Assignment 2: Text Analytics for National Institutes of Health (100 points)

Student Name: Abdul Rahman Gulam

Fall 2019

Purpose: To perform text analytics including creating word clouds, perform sentiment analysis and topic modeling

Description: The data for this assignment has been collected from psychcentral.com. This website offers an online forum for posting questions and answers related to mental health. Please visit https://forums.psychcentral.com for more information. Our objective is to perform text analytics to discover useful information related to mental health.

Instructions: Please follow these steps:

1. In Canvas, navigate to Assignments and then Assignment2
2. Download and save the data set psychcentral_data.csv
3. Read the file: data <- fread("psychcentral_data.csv", sep=",", header=T, strip.white = T, na.strings = c("NA","NaN","","?"))
   - **3.1. (1 point)** What are the column names in the data?

     There are four columns in the data : row, q_subject, q_content,answers.
   - 3.2. **(1 point)** How many rows does this data have?

     There are 8360 rows.

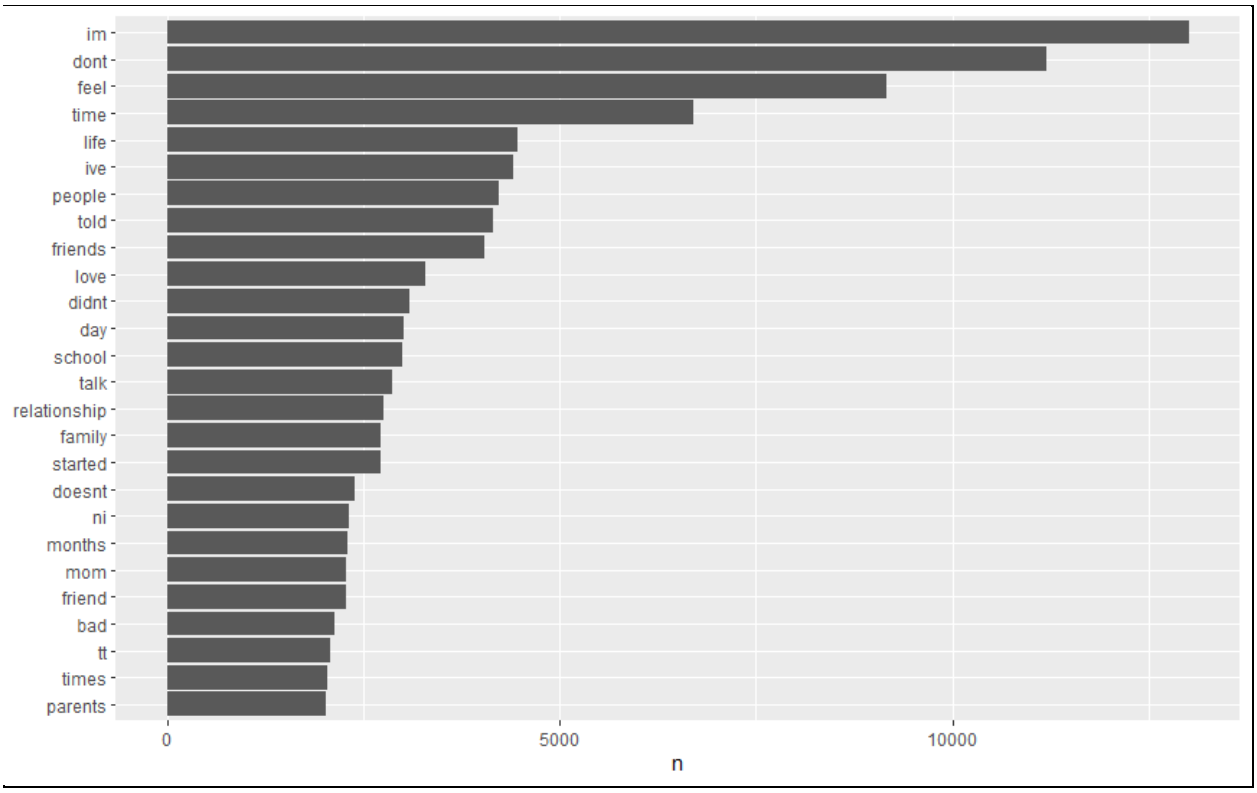
4. Use libraries “dplyr” and “tidytext” to tokenize column q_content. Then remove the stop-words. The count the number of tokens.
   - 4.1. **(2 points)** What are the top five tokens returned?

     [1] Im 13012
     [2] Don’t 11197
     [3] Feel 9168
     [4] Time 6697
     [5] Life 4464

A tibble: 46,081 x 2

	word	n
	<chr>	<int>
1	im	13012
2	dont	11197
3	feel	9168
4	time	6697
5	life	4464
6	ive	4403
7	people	4233
8	told	4150
9	friends	4045
10	love	3281

4.2. **(2 points)** Use library "ggplot2" to create a visualization that shows the frequency of the tokens that appeared for at least 2000 times. (Hint: Change n in argument filter to 2000). Paste the visualization below:



4.3. **(2 points)** Based on the results in 4.2., would you suggest stemming on this text? Why? Bring one example from the visualization above that shows stemming should be done on this text?

Yes. I would suggest stemming on this text. Stemming helps us to prevent the similar words getting repeated. For instance, in the above visualization we can see two words - - "friends" and "friend". To avoid words being counted twice stemming is required.

4.4. Install "SnowballC" package using install.packages("SnowballC", repos = "https://cran.r-project.org"). Use library "SnowballC" to stem q_content using the code below:

```
library(SnowballC)
tidy_text <- data %>%
    unnest_tokens(word, q_content) %>%
    mutate(word = wordStem(word))
```

4.4.1. **(2 points)** Then remove the stop-words. Now what are the top five tokens after stemming?

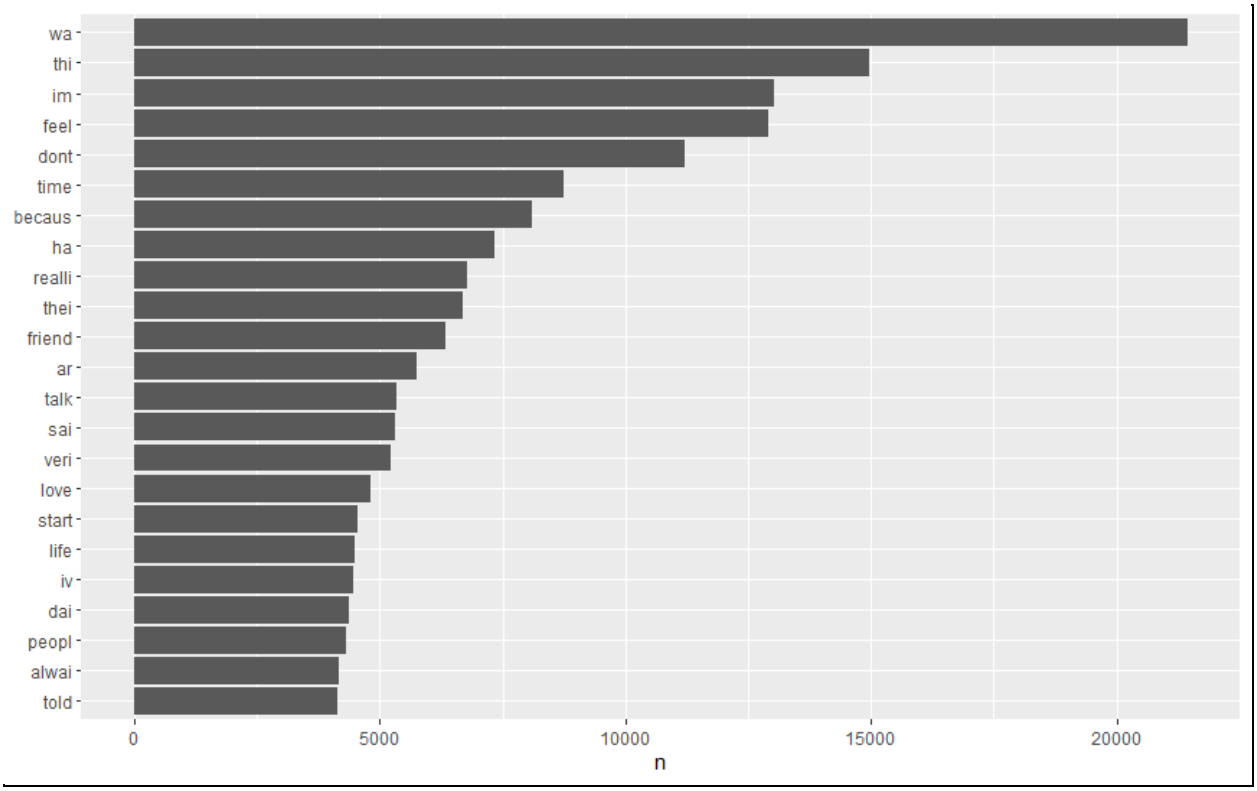
[1] Wa : 21437

[2] Thi: 14961

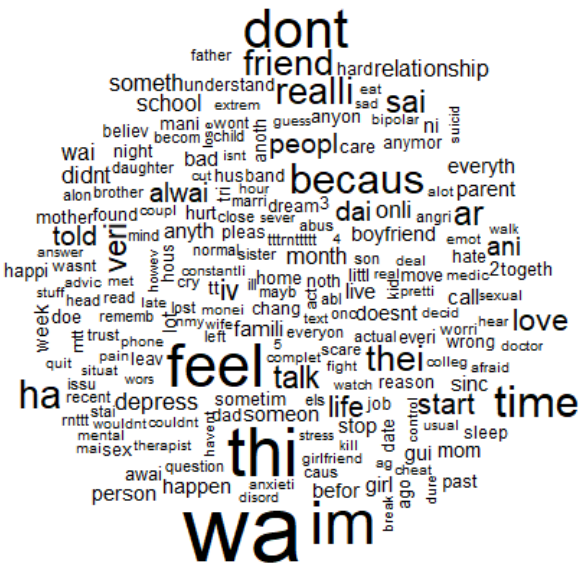
[3] Im: 13016
[4] Feel:12905
[5] Don't: 11197

```
> tidy_text %>%
+ count(word, sort=TRUE)
# A tibble: 36,404 x 2
   word       n
   <chr>  <int>
 1 wa     21437
 2 thi    14961
 3 im     13016
 4 feel   12905
 5 dont   11197
 6 time    8755
 7 becaus  8104
 8 ha      7340
 9 realli  6780
10 thei    6698
# ... with 36,394 more rows
> library(dplyr)
```

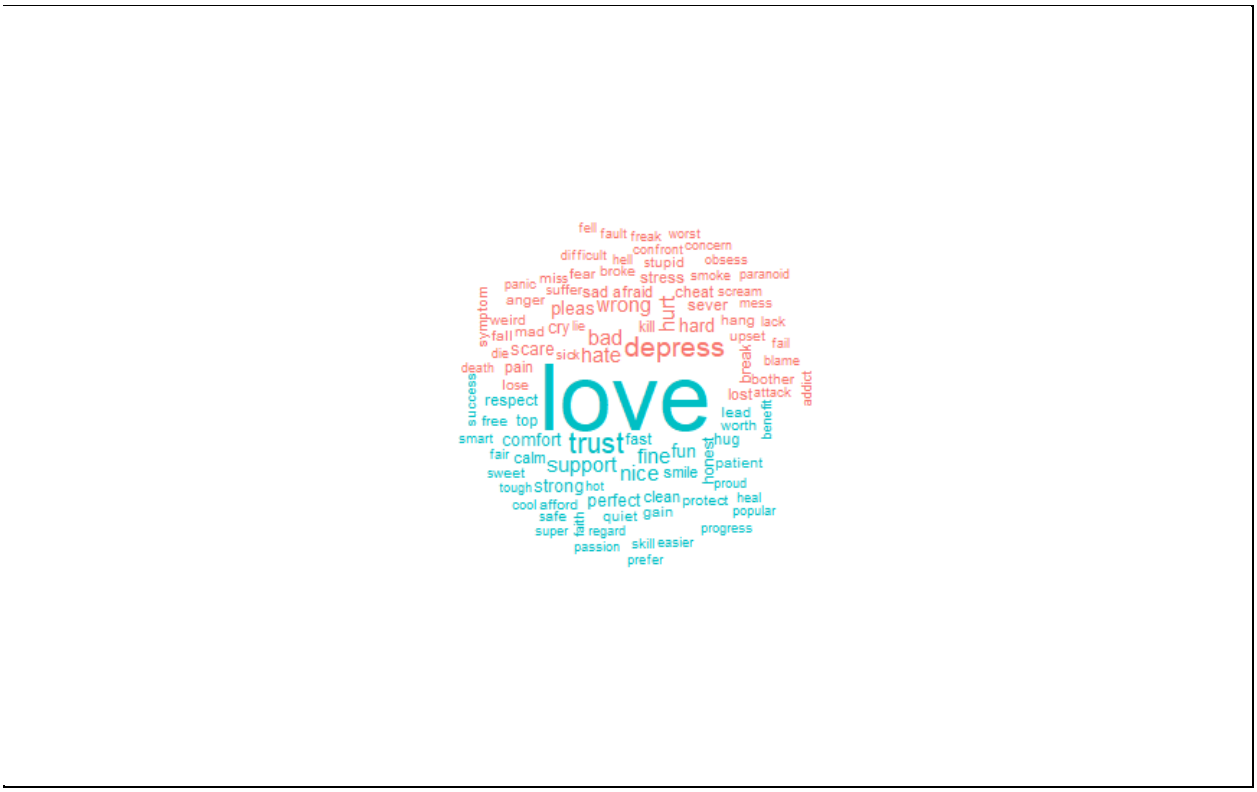
4.4.2. **(2 points)** Use library "ggplot2" to create a visualization that shows the frequency of the tokens that appeared for at least 4000 times. (Hint: Change n in argument filter to 4000). Paste the visualization below:



4.4.3. **(3 points)** Use library "wordcloud" to create a word cloud with the 200 most used tokens. Paste the visualization below:



4.4.4. **(5 points)** Create a color-coded word cloud based on sentiment. Use the most frequent 100 tokens for positive and negative words. Paste the word cloud in the space below:



5. Repeat all the steps in question 4 but this time for column answers.

5.1. **(2 points)** What are the top five tokens returned?

[1] Don't: 18010

[2] Feel:13279

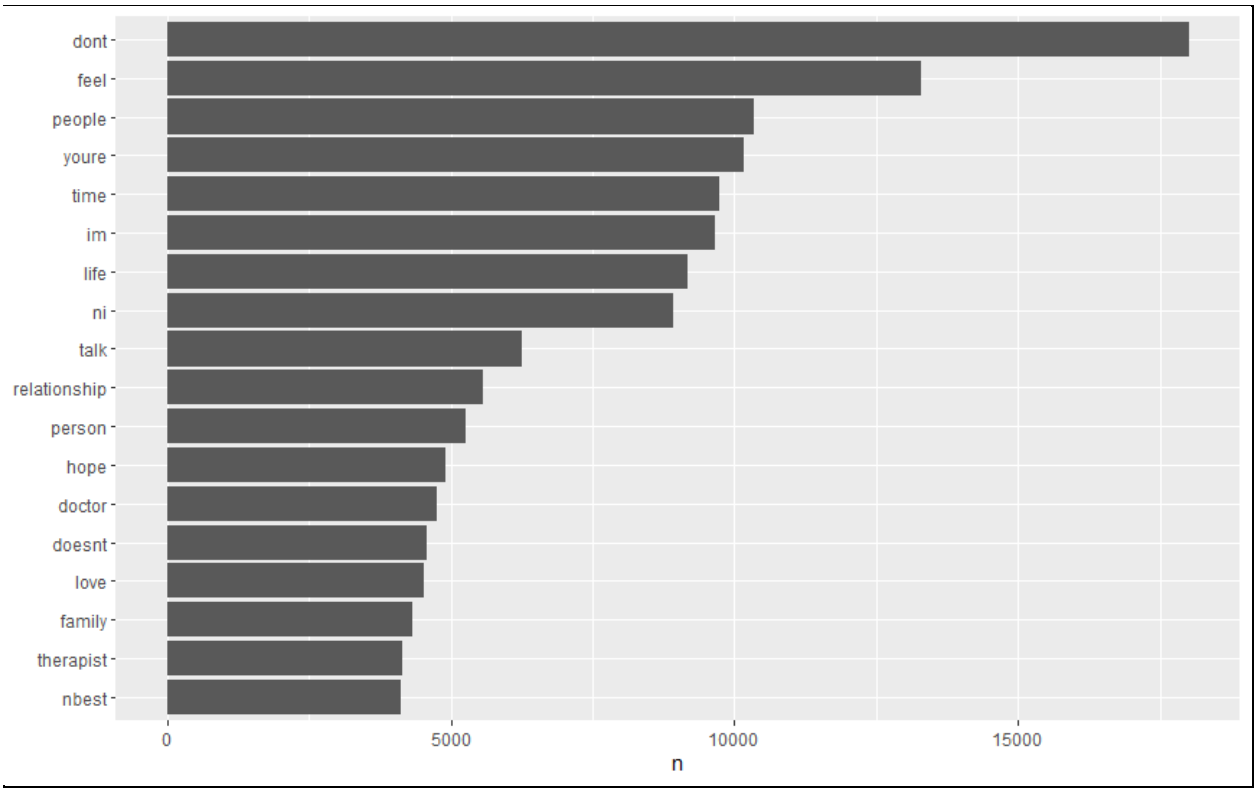
[3] People:10334

[4] Youre:10162

[5] Time:9729

```
> tidy_text %>%
+ count(word, sort = TRUE)
# A tibble: 54,645 x 2
   word              n
   <chr>         <int>
 1 dont          18010
 2 feel          13279
 3 people        10334
 4 youre         10162
 5 time           9729
 6 im             9664
 7 life           9169
 8 ni             8913
 9 talk           6245
```

5.2. **(2 points)** Use library "ggplot2" to create a visualization that shows the frequency of the tokens that appeared for at least 4000 times. (Hint: Change n in argument filter to 4000). Paste the visualization below:



5.3. Install “SnowballC” package using install.packages("SnowballC", repos = "https://cran.r-project.org"). Use library “SnowballC” to stem answers using the code below:

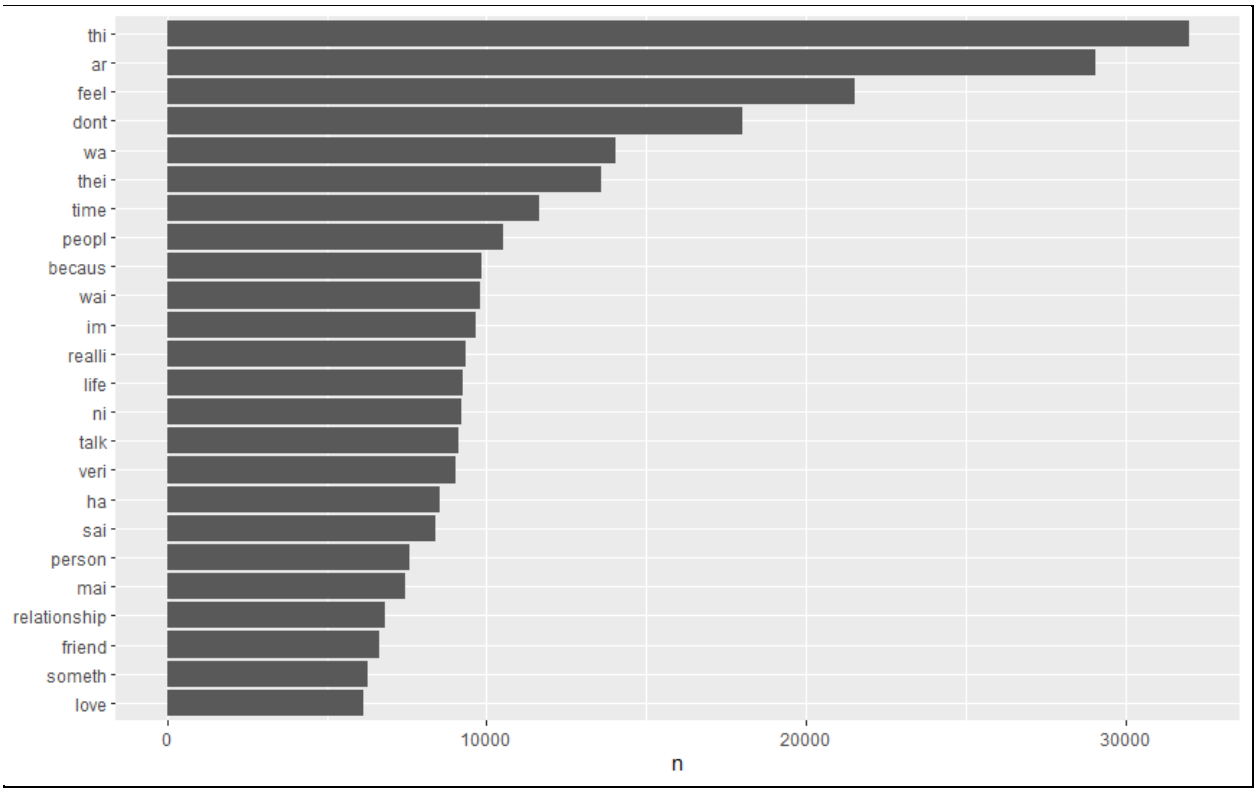
```
library(SnowballC)
tidy_text <- data %>%
    unnest_tokens(word, answers) %>%
    mutate(word = wordStem(word))
```

5.3.1. **(2 points)** Then remove the stop-words. Now what are the top five tokens after stemming?

[1] Thi:31989
[2] Ar:29063
[3] Feel:21550
[4] Don’t:18011
[5] Wa:14041

```
tidy_text %>%
count(word, sort = TRUE)
A tibble: 42,210 x 2
   word       n
   <chr>  <int>
 1 thi    31989
 2 ar     29063
 3 feel   21550
 4 dont   18011
 5 wa     14041
 6 thei   13587
 7 time   11661
 8 peopl  10537
 9 becaus  9834
10 wai     9797
```

5.3.2. **(2 points)** Use library "ggplot2" to create a visualization that shows the frequency of the tokens that appeared for at least 6000 times. (Hint: Change n in argument filter to 6000). Paste the visualization below:



5.3.3. **(6 points)** Use library "wordcloud" to create a word cloud with the 200 most used tokens. Paste the visualization below:

5.3.4. **(6 points)** Create a color-coded word cloud based on sentiment. Use the most frequent 100 tokens for positive and negative words. Paste the word cloud in the space below:

6. Use the following code to perform topic-modeling on q_content:

```
library(RTextTools)
library(tm)
library(wordcloud)
library(topicmodels)
library(slam)
data <- data[1:1000,] # We perform LDA on the rows 1 through 1000 in the data.
corpus <- Corpus(VectorSource(data$q_content), readerControl=list(language="en"))
dtm <- DocumentTermMatrix(corpus, control = list(stopwords = TRUE, minWordLength = 2,
removeNumbers = TRUE, removePunctuation = TRUE,  stemDocument = TRUE))
rowTotals <- apply(dtm , 1, sum) #Find the sum of words in each Document
dtm.new   <- dtm[rowTotals> 0, ] #remove all docs without words
lda <- LDA(dtm.new, k = 5) # k is the number of topics to be found.
```

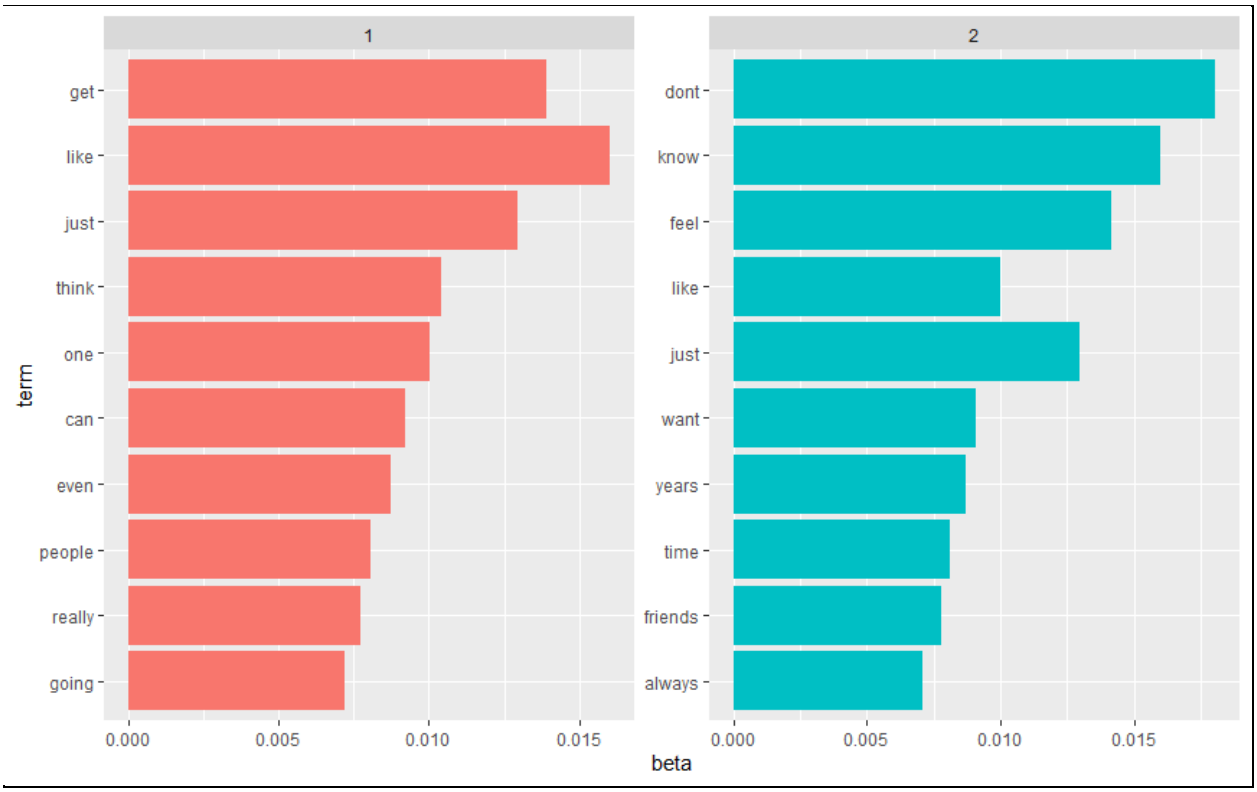
6.1. **(5 points)** The code above will create the beta scores for each document per topic (k = 5). Then create bar plots (similar to what we created in class) for each topic for 10 tokens (top_n(10, beta)). Paste the visualization below.

6.2. **(5 points)** Based on the visualization in 6.1., what can you say about k? Would you try a larger k or a smaller k?

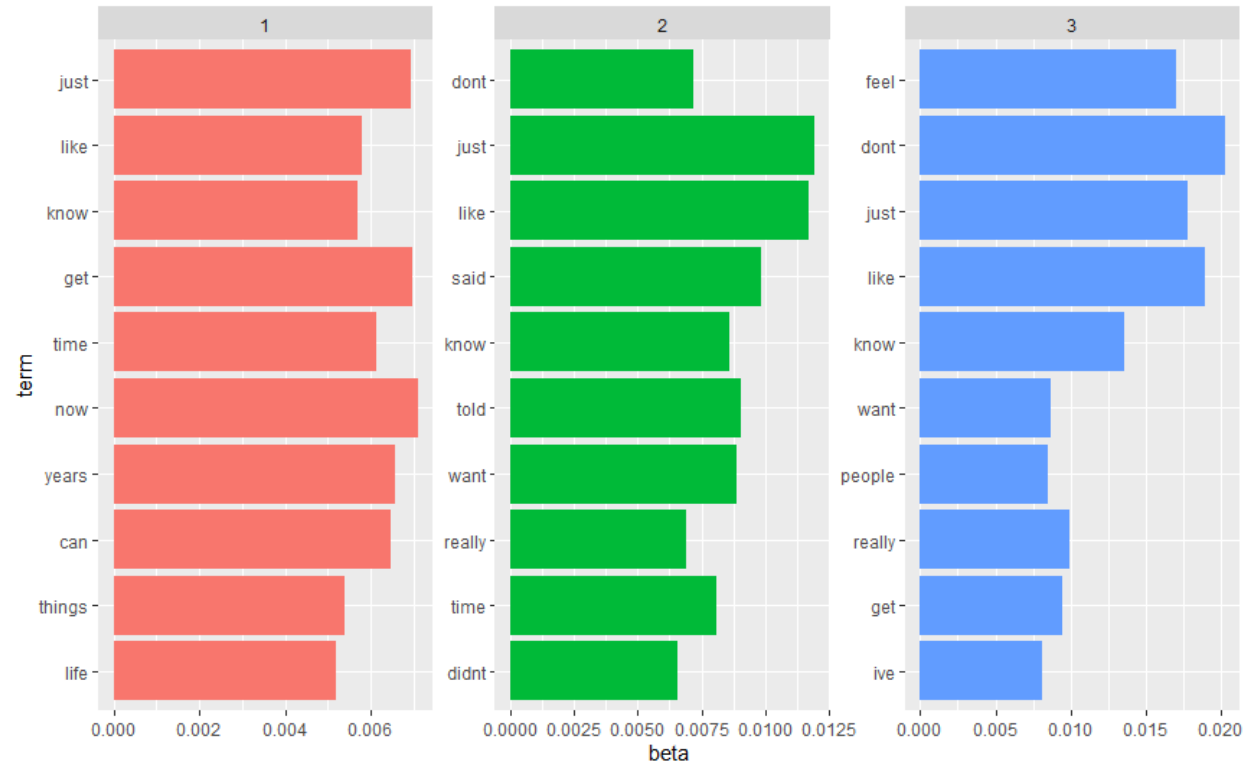
From the visualization we can infer that the words like don't, just, like, feel, know are common and thus we cannot figure out much about the topic and need more information to find out the topic. Thus, we need to try out for larger value for K.

6.3. **(10 points)** Repeat 6.1. with the following ks:

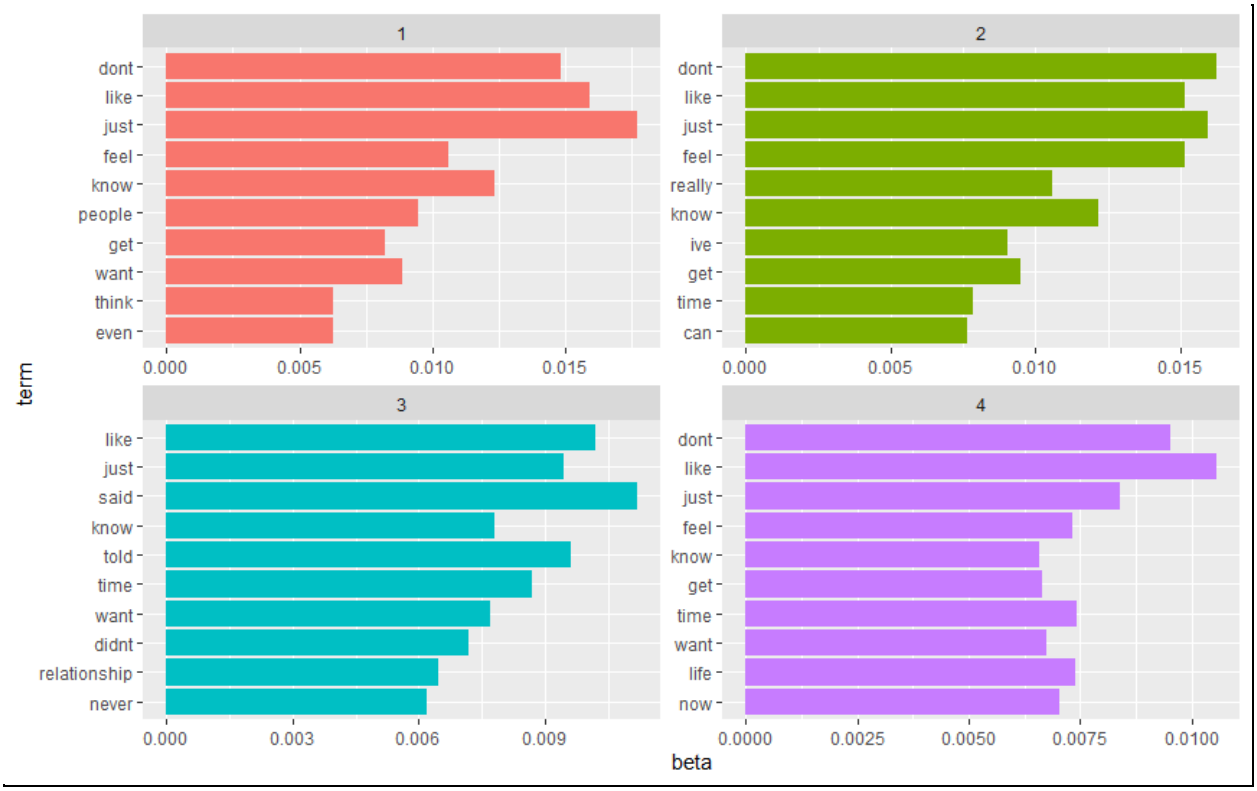
6.3.1. K = 2. Paste your visualization in the space below:



6.3.2. K = 3. Paste your visualization in the space below:



6.3.3. K = 4. Paste your visualization in the space below:



6.3.4. K = 10. Paste your visualization in the space below:

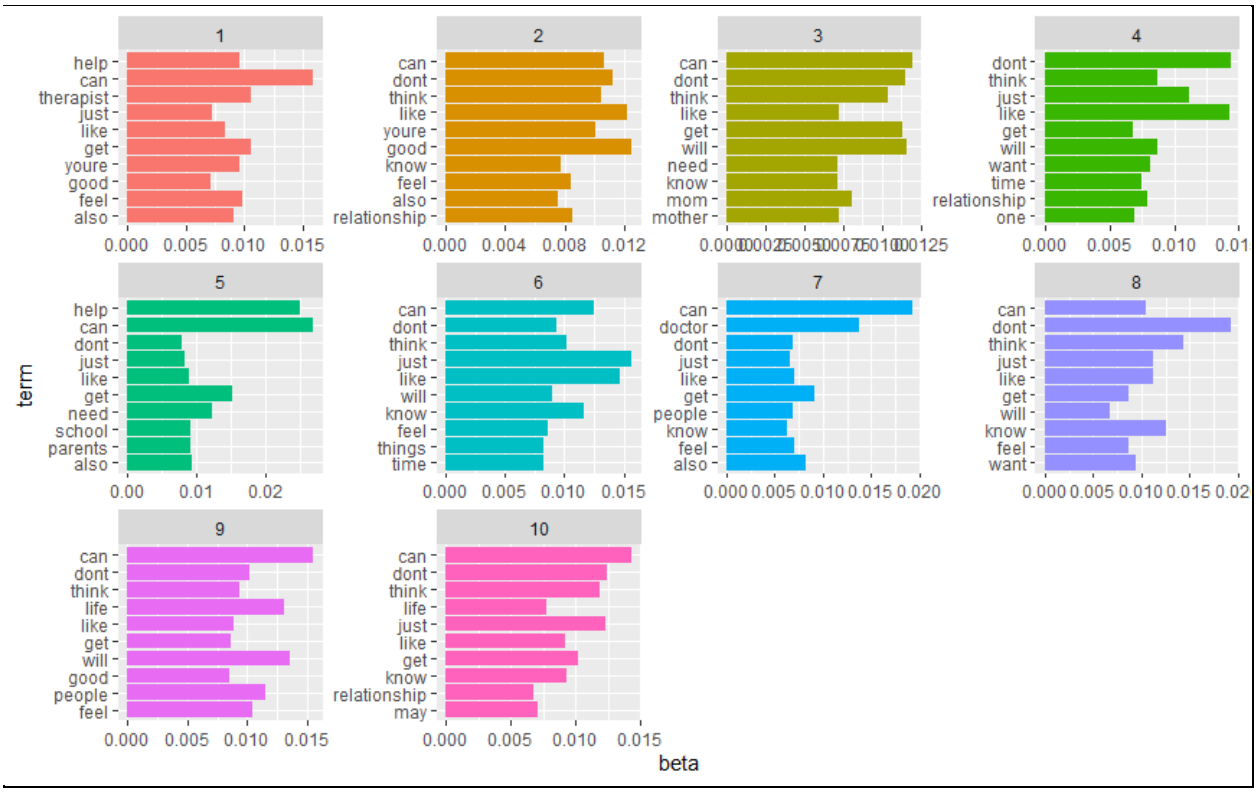
6.3.5. Based on the results recommend the number of topics that would be appropriate for this corpus.

I recommend, K=10 as I can find new words like love, mother, school, family.

7. Use the following code to perform topic-modeling on answers:

```
library(RTextTools)
library(tm)
library(wordcloud)
library(topicmodels)
library(slam)
data <- data[1:1000,] # We perform LDA on the rows 1 through 1000 in the data.
corpus <- Corpus(VectorSource(data$answers), readerControl=list(language="en"))
dtm <- DocumentTermMatrix(corpus, control = list(stopwords = TRUE, minWordLength = 2,
removeNumbers = TRUE, removePunctuation = TRUE,  stemDocument = TRUE))
rowTotals <- apply(dtm , 1, sum) #Find the sum of words in each Document
dtm.new   <- dtm[rowTotals> 0, ] #remove all docs without words
lda <- LDA(dtm.new, k = 10) # k is the number of topics to be found.
```

7.1. **(5 points)** The code above will create the beta scores for each document per topic (k = 10). Then create bar plots (similar to what we created in class) for each topic for 10 tokens (top_n(10, beta)). Paste the visualization below.

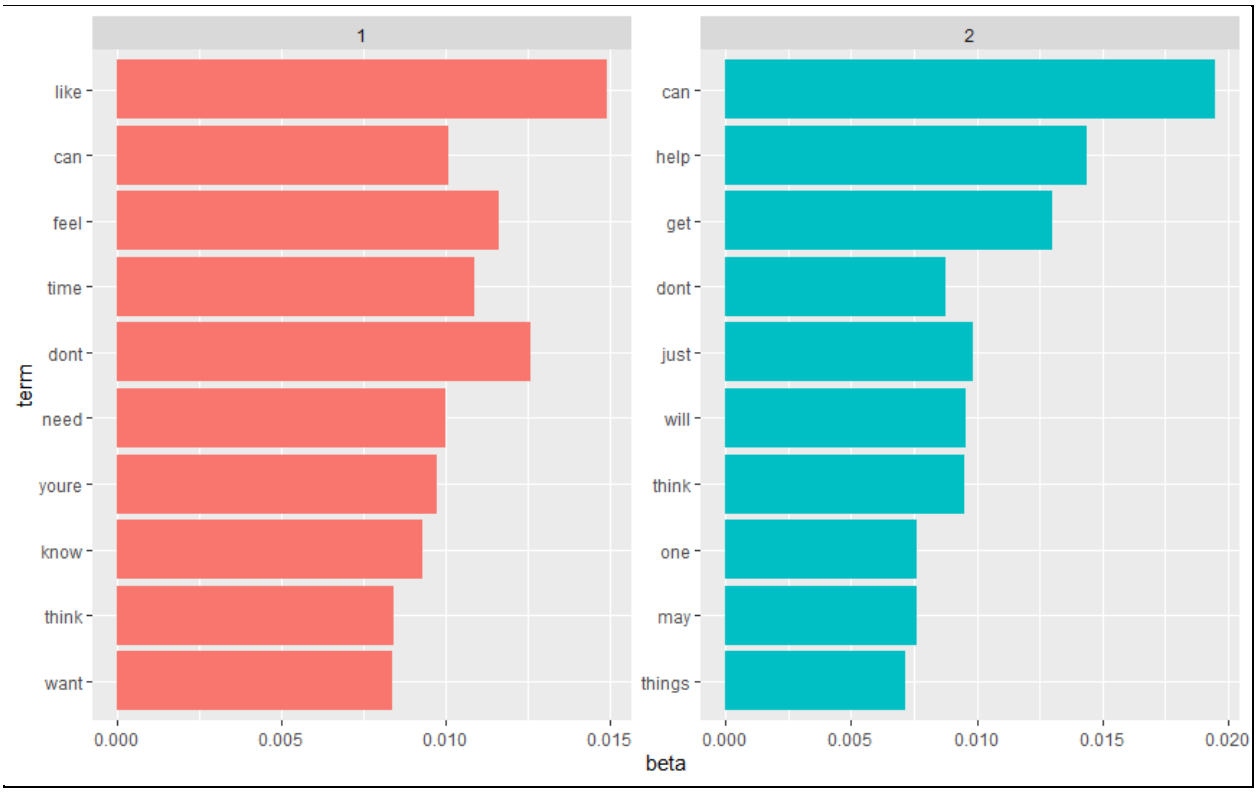


7.2. **(5 points)** Based on the visualization in 6.1., are the tokens in all topics similar? Then what can you say about k? Would you try a larger k or a smaller k?

The words like can, don', just, like, know, is common. Hence, we need larger K values to determine more about the topics.

7.3. **(10 points)** Repeat 6.1. with the following ks:

7.3.1. K = 2. Paste your visualization in the space below:



7.3.2. K = 8. Paste your visualization in the space below:

7.3.3. K = 11. Paste your visualization in the space below:

7.3.4. K = 14. Paste your visualization in the space below:

7.3.5. Based on the results recommend the number of topics that would be appropriate for this corpus.

I would recommend K = 14 as more new words like sex, relationship, doctor, parents appear in the visualization.

8. **(20 points)** Suppose that you are a researcher who works for National Institutes of Health (NIH). You are working on a project that aims to identify the most important reasons for mental disorders. Based on your analysis above, can we propose any hypothesis about the reasons for mental disorders in the society? Please explain.

Based on our text analysis the most frequent words that appear are feel, help, people, good, doctor, friends. Based on our word cloud the frequent positive words are "love, luck, trust, support, respect, honest etc.…" and the frequent negative words are "depress, hurt, stress, cheat, concern etc.…"
Thus, we can observe that all the words are related to people and their emotions. Thus, the reasons for mental disorders are mostly related to people and their relationships connected with the close ones. Thus, society that exhibit positive emotions such as love, luck, trust, support, honesty etc.…will help to reduce the mental disorders.

R-CODE

```
getwd()
setwd("C:/Users/Abdul Rahman/Documents/Train")
data_csv<-read.csv("C:/Users/Abdul Rahman/Documents/Train/psychcentral_data.csv")
library(data.table)
data = fread("C:/Users/Abdul Rahman/Documents/Train/psychcentral_data.csv",
          strip.white=T, sep=",", header=T, na.strings=c("","" ", "NA","nan", "NaN", "nannan"))
nrow(data)
ncol(data)
head(data, n=5)

library(tidytext)

text <- c("Because I could not stop for Death -",
        "He kindly stopped for me -",
        "The Carriage held but just Ourselves -",
        "and Immortality")
text
library(dplyr)
text_df <- data_frame(line = 1:4, text = text)
head(text_df)

text_df %>%
  unnest_tokens(word, text)

tidy_text <- data %>%
  unnest_tokens(word, q_content)
tidy_text[1:40]

data(stop_words)
tidy_text <- tidy_text %>%
  anti_join(stop_words)

tidy_text %>%
  count(word, sort = TRUE)

library(ggplot2)
tidy_text %>%
  count(word, sort = TRUE) %>%
  filter(n > 2000) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +
  coord_flip()
```

```
library(SnowballC)
tidy_text <- data %>%
  unnest_tokens(word, q_content) %>%
  mutate(word = wordStem(word))
data(stop_words)
tidy_text<-tidy_text %>%
  anti_join(stop_words)
tidy_text %>%
  count(word, sort=TRUE)
library(ggplot2)
tidy_text %>%
  count(word, sort = TRUE) %>%
  filter(n > 4000) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_bar(stat = "identity") +
  xlab(NULL) +
  coord_f

library(wordcloud)
tidy_text %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 200))

library(slam)
tidy_text %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 200))


library(reshape2)
tidy_text %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("#F8766D", "#00BFC4"),
                   max.words = 100)


library(RTextTools)
library(tm)
library(wordcloud)
library(topicmodels)
library(slam)
```

```
data <- data[1:1000,]
corpus <- Corpus(VectorSource(data$answers), readerControl=list(language="en"))

dtm1<-
DocumentTermMatrix(corpus,control=list(stopwords=TRUE,minWordLength=2,removeNumber=TR
UE,
                                       removePunctuation=TRUE,stemDocument=TRUE))
dtm1
rowTotals<-apply(dtm1,1,sum)
dtm.new1<-dtm1[rowTotals>0,]
lda1 <-LDA(dtm.new1,k=11)
da1

library(tidytext)
lda_td1<-tidy(lda1)
lda_td1
top_terms <- lda_td1 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

data <- data[1:1000,]
corpus <- Corpus(VectorSource(data$answers), readerControl=list(language="en"))

dtm1<-
DocumentTermMatrix(corpus,control=list(stopwords=TRUE,minWordLength=2,removeNumber=TR
UE,
                                       removePunctuation=TRUE,stemDocument=TRUE))
dtm1
rowTotals<-apply(dtm1,1,sum)
dtm.new1<-dtm1[rowTotals>0,]
lda1 <-LDA(dtm.new1,k=14)
da1

library(tidytext)
lda_td1<-tidy(lda1)
lda_td1
top_terms <- lda_td1 %>%
```

```
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```