0       0        0     1        1        5
3      0            1       0    0   1       0        0     1        0        2
4      0            3       0    0   2       1        1     3        2        4
5      0            2       0    0   2       0        1     4        1        3
6      0            0       0    0   0       0        0     0        0        1
7      0            2       0    0   2       0        0     4        0        6
8      0            4       0    0   4       0        1     4        0        5
9      0            3       0    0   3       0        1     3        0        5
10     1            1       0    1   0       0        1     1        1        3
```

The next step is to create two plots charts to help visually analyze the emotions in this survey text. First, perform some data transformation and clean-up steps before plotting charts. The first plot shows the total number of instances of words in the text, associated with each of the eight emotions.

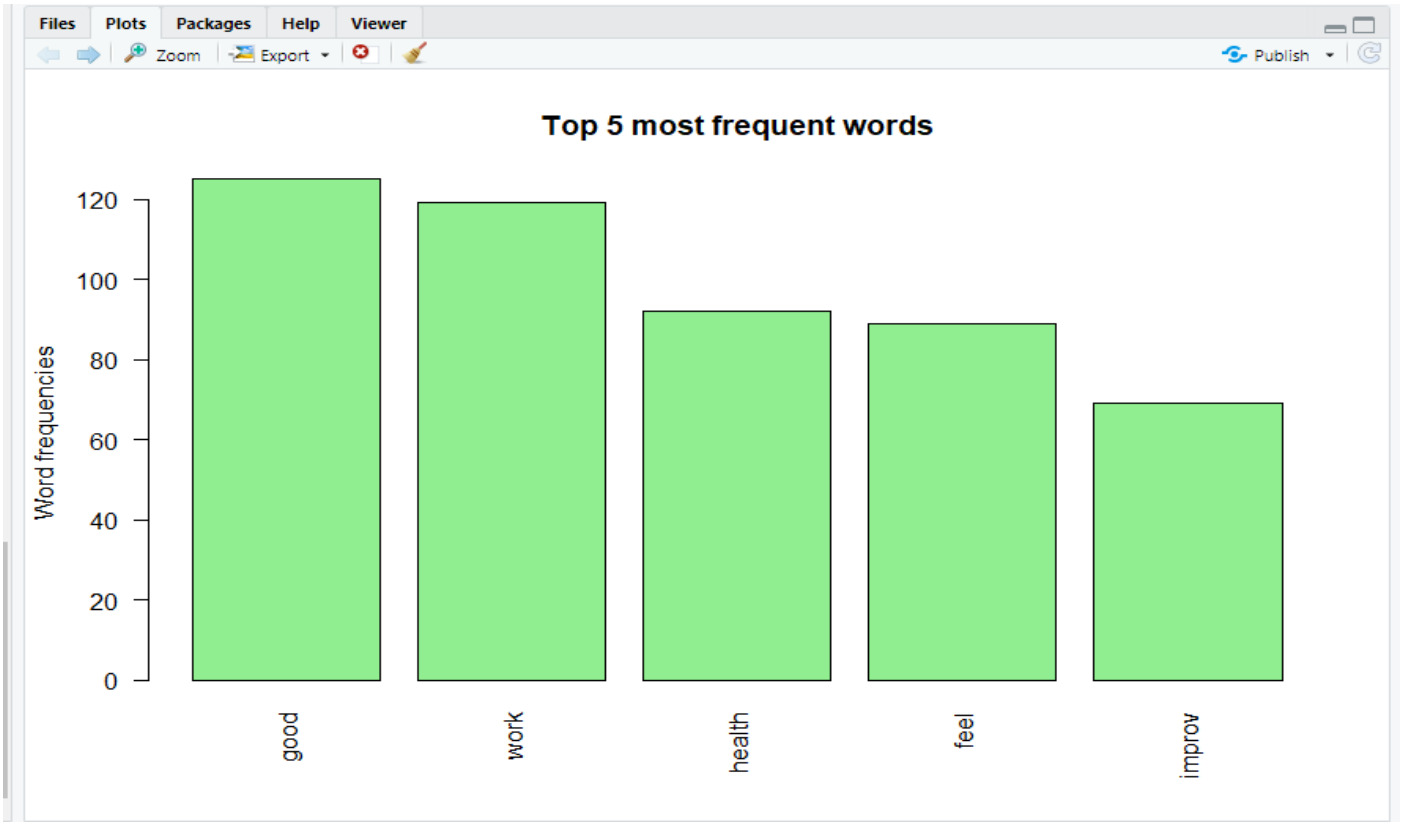


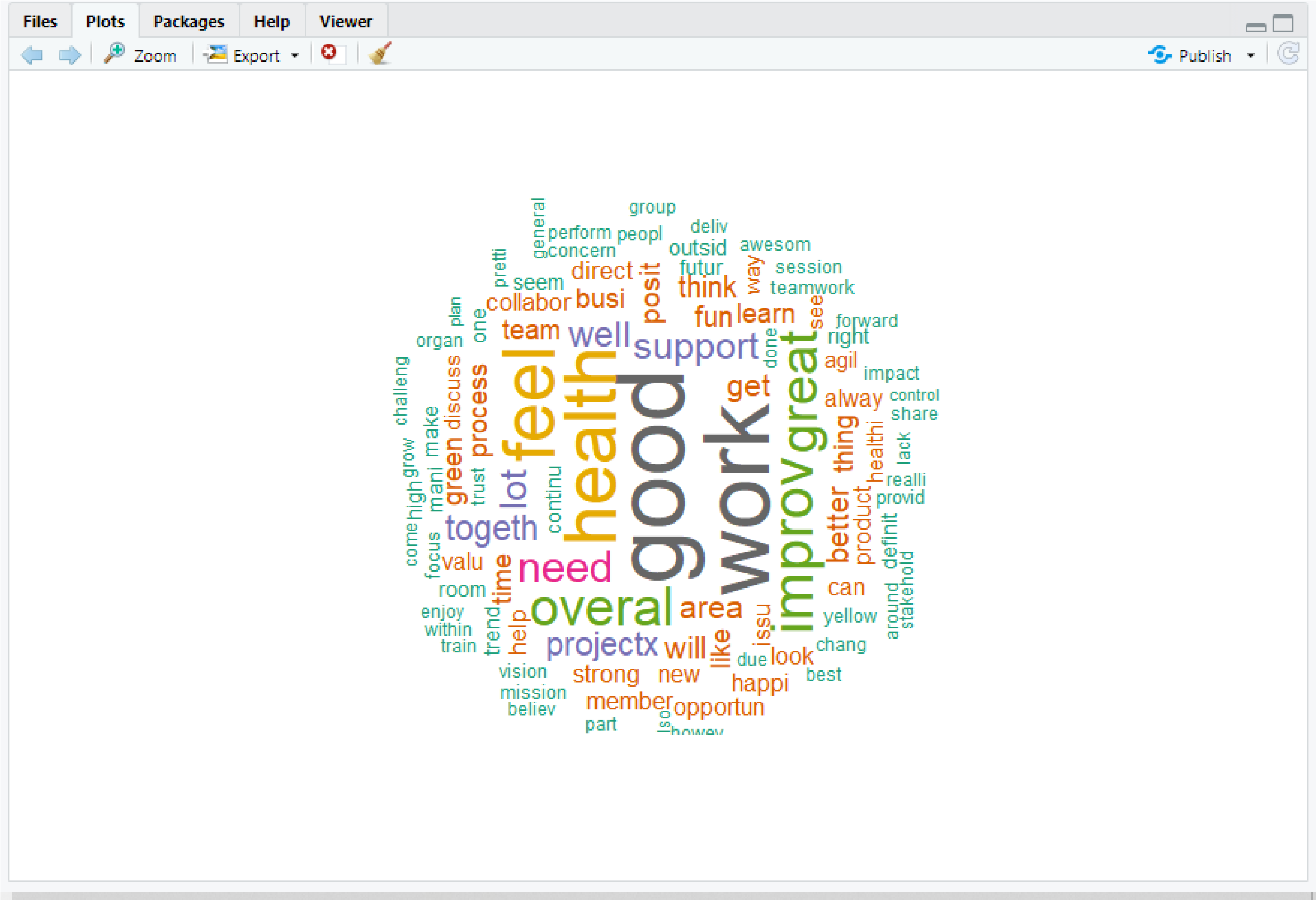
This bar chart demonstrates that words associated with the positive emotion of “trust” occurred about five hundred times in the text, whereas words associated with the negative emotion of “disgust” occurred less than 25 times. A deeper understanding of the overall emotions occurring in the survey response can be gained by comparing these number as a percentage of the total number of meaningful words.



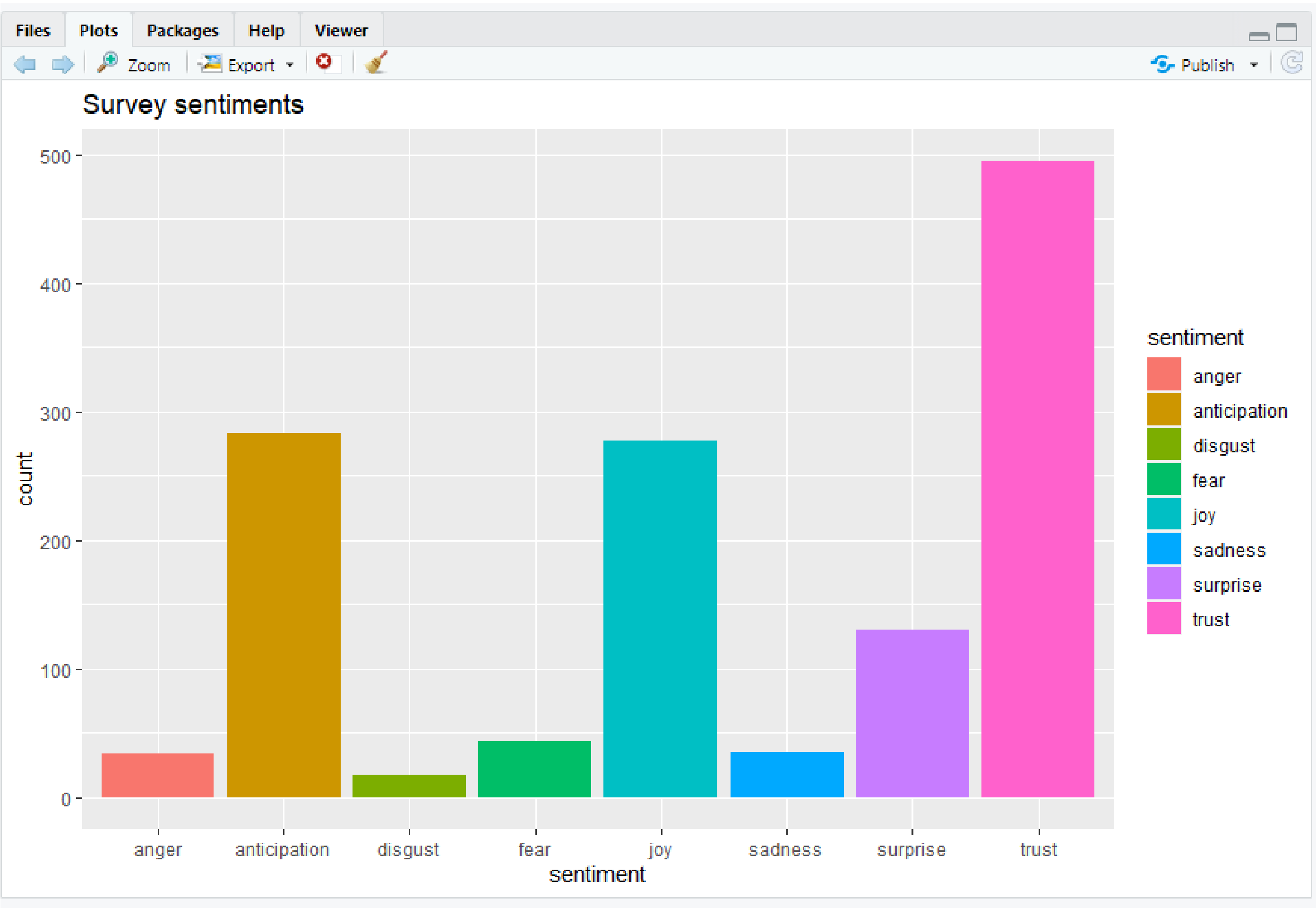
This bar plot allows for a quick and easy comparison of the proportion of words associated with each emotion in the text. The emotion "trust" has the longest bar and shows that words associated with this positive emotion constitute just over 35% of all the meaningful words in this text. On the other hand, the emotion of "disgust" has the shortest bar and shows that words associated with this negative emotion constitute less than 2% of all the meaningful words in this text. Overall, words associated with the positive emotions of "trust" and "joy" account for almost 60% of the meaningful words in the text, which can be interpreted as a good sign of team health.

PLOTS AND GRAPHS

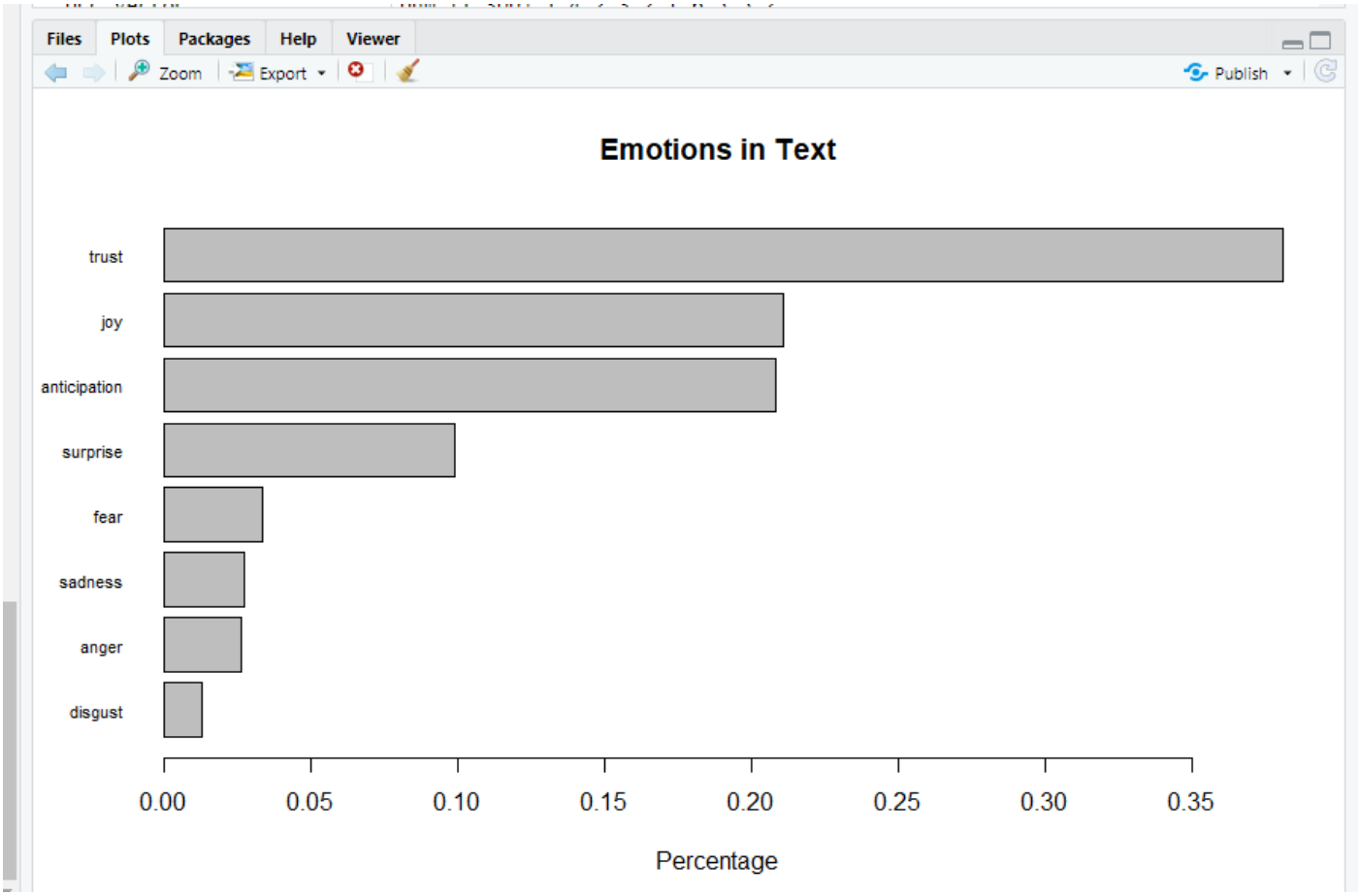




Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
good
work
health
feel
improv
great
overal
need
support
well
togeth
projectx
lot
process
team
busi
direct
think
learn
fun
get
posit
area
better
thing
alway
can
look
happi
strong
new
member
opportun
will
like
valu
time
help
green
discuss
collabor
seem
perform
peopl
group
deliv
outsid
futur
awesom
session
teamwork
forward
right
agil
impact
control
share
realli
provid
lack
healthi
product
definit
around
stakehold
yellow
chang
best
due
issu
vision
mission
believ
part
enjoy
within
train
room
focus
come
high
grow
mani
make
challeng
organ
plan
pretti
one
general
concern
trust
continu
done
see
way
trend



Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Survey sentiments
count
0
100
200
300
400
500
anger
anticipation
disgust
fear
joy
sadness
surprise
trust
sentiment
sentiment
anger
anticipation
disgust
fear
joy
sadness
surprise
trust


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Emotions in Text
trust
joy
anticipation
surprise
fear
sadness
anger
disgust
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Percentage

P.T.O