Big Data Analytics Lab

AIM: Perform text analysis using R/ Python.

Theory:

R Programming Language is used for statistical computing and is used by many data miners and statisticians for developing statistical software and data analysis. It includes machine learning algorithms, linear regression, time series, and statistical inference to name a few. R and its libraries implement a wide variety of statistical and graphical techniques, including linear and non-linear modeling, classical, statistical tests, time-series analysis, classification, clustering, and others.

Any value written inside the double quote is treated as a string in R. String is an array of characters and these collections of characters are stored inside a variable. Internally R stores every string within double quotes, even when you create them with a single quote.

Text Processing in R
Using Built-in Type in R
Using Tidyverse module
Using regex and external module
Using grep()

Text analysis (TA) is a machine learning technique used to automatically extract valuable insights from unstructured text data. Companies use text analysis tools to quickly digest online data and documents, and transform them into actionable insights.

You can us text analysis to extract specific information, like keywords, names, or company information from thousands of emails, or categorize survey responses by sentiment and topic.
When you put machines to work on organizing and analyzing your text data, the insights and benefits are huge.

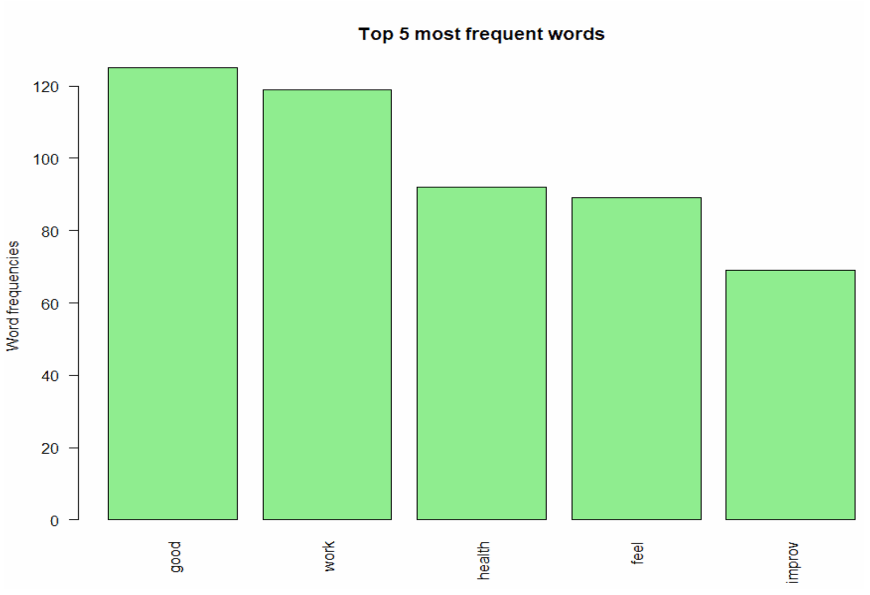
Steps involved:
Step 1: Import dataset with setting delimiter
Step 2: Text Cleaning or Preprocessing
    Remove Punctuations, Numbers
    Stemming
    Convert each word into its lower case
Step 3: Tokenization, involves splitting sentences and words from the body of the text.
Step 4: Making the bag of words and analyse the final result.

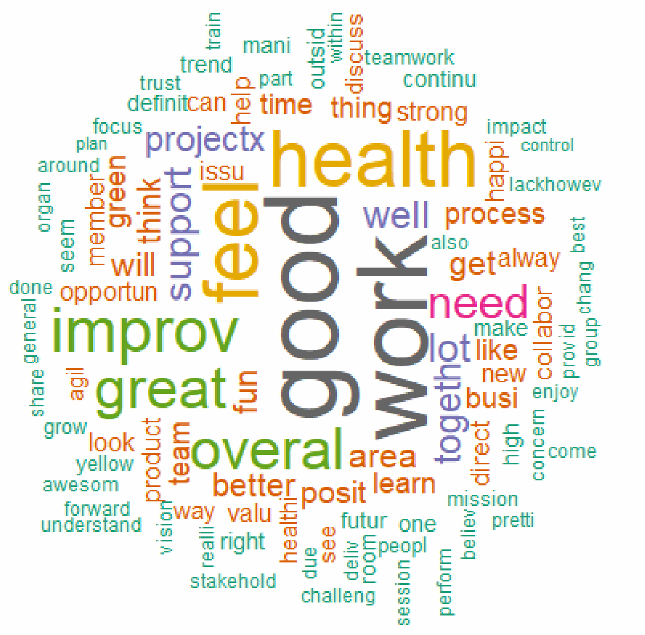
CODE:
To be executed in R tool with the data set named:
“TeamHealthRawDataForDemo”.Lessen the number of lines for quick execution.

```
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")


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 custom 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"))
```

Output:


Top 5 most frequent words
Word frequencies
good
work
health
feel
improv


good
work
health
feel
improv
great
overal
need
well
process
projectx
support
think
team
fun
lot
together
busi
like
new
area
better
posit
learn
time
thing
strong
get
alway