APPENDIX

```
# Installing the required packages for the analysis
install.packages("tidyverse")
install.packages("dplyr")
install.packages("tidytext")
# Loading libraries
library(dplyr)
library(tidyr)
library(stringr)
library(tidytext)

setwd("/Users/haris/Downloads/Supporting materials for the coursework assignment-20221215")
# setting the working directory to the folder containing the data
# please change this according to your folder location.

modern_words <- read.delim("modern_word_count.txt") # loading the txt file
nrow(modern_words)
str(modern_words) # checking the structure of the data
modern_words%>%arrange(word_count)%>%head(50) # checking first 50 values after arranging
modern_words%>%arrange(desc(word_count))%>%head(50) # checking last 50 values after arranging
# this data is provided to us for the sentiment analysis but this data has
# very limited use here as this contains modern english words taken from the web and
# the play is written in antient english.

King_Lear <- read.csv("King_Lear_words_and_players_only.csv") #loading the given csv file
str(King_Lear)
King_Lear%>%head(30)

data(stop_words)  # loading the stop words data
stop_words
nrow(stop_words) # checking the number of rows of this data as to confirm weather this can be used for
# removing unncessory words to clean the king_lear data

King_Lear_tidy <- King_Lear %>%
  unnest_tokens(word, text)%>%
  anti_join(stop_words)
print(head(King_Lear_tidy))  #checking the head
King_Lear_tidy %>%count(word, sort = TRUE)%>%head(50)  # count most common words

# we have to create a custom stop words data to be used to remove the
# common words used in antient english for cleaning the data
custom_stop_word <- bind_rows(tibble(word = c("thou","thy","thee","tis"),
                                     lexicon = c("custom")),
                              stop_words)

head(custom_stop_word)
# removal of custom stop words in King_Lear
King_Lear_tidy <- King_Lear_tidy %>%
  anti_join(custom_stop_word)
King_Lear_tidy

# we forst use "nrc" library for the sentiment analysis
nrc_joy <- get_sentiments("nrc")  # checking the data in "nrc"
nrc %>% count(sentiment)

king_lear_joy_words <- King_Lear_tidy %>%inner_join(nrc_joy)%>%
  count(word, sort = TRUE)   # segregatig joy words
```

```
king_lear_joy_words
print(plot(king_lear_joy_words$n)) # plotting graph for the joy words
king_lear_joy_words%>%arrange(desc(n))%>%head(10)

data1 <- King_Lear_tidy %>% inner_join(nrc) # joining the data with "nrc"
count(data1)/20     # calculating the number of words in each part of 20 parts
data1
data1_mutated <- mutate(data1,sno=row_number()) %>% count( index = sno %/% 407, sentiment)
# we divided the data into 20 eaual parts as to observe and compare the sentiments
# in each part because if we use single values, this cannot be analysed.

data2_mutated <- data1_mutated %>% pivot_wider(names_from = sentiment, values_from = n, values_fill = 0)
# converting the data into a wider format for analysis
data1_mutated # checking the data
data2_mutated
king_lear_emotion <- data1 %>%filter(player=="LEAR") %>% mutate(sno=row_number()) %>%
  count( index = sno %/% 100, sentiment) # we can filter the words by king lear and analyse this
# to plot a graph which can show the changing emotions of king lear throughout the play
# here we have divided king lear's words into multiple parts to analyse each part
king_lear_emotion
library(ggplot2)

# we can observe the data of each emotion here as a graph
ggplot(data2_mutated, aes(index, anger)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, anticipation)) +
  geom_col()
ggplot(data2_mutated, aes(index, disgust)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, fear)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, joy)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, sadness)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, surprise)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, trust)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, positive)) +
  geom_col(show.legend = FALSE)
ggplot(data2_mutated, aes(index, negative)) +
  geom_col(show.legend = FALSE)
ggplot(data = data1_mutated,aes(index,n,color = sentiment)) +
  geom_line() # combined line graph of each emotion throughout the play
ggplot(data1_mutated,aes(index,n,color = sentiment))+
  geom_line()+
  facet_wrap(facets = vars(sentiment)) # graphs of each emotions as line graph

ggplot(king_lear_emotion,aes(index,n,color = sentiment))+
  geom_col()+
  facet_wrap(facets = vars(sentiment)) # king lear's emotions throughout the play
# now we use afinn lexicon for the sentiment analysis

King_Lear_words <- King_Lear_tidy %>%count(word, sort = TRUE)%>%
  inner_join(get_sentiments("afinn"),"word")%>%
  mutate(weighted=n*value)  # joining the data from "afinn"
King_Lear_words$value%>%table()%>%barplot()  # plotting a bar graph for "afinn" analysis
King_Lear_words$weighted%>%sum()  # calculating the total score of the play by adding the weighted scores
```

```
# using bing library

bing <- get_sentiments("bing")
head(bing)
table(bing$sentiment)

King_Lear_tidy %>%
  inner_join(get_sentiments("bing"),"word") %>%count(player,sentiment)
# we can check the number of positive and negative words said by each character
king_lear_bing <- King_Lear_tidy %>% inner_join(bing)# joining "bing" data for analysis
(king_lear_bing%>%count())/100  # calculating number of words in each part of total 100 parts
king_lear_bing <- mutate(king_lear_bing,sno=row_number()) %>% count( index = sno %/%21 , sentiment)
king_lear_bing%>%head(20)
king_tidy_bing_wide <- king_lear_bing  %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative) # converting to the wide format for separating each sentoment
king_tidy_bing_wide%>%head(20)

ggplot(king_tidy_bing_wide, aes(index, sentiment)) +
  geom_col(show.legend = FALSE)  # generating graph for each part
```

R PROGRAMMING ASSIGNMENT

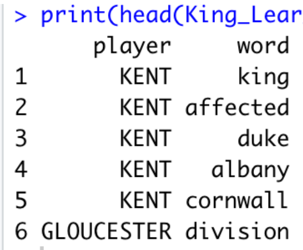
In this assignment, Text analysis of the play “king Lear” by Shakespeare is done by three sentiment analysis Libraries in the R programming Language.

All the R coding text is given in the Appendix.

We were given a CSV file containing the data with two columns which include statements spoken by each character and the name of that character. Analysis of this data is performed in this report.

We should start with Installing the required R packages for the analysis of this Text.
Firstly, we load the data from the file into a data frame, this data frame (named “king_lear”) is by default in a non-tidy format, so we convert this into a tidy format using the “unnest_tokens” function and along with that we remove the stop words from that data by “anti_join” function using the “stop_words” package which contains the data of most common words which do not contribute to the sentiment analysis.
We name this new data frame “King_Lear_tidy”

Then we check the head(first six values) of the data frame which is as follows:

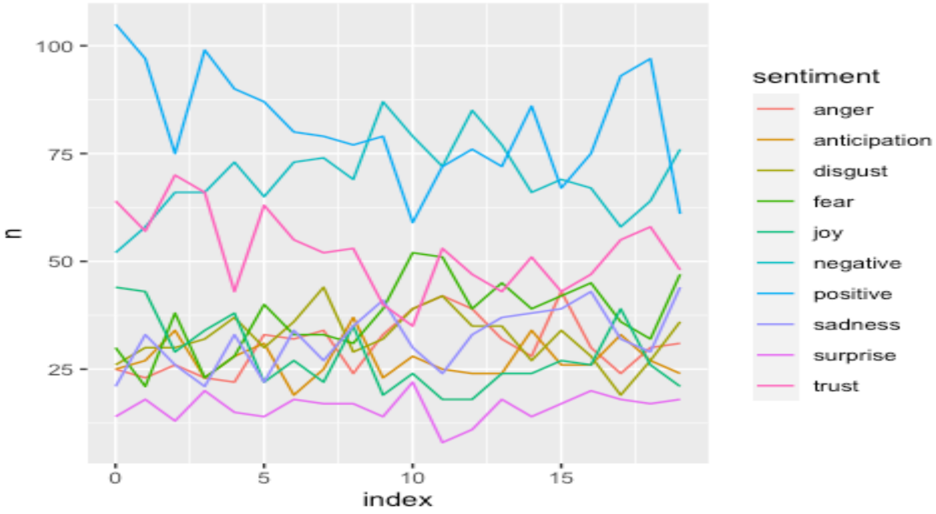


```
> print(head(King_Lear_tidy))
      player     word
1       KENT     king
2       KENT affected
3       KENT     duke
4       KENT   albany
5       KENT cornwall
6 GLOUCESTER division
```

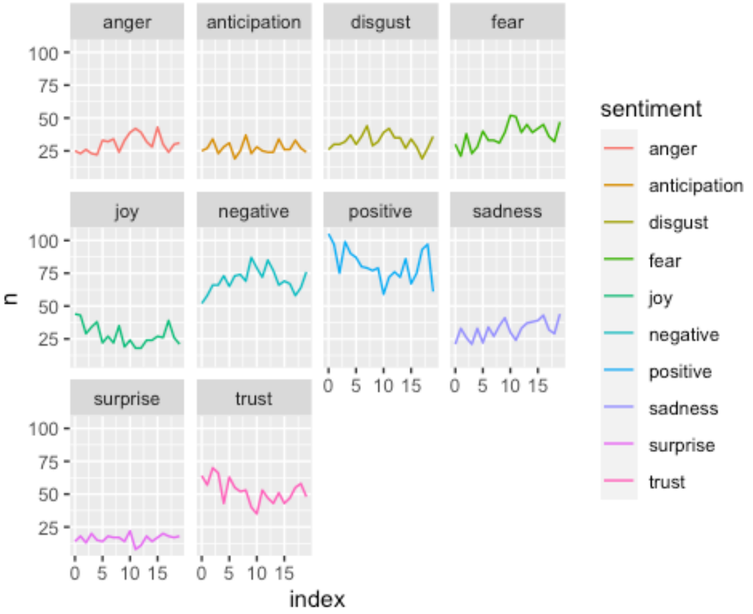
After analyzing the most common words in the resulting data frame, there was a need to remove some more common words which were used as analogous to the modern common words. These words are: - "thou","thy","thee" and "tis".
They were removed from the data frame by creating a custom stop words data frame and then removing this from the “King_Lear_tidy” data frame by using anti_join.
We finally got the cleaned data frame which is ready to be analyzed.
In this analysis, there are three lexicons that can be used for the sentiment analysis: -

- afinn from Finn Arup Nielsen,
- bing from Bing Liu and collaborators, and
- nrc from Saif Mohammad and Peter Turney

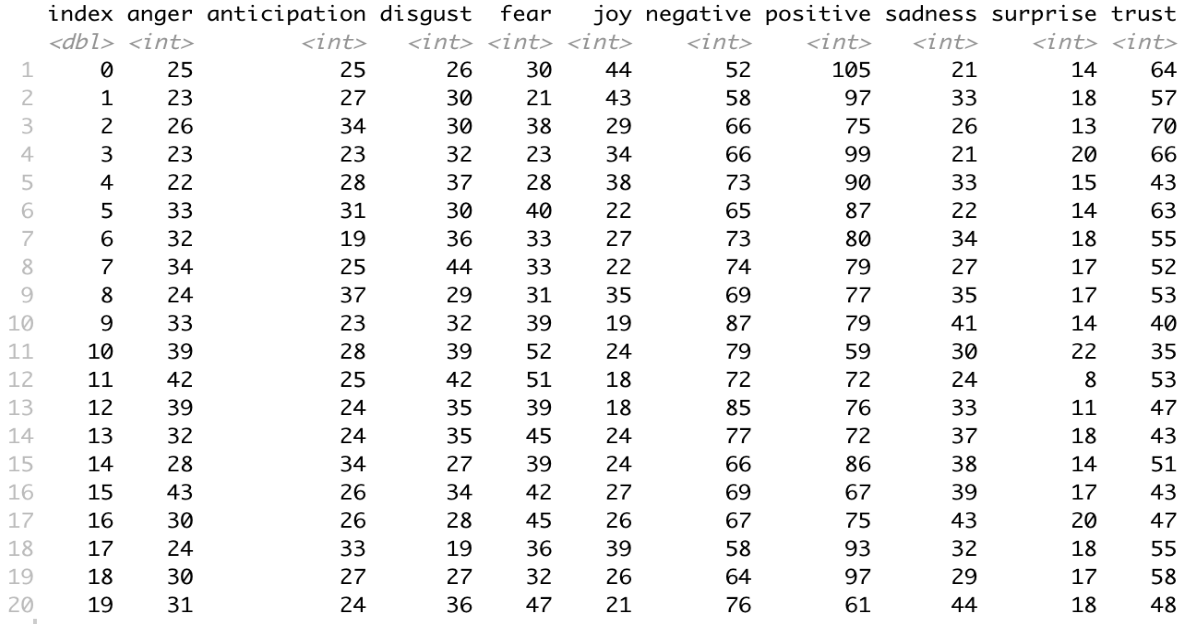
We first use “nrc” to analyze the sentiments across the given play. From the sentiment data frame of the “nrc”, we join the columns of this to our tidy data frame “king_lear_tidy”.
This data is observed and concluded that a single word categorized into different emotions can’t give us insights about the whole play, hence we divide the words into groups of 20 equal parts by diving rows by 407 and creating a variable “Index”.
We can then plot the line graph of the data frame using “ggplot” function.
This data frame can also be divided into pivot wider format to present a table showing different emotions in different parts of the play and from that data bar graph can be plotted.



Reg no. 2205006

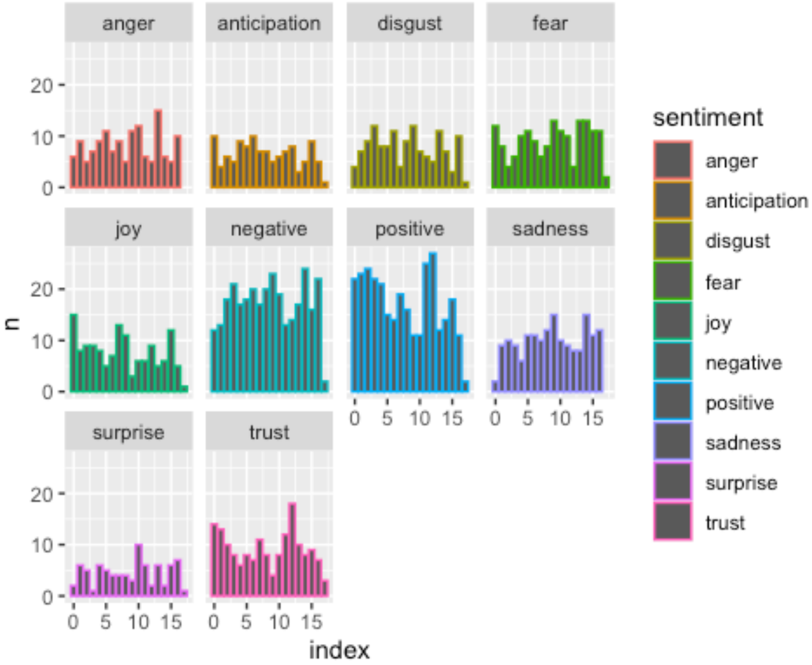


This graph shows how different emotions are changing throughout the play.
The wider format table is created by changing the data frame.
This table shows us the frequency of the emotional words in each interval of 407 words: -

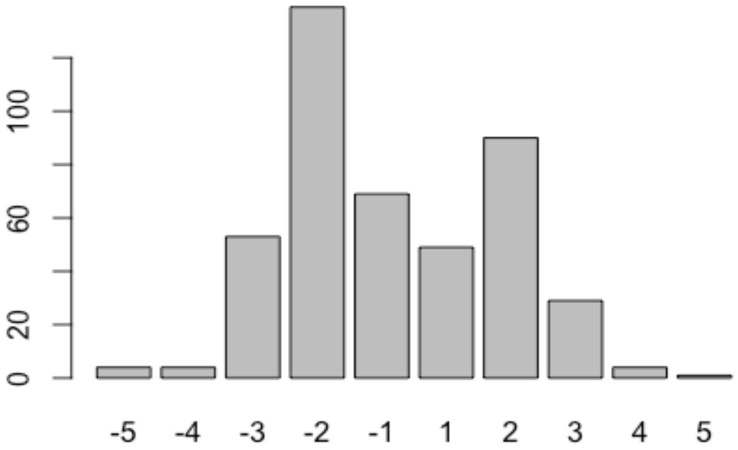


```
   index anger anticipation disgust  fear   joy negative positive sadness surprise trust
   <dbl> <int>        <int>   <int> <int> <int>    <int>    <int>   <int>    <int> <int>
 1     0    25           25      26    30    44       52      105      21       14    64
 2     1    23           27      30    21    43       58       97      33       18    57
 3     2    26           34      30    38    29       66       75      26       13    70
 4     3    23           23      32    23    34       66       99      21       20    66
 5     4    22           28      37    28    38       73       90      33       15    43
 6     5    33           31      30    40    22       65       87      22       14    63
 7     6    32           19      36    33    27       73       80      34       18    55
 8     7    34           25      44    33    22       74       79      27       17    52
 9     8    24           37      29    31    35       69       77      35       17    53
10     9    33           23      32    39    19       87       79      41       14    40
11    10    39           28      39    52    24       79       59      30       22    35
12    11    42           25      42    51    18       72       72      24        8    53
13    12    39           24      35    39    18       85       76      33       11    47
14    13    32           24      35    45    24       77       72      37       18    43
15    14    28           34      27    39    24       66       86      38       14    51
16    15    43           26      34    42    27       69       67      39       17    43
17    16    30           26      28    45    26       67       75      43       20    47
18    17    24           33      19    36    39       58       93      32       18    55
19    18    30           27      27    32    26       64       97      29       17    58
20    19    31           24      36    47    21       76       61      44       18    48
```

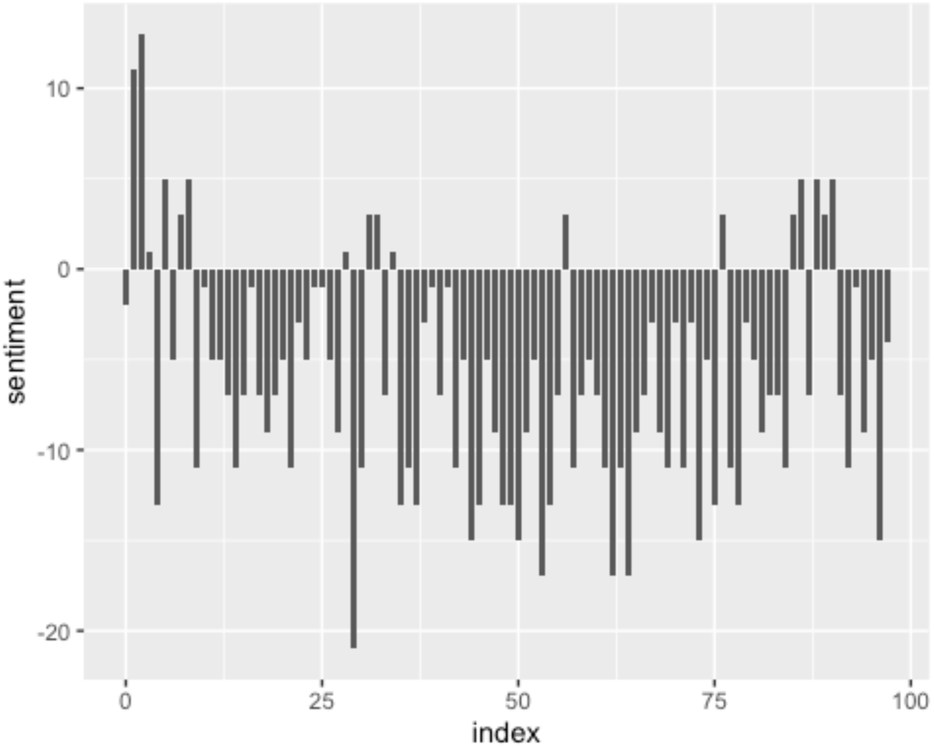
We can also observe only “king Lear’s” emotions throughout the play by filtering out the character and then plotting the column graph:



We can also use “afinn” library to assign the negative and positive scores to the words in the king Lear play. By joining data from this library to king Lear, we can plot a bar graph between the occurrence of negative words or positive words with their frequency across the entire play. This gives us an idea of the impact of negative and positive sentiments in the play.



We can calculate the weightage of each word by multiplying frequency with “afinn” score.
By adding all the values of the weighted column, we can calculate the total score of the play which is: “-536”. This shows that the overall sentiment of the play is negative.
After this, we use the “bing” library for the sentimental analysis. We can categorize each of the words into positive and negative sentiments with the help of inner_join function. Then we divide the data into 100 equal parts and add their units (+1 for positive and -1 for negative) to get the total sentiment score of each part. After obtaining this, the graph is plotted between sentiment and Index (each part).



Conclusion: -
In this report, we can conclude that the overall sentiment of this play is negative and from the graph by using “nrc”, we can observe how the different emotions are changing throughout the play and “king Lear’s emotions for example “trust”, which is quite high at the start of the play but decreased at the end of the play. From the graph by using “afinn” library, we can analyze that words which are negative to the value of -2 are highest in number throughout the play and overall play is certainly negative which is calculated from the weighted score. And in the last we use “bing” library, in this, we can see the parts and the intensity of positive and negative emotions throughout the play.