Assignment 2

Aulia Dini Rafsanjani   Chia Wen Cheng   Wenqing Qian

```
library(RedditExtractoR)
library(tidyverse)
library(lubridate)
library(qdap)
library(SentimentAnalysis)
library(quanteda)
library(vader)
library(GGally)
library(wordcloud)
library(RColorBrewer)
library(wordcloud2)
library(tm)
library(kableExtra)
```

Data collection

```
vegan_subreddits <- find_subreddits(keywords = "vegan")
write.csv(x = vegan_subreddits, file = "vegan_subreddits.csv")
```

```
vegan_subreddits <- read.csv("vegan_subreddits.csv")
# head(vegan_subreddits)
n_subreddits <- nrow(vegan_subreddits)
# n_subreddits
```

For the first step, we set "vegan" as searching keyword to look for subreddits related to this topic and finally get 198 subreddits. Most of their names explicitly contain our keyword "vegan" (87.7907%), and the others are mainly large subreddits with relatively general topics, such as `r/funny` and `r/teenagers`.

```
vegan_subreddits1 <- vegan_subreddits[1:(n_subreddits/3), ]
vegan_subreddits2 <- vegan_subreddits[(n_subreddits/3+1):(n_subreddits/3*2), ]
vegan_subreddits3 <- vegan_subreddits[(n_subreddits/3*2+1):n_subreddits, ]
```

```
vegan_posts <- data.frame()
# Substitute "vegan_subreddits1" to "vegan_subreddits2" or "vegan_subreddits3"
for (sr in vegan_subreddits1$subreddit) {
  new_post <- find_thread_urls(keywords = "vegan", subreddit = sr, period = "week")
  if (!is.null(ncol(new_post))) {   # If there are any relevant posts
    vegan_posts <- vegan_posts %>%
      bind_rows(new_post)
  }
```

```r
  # Sys.sleep(2)
}
vegan_posts <- vegan_posts %>%
  drop_na() %>%
  distinct() %>%
  mutate(collect_time = now("EST"))
date_today <- str_c(str_extract_all(ymd(today("EST")), "\\d")[[1]], collapse = "")
write.csv(x = vegan_posts,
          file = paste0("posts/vegan_posts", date_today, "_1.csv"))
head(vegan_posts)
```

```r
vegan_content <- get_thread_content(urls = vegan_posts$url)

vegan_threads <- vegan_content$threads
write.csv(x = vegan_threads,
          file = paste0("threads/vegan_threads", date_today, "_1.csv"))
head(vegan_threads)

vegan_comments <- vegan_content$comments
write.csv(x = vegan_comments,
          file = paste0("comments/vegan_comments", date_today, "_1.csv"))
head(vegan_comments)
```

And then we split the subreddit list into three parts and collect posts and comments by searching "vegan" inside each subreddit for a whole week, from February 27 to March 5. As mentioned above, our list includes some subreddits with very broad topics and numerous followers. If we don't narrow down the searching scope via adding keywords, we will end up in retrieving a lot of irrelevant information. For this reason, we exclude the data collected on February 26.

Data cleaning and pre-processing

```r
merge_all <- function(type) {
  dates <- c("0305", "0304", "0303", "0302", "0301", "0228", "0227")
  filenames <- paste0(type, "/", "vegan_", type, "2023", dates, "_", 1:3, ".csv")

  dt <- data.frame()
  for (f in filenames) {
    dt_day <- read.csv(f)
    dt <- dt %>%
      bind_rows(dt_day) %>%
      distinct()

    if (type != "comments") {
      dup_flag <- duplicated(dt[, c("title", "text", "subreddit", "url")])
      dt <- dt %>%
        filter(!dup_flag)
    } else {
      dup_flag <- duplicated(dt[, c("url", "author", "comment")])
      dt <- dt %>%
        filter(!dup_flag)
```

```
    }
  }

  return(dt)
}
```

```
# posts <- merge_all(type = "posts")
# threads <- merge_all(type = "threads")
# comments <- merge_all(type = "comments")

posts <- read.csv("posts.csv")
threads <- read.csv("threads.csv")
comments <- read.csv("comments.csv")
data.comments <- comments   # for later use
```

After combining all the data collected, we obtain 1297 posts and 32672 comments in total. Since there might be some overlapping across data collected in each day, we have removed duplicated items according to content, subreddit, and url for posts/threads and url, author, and comment for comments (i.e., these variables are used to identify a certain post or comment). We keep posts and comments collected most recently. Given our strict searching conditions and the small amount of posts returned, we don't find many posts that are not related to our topic when going through the dataset manually, so no special investigation and data selection is required. Since comments are replying to posts highly relevant to our topic, we may also assume that all the comments are pertinent, too.

Exploratory analysis

We are going to do a series of exploration within data we collected and cleaned. The exploratory analysis starts from re-encoding the time the posts, threads, and comments were published.

The timestamp variable is given as a UNIX timestamp, the number of seconds from 1/1/70. So we first convert this variable into a date and time.

```
posts$datetime <- as_datetime(posts$timestamp)
head(posts[, c("date_utc", "datetime")])
```

```
    date_utc            datetime
1 2023-02-28 2023-02-28 19:17:48
2 2023-03-02 2023-03-02 20:14:48
3 2023-03-05 2023-03-05 12:50:25
4 2023-03-05 2023-03-05 13:06:01
5 2023-03-04 2023-03-04 07:40:18
6 2023-02-26 2023-02-26 13:02:00
```

```
threads$datetime <- as_datetime(threads$timestamp)
head(threads[, c("date", "datetime")])
```

```
        date            datetime
1 2023-02-28 2023-02-28 19:17:48
2 2023-03-02 2023-03-02 20:14:48
3 2023-03-05 2023-03-05 12:50:25
```

```
4 2023-03-05 2023-03-05 13:06:01
5 2023-03-04 2023-03-04 07:40:18
6 2023-02-26 2023-02-26 13:02:00
```

```
comments$datetime <- as_datetime(comments$timestamp)
head(comments[, c("date", "datetime")])
```

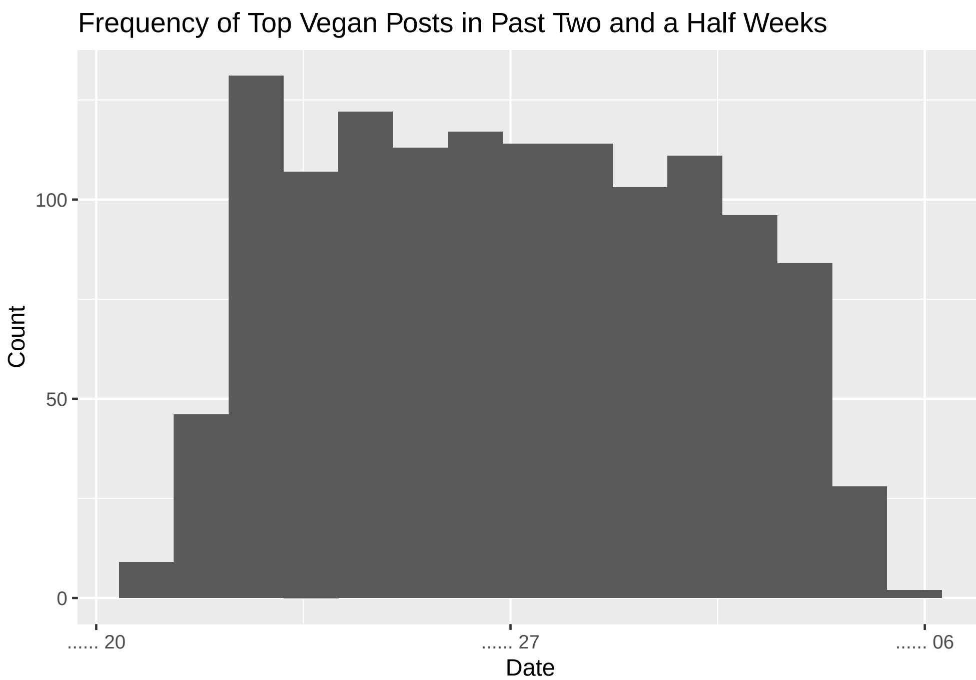
```
        date            datetime
1 2023-03-01 2023-03-01 05:18:54
2 2023-03-01 2023-03-01 06:20:14
3 2023-03-02 2023-03-02 17:52:01
4 2023-03-02 2023-03-02 17:53:38
5 2023-03-01 2023-03-01 08:38:22
6 2023-03-01 2023-03-01 11:25:43
```

With the readible publish time of each post, thread, and comment, we are interested in knowing that in the past two and a half weeks, when were the top vegan subreddit posts posted.

```
posts %>%
  group_by(date_utc) %>%
  summarize(num_posts = n())
```

```
# A tibble: 14 x 2
   date_utc   num_posts
   <chr>          <int>
 1 2023-02-20         1
 2 2023-02-21        25
 3 2023-02-22       132
 4 2023-02-23       128
 5 2023-02-24       126
 6 2023-02-25       120
 7 2023-02-26       126
 8 2023-02-27       132
 9 2023-02-28       134
10 2023-03-01       111
11 2023-03-02       113
12 2023-03-03        91
13 2023-03-04        56
14 2023-03-05         2
```

```
ggplot(data = posts, mapping = aes(x = datetime)) +
  geom_histogram(bins = 15) +
  xlab("Date") +
  ylab("Count") +
  ggtitle("Frequency of Top Vegan Posts in Past Two and a Half Weeks")
```



The duration of our data collection lasted from February 20 to March 6. (We restate this just to make up the flaws in the name of x-axis in the histogram.) From the histogram, we know that throughout the past two weeks from February 20 to March 6, there were the most posts related to vegan published on February 28. February 22 and 27 had similar quantity of posts related to vegan and were the days with second most posts among the 15 days. February 23 were in the third place of number of posts regarding vegan. This phenomenon indicating more discussions on Reddit about vegan around the end of February may be related to the fact that February is the Vegan Cuisine Month. Furthermore, the trend that more vegan-related posts were observed in February comparing to March is a side evidence of the possible explanation of the impact of Vegan Cuisine Month.

We are also interested in revealing the relationship between score and hour of the posting.

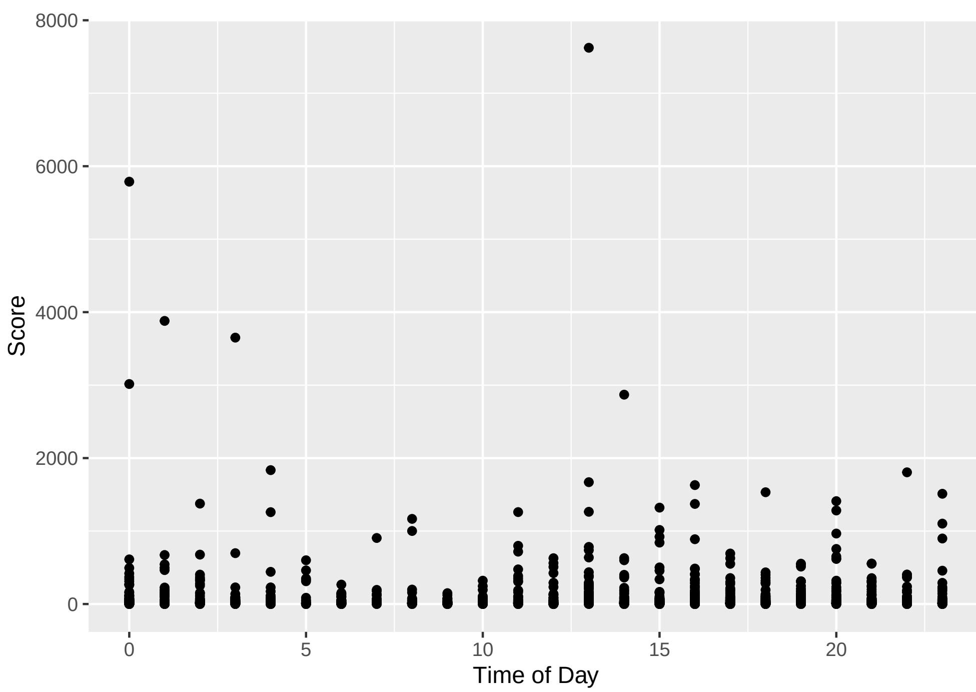
```
threads$datetime <- as_datetime(threads$timestamp)
threads$timeofday <- format(as.POSIXct(threads$datetime), format = "%H")

threads %>%
  group_by(timeofday) %>%
  summarize(median_score = median(score)) %>%
  arrange(desc(median_score)) %>%
  print(n = 100)
```

```
# A tibble: 24 x 2
   timeofday median_score
   <chr>            <dbl>
 1 11                36
 2 13                34
 3 21                33.5
 4 20                32.5
 5 00                31
 6 15                31
 7 16                29.5
```

```
 8 19                 27
 9 02                 26
10 17                 25
11 18                 24.5
12 06                 23
13 03                 22
14 04                 22
15 14                 22
16 12                 21
17 07                 19
18 23                 19
19 10                 18
20 01                 17
21 05                 16
22 22                 16
23 09                 15
24 08                 12
```

```r
ggplot(data = threads, mapping = aes(x = as.numeric(timeofday), y = score)) +
  geom_point() +
  xlab("Time of Day") +
  ylab("Score")
```



According to the plot, scores are similar across hours of a day, while late mornings from 5-9am seems to have the lowest average scores and late nights from 9pm-12am seems to have the highest average scores. Several extreme outliers are seen at 1pm, 12am, 1am, and 3am.

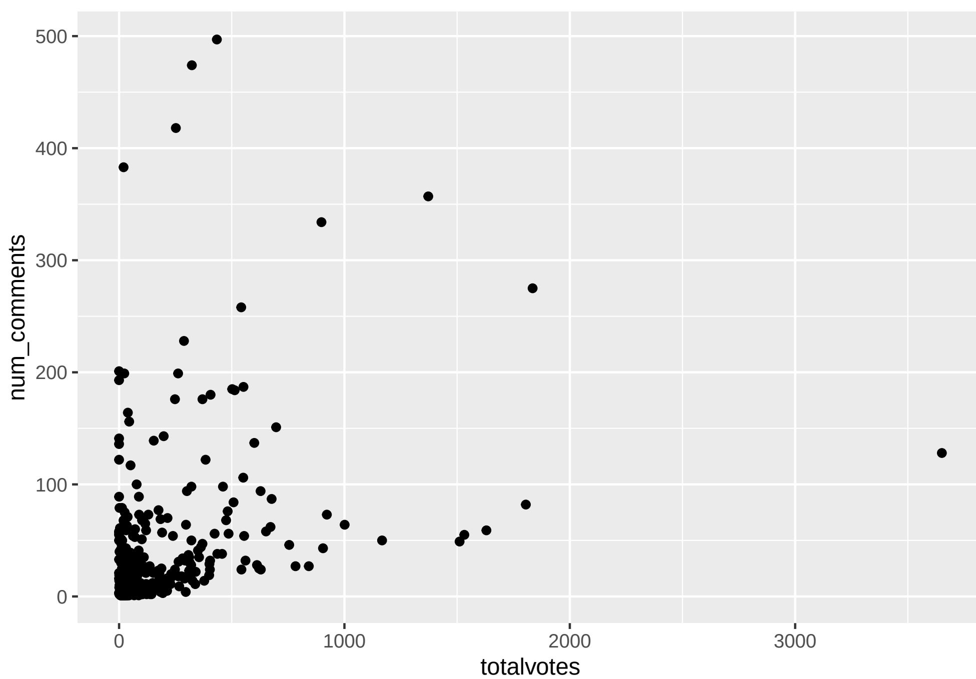
Next on, we would like to know if comments with more total votes have more comments.

```
threads$totalvotes <- threads$upvotes + threads$downvotes
comments$dummy <- 1
comments <- aggregate(list("num_comments" = comments$dummy),
                      list("url" = comments$url),
                      sum)
head(comments)
```

```
                                                                                                                        url
1     https://www.reddit.com/r/AntiVegan/comments/11b9n28/are_you_an_exvegan_what_made_you_go_vegan_and/
2      https://www.reddit.com/r/AntiVegan/comments/11c2wqw/i_was_raised_by_vegan_parents_but_im_not_now/
3 https://www.reddit.com/r/AntiVegan/comments/11cabpb/tired_of_my_vegan_roomate_no_regrets_i_am_writing/
4                             https://www.reddit.com/r/AntiVegan/comments/11cqdmd/okay_serious_question/
5           https://www.reddit.com/r/AntiVegan/comments/11cry0v/how_many_times_have_you_guys_heard_this/
6                                    https://www.reddit.com/r/AntiVegan/comments/11d7gud/my_lunch_today/
  num_comments
1            8
2           35
3           27
4            6
5           30
6            5
```

```
comments <- merge(comments, threads, by = "url", all.x = TRUE)

ggplot(data = comments, mapping = aes(x = totalvotes, y = num_comments)) +
  geom_point()
```




```
cor(comments$totalvotes,comments$num_comments)
```

```
[1] 0.3695229
```

```r
summary(lm(num_comments ~ totalvotes, data = comments))
```

```
Call:
lm(formula = num_comments ~ totalvotes, data = comments)

Residuals:
    Min      1Q  Median      3Q     Max 
-169.73  -22.48  -17.02   -1.98  441.30 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 23.046513   2.995746   7.693 9.33e-14 ***
totalvotes   0.075236   0.008969   8.389 6.56e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.98 on 445 degrees of freedom
Multiple R-squared:  0.1365,    Adjusted R-squared:  0.1346 
F-statistic: 70.37 on 1 and 445 DF,  p-value: 6.562e-16
```

It is difficult to recognize any trend from the plot directly. However, the correlation coefficient between number of total votes and number of comments is 0.3695229, which is weakly correlated.

The linear regression model shows an intercept at 23.046513, which points out the number of comments when no total vote is performed. The slope of 0.075236 implies a positive relationship between the number of comments and the total votes. When the total vote increases by 1 unit, the number of comments is estimated to be associated with an increase of 0.075236 unit. This is statistically significant at 95% confidence level.

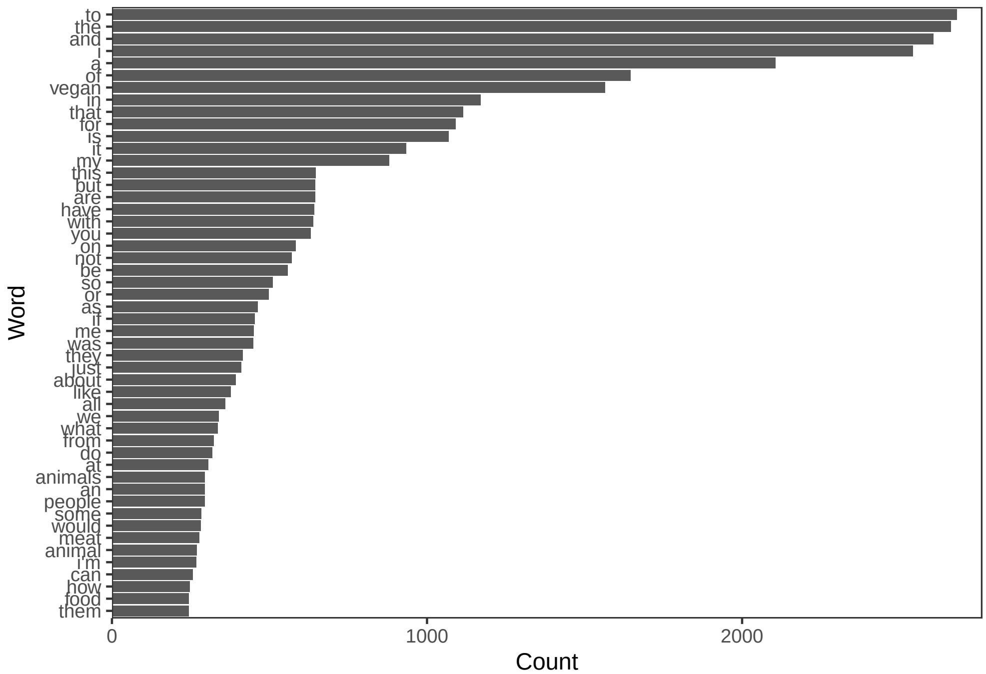
Frequent terms

Since only some posts contain text, we create a new field that combines the post title and post text into one string.

```r
posts$title_text <- paste(posts$title, posts$text)
```

We first examine the top 50 most frequently used words in posts.

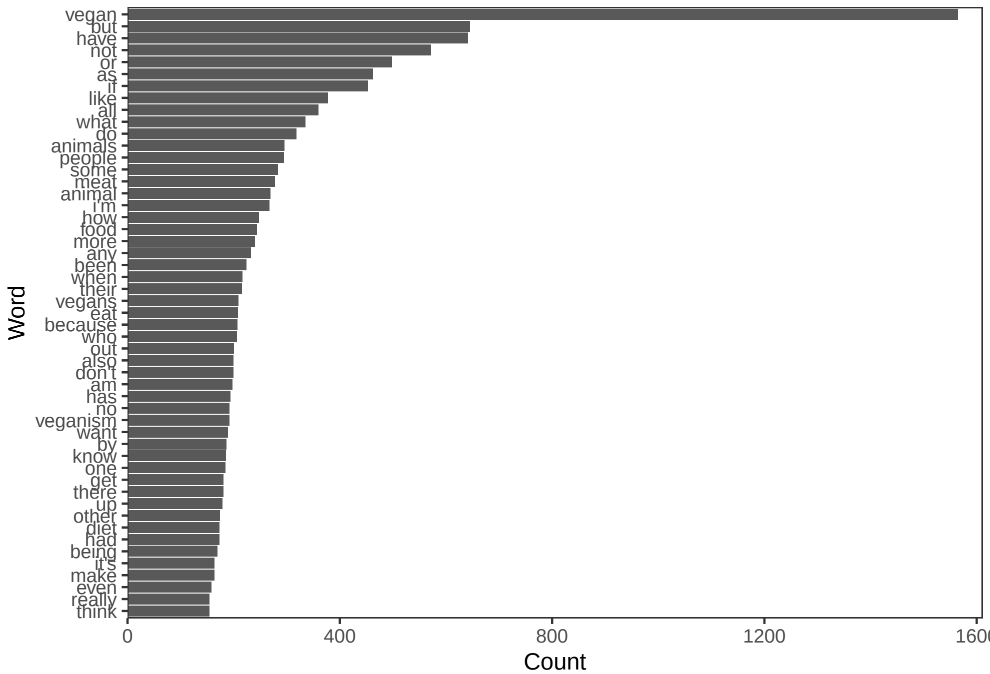
```r
frequent_terms <- freq_terms(posts$title_text, 50)
plot(frequent_terms)
```



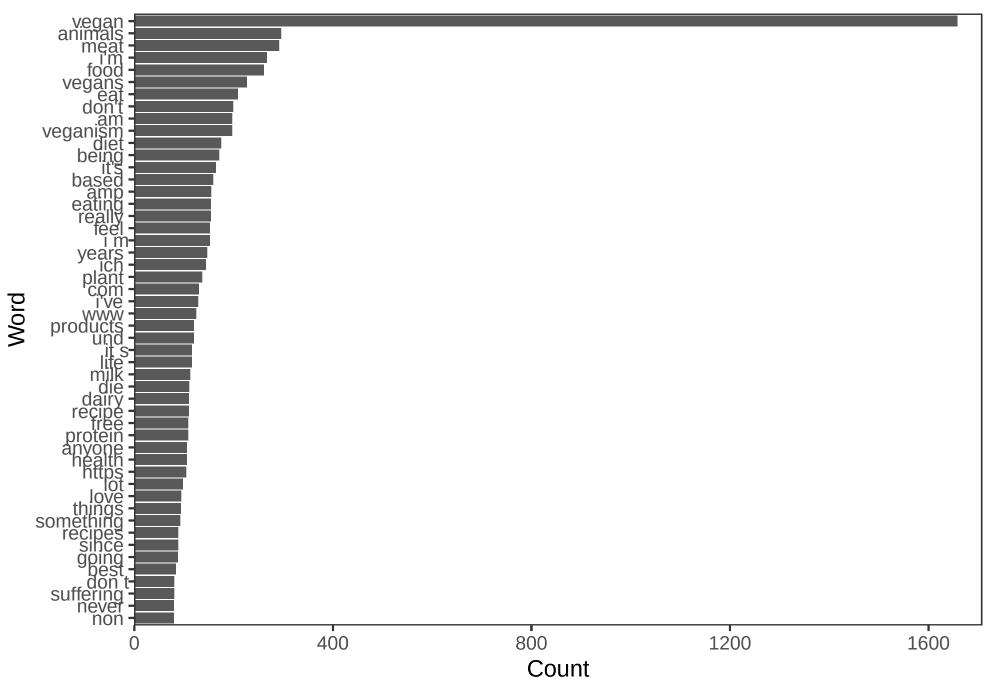
The top 50 most frequently used words are very generally seen in daily life conversations, including subjects (i.e. I, you, we, she, etc.), prepositions (i.e. to, of, in, for, from, on, etc.), verbs (i.e. do, make, have, etc.), auxiliary verbs (i.e. be, will, can, etc.), and many unspecific words that can be found in any occasions. The two frequently seen terms in posts we scraped that are highly related to our topic of vegan are "vegan" and "animals."

We then intend to exclude words that are too generally spoken around.

```
frequent_terms_ex <- freq_terms(posts$title_text, 50,
                                stopwords = c("I", "you", "we", "she", "he", "they", "it",
                                              "a", "an", "the", "is", "are", "her", "his",
                                              "us", "our", "your", "ours", "yours", "theirs",
                                              "them", "my", "just", "can", "on", "in", "of",
                                              "to", "from", "with", "at", "for", "about",
                                              "was", "were", "will", "would", "so", "and",
                                              "that", "this", "be", "me"))
plot(frequent_terms_ex)
```




```
bagVegans <- posts$title_text %>%
  iconv ("latin1", "ASCII", sub = "") %>%
  scrubber () %sw%
  qdapDictionaries::Top200Words
frequent_terms_ex1 <- freq_terms(bagVegans, 50)
plot(frequent_terms_ex1)
```



"Vegan" became the most frequently used term in the posts after excluding some of the generally seen words.

“Meat,” “animal(s),” “diet,” and “dairy” are normal to us because these are the most popular concepts or questions people usually have when it comes to vegan. Some informal English are also highly typed in posts. Interestingly, “anarchy,” “recipe,” and “donate” are also used with high frequency. It also turns out that using the package to get rid of words often seen in daily dialogue is more effective than filtering “stopwords” individually.

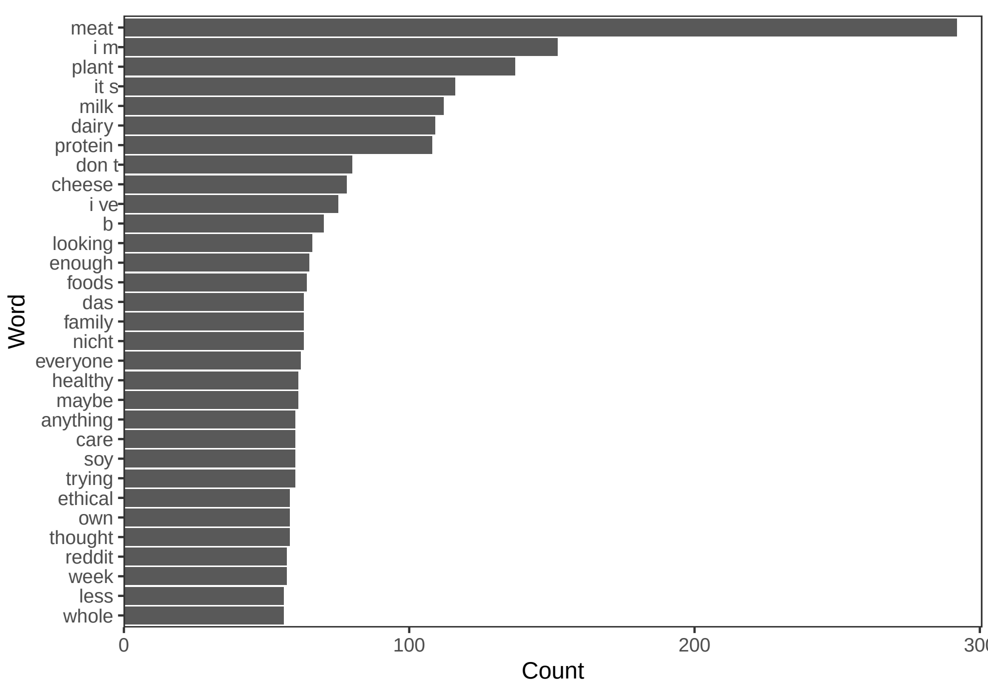
We would then like to generate a word cloud, which visually presents the most frequently used terms by placing them around the center and magnifying the size.

```r
# Create a vector containing only the text
vegan_text <- posts$text
# Create a corpus
vegan_docs <- Corpus(VectorSource(vegan_text))
# Clean the text
docs <- vegan_docs %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
docs <- tm_map(vegan_docs, content_transformer(tolower))
docs <- tm_map(vegan_docs, removeWords, stopwords("english"))
# Create a document-term-matrix
dtm <- TermDocumentMatrix(docs)
matrix <- as.matrix(dtm)
words <- sort(rowSums(matrix),decreasing = TRUE)
df <- data.frame(word = names(words),freq = words)
# Generate word cloud
set.seed(798) # for reproducibility
wordcloud(words = df$word, freq = df$freq, min.freq = 1,
          max.words = 200, random.order = FALSE, rot.per = 0.35,
          colors = brewer.pal(8, "Dark2"), scale = c(5, 0.25))
```

With these code lines, non-formal English terms or expressions are excluded from the result. Thus, the report is different comparing to the previous one. "Vegan" appears to remain to be the most frequently used in posts after filtering. There are some interesting words in our result, including "philosophical," "effective," "sugar," "saving," and more.

In the last session of exploratory analysis, we would like to see What kind of food appears more frequently in the posts we've collected.

```
bagVegans <- posts$title_text %>%
  iconv ("latin1", "ASCII", sub = "") %>%
  scrubber () %sw%
  qdapDictionaries::Top200Words
frequent_terms_ex2 <- freq_terms(bagVegans, 30,
                                 stopwords = c("animals", "i'm", "eat", "vegans",
                                              "veganism", "don't", "diet", "am",
                                              "being", "amp", "it's", "based",
                                              "i' m", "really", "eating", "ich",
                                              "years", "feel", "com", "i've",
                                              "products", "www", "life", "it' s",
                                              "recipe", "anyone", "https", "never",
                                              "things", "die", "free", "suffering",
                                              "und", "since", "love", "something",
                                              "going", "lot", "while", "best",
                                              "don' t", "health", "saying", "few",
                                              "can't", "actually", "got", "thanks",
                                              "making", "i m", "it s", "i ve", "don t",
                                              "recipes", "always", "without", "non",
                                              "better", "someone", "vegetarian",
                                              "started", "those", "every", "comments",
                                              "vegan", "food"))
plot(frequent_terms_ex2)
```



Since foods can be mentioned in multiple ways–broadly in categories or cuisines, or with details by names or nutrition of a specific food, we limit the result to 30 words at the end. However, throughout the exploration process looking for words to exclude, we overwhelmingly deployed 50 words in a round, and conducted 5 rounds of exploration, aggregatively expanding the "stopword list" to more than 30 vocabulary. The most frequently mentioned foods are "meat," "plant," "protein," "dairy," "milk," "cheese," and "soy."

Sentiment analysis

We do the sentiment analysis using the dictionary based method.

```
# check column names
colnames(posts)
# check subreddit
table(posts$subreddit)
```

```
# sentiment analysis
sentiments <- analyzeSentiment(iconv(as.character(posts$title), to = "UTF-8"))
head(sentiments)
```

```
  WordCount SentimentGI NegativityGI PositivityGI SentimentHE NegativityHE
1        11         0.0          0.0          0.0           0            0
2         2        -0.5          0.5          0.0           0            0
3         4         0.0          0.0          0.0           0            0
4         2         0.5          0.0          0.5           0            0
5         5         0.0          0.0          0.0           0            0
6         6         0.0          0.0          0.0           0            0
  PositivityHE SentimentLM NegativityLM PositivityLM RatioUncertaintyLM
1            0         0.0          0.0            0                  0
2            0        -0.5          0.5            0                  0
3            0         0.0          0.0            0                  0
4            0         0.0          0.0            0                  0
5            0         0.0          0.0            0                  0
6            0         0.0          0.0            0                  0
  SentimentQDAP NegativityQDAP PositivityQDAP
1           0.0            0.0            0.0
2          -0.5            0.5            0.0
3           0.0            0.0            0.0
4           0.5            0.0            0.5
5           0.0            0.0            0.0
6           0.0            0.0            0.0
```

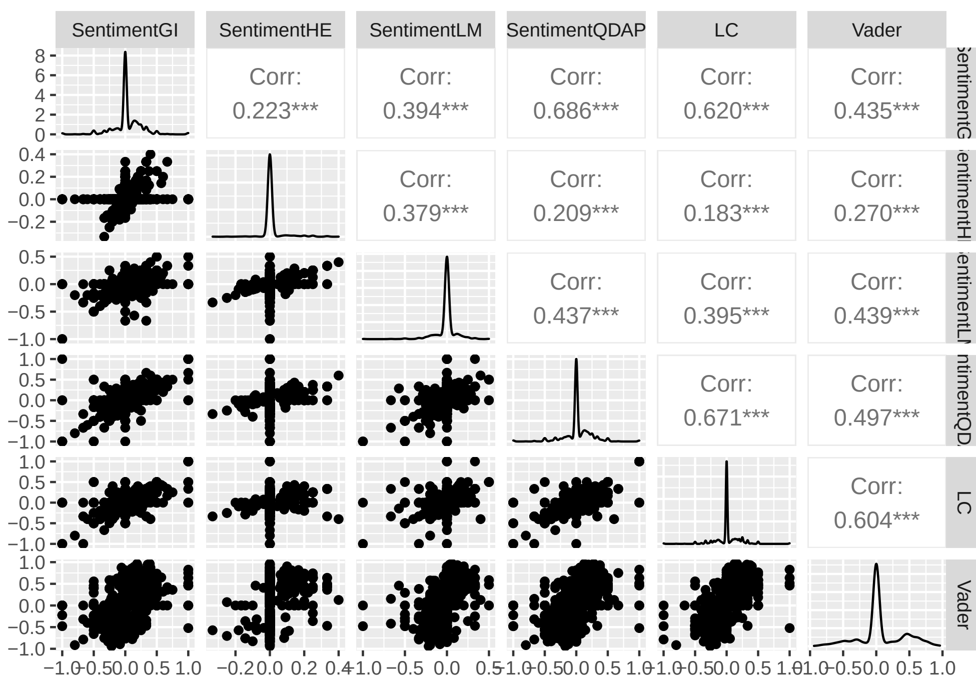
```
# check dictionary GI
DictionaryGI$positive[1:100]
DictionaryGI$negative[1:100]

# check dictionary LSD
data_dictionary_LSD2015$negative[1:50]
data_dictionary_LSD2015$positive[1:50]
data_dictionary_LSD2015$neg_positive[1:50]
data_dictionary_LSD2015$neg_negative[1:50]
```

```r
# calculate SentimentLC
tokenized <- tokens_lookup(tokens(posts$title),
                           dictionary = data_dictionary_LSD2015,
                           exclusive = FALSE)
sentiments$LCpos <- sapply(tokenized, function(x) {
  sum(x == "POSITIVE") - sum(x == "NEG_POSITIVE") + sum(x == "NEG_NEGATIVE")
  })
sentiments$LCneg <- sapply(tokenized, function(x) {
  sum(x == "NEGATIVE") - sum(x == "NEG_NEGATIVE") + sum(x == "NEG_POSITIVE")
  })
sentiments$LC <- (sentiments$LCpos - sentiments$LCneg) / sentiments$WordCount
```

```r
# calculate sentiments using vader
vader_scores <- vader_df(posts$title)
sentiments$Vader <- vader_scores$compound
```

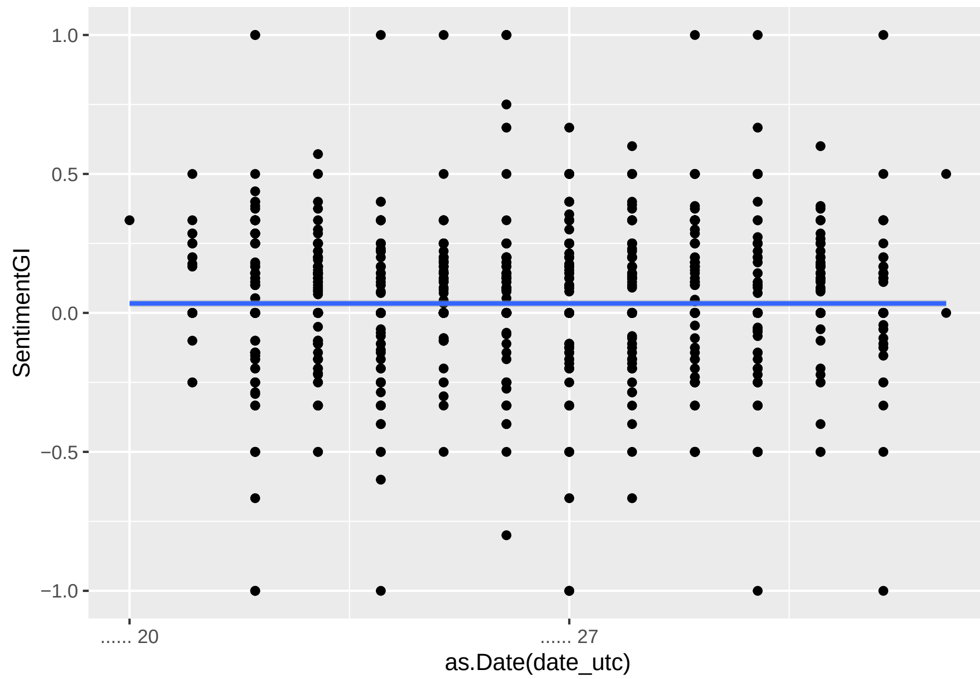
```r
# compare different sentiments
with(sentiments,
     ggpairs(data.frame(SentimentGI, SentimentHE, SentimentLM, SentimentQDAP, LC, Vader)))
```



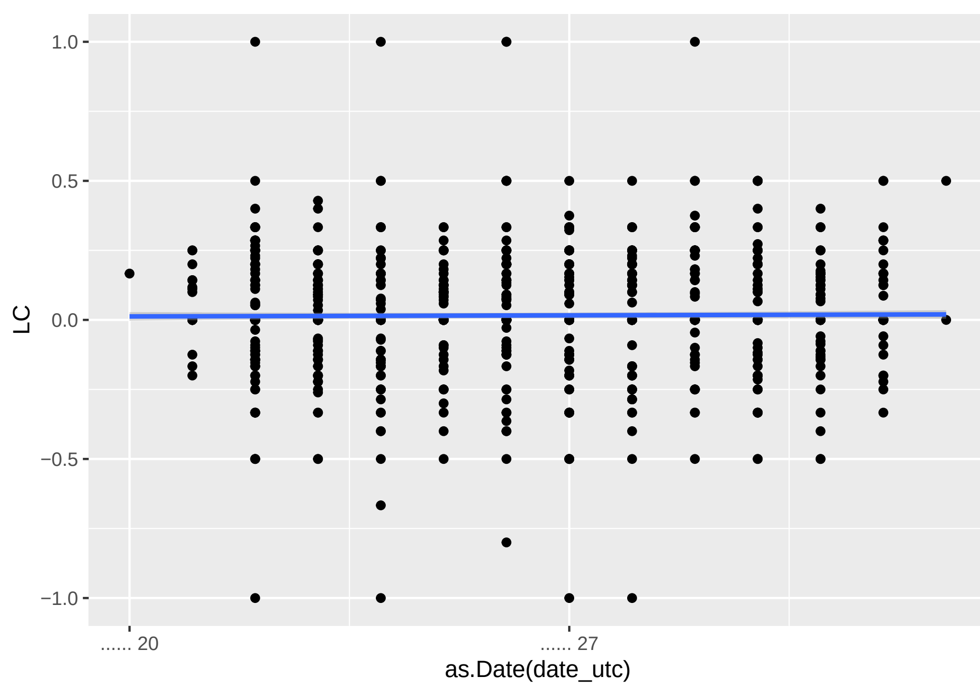
We decide to pick sentiment sentimentGI and sentimentLC and analyze the difference.

```r
# merge data posts and data sentiments
all_posts_data <- cbind(posts, sentiments)
```

```r
# plot of sentimentGI over time
ggplot(data = all_posts_data, mapping = aes(x = as.Date(date_utc), y = SentimentGI)) +
  geom_point() +
  geom_smooth()
```




```
# plot of LC over time
ggplot(data = all_posts_data, mapping = aes(x = as.Date(date_utc), y = LC)) +
  geom_point() +
  geom_smooth()
```

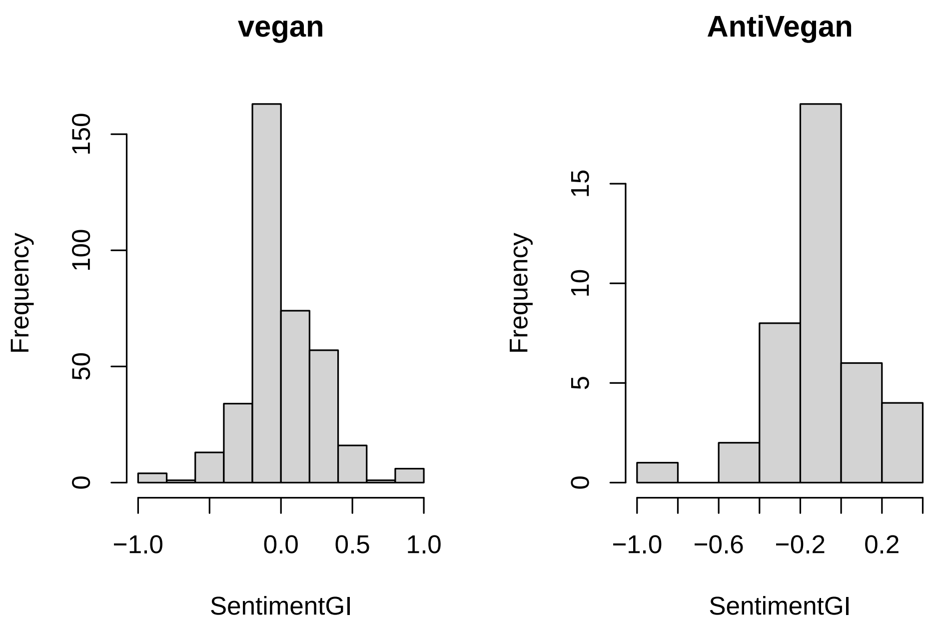


Plot of sentimentGI and LC showed a similar pattern, which is the straight line, means that the sentiments of posts about vegan was stable from February 20, 2023 to March 5, 2023.

Additional questions

Question 1: Is there any difference in sentiment between `r/vegan` and `r/AntiVegan`? (using GI and LC)

```r
# histogram and statistical test, sentimentGI
par(mfrow = c(1,2))
hist(x = all_posts_data$SentimentGI[all_posts_data$subreddit == "vegan"],
     main = "vegan", xlab = "SentimentGI")
hist(x = all_posts_data$SentimentGI[all_posts_data$subreddit == "AntiVegan"],
     main = "AntiVegan", xlab = "SentimentGI")
```

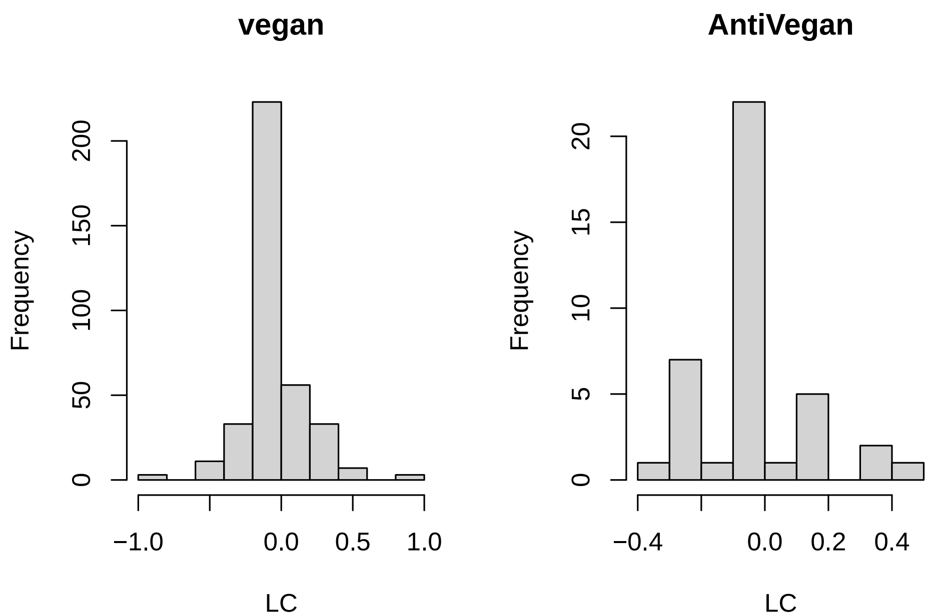



```r
t.test(all_posts_data$SentimentGI[all_posts_data$subreddit == "vegan"],
       all_posts_data$SentimentGI[all_posts_data$subreddit == "AntiVegan"])
```

```
    Welch Two Sample t-test

data:  all_posts_data$SentimentGI[all_posts_data$subreddit == "vegan"] and all_posts_data$SentimentGI[al
t = 2.6476, df = 49.462, p-value = 0.01085
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.02638204 0.19240938
sample estimates:
  mean of x   mean of y 
 0.05505949 -0.05433622 
```

```r
# histogram and statistical test, LC sentiment
par(mfrow = c(1,2))
hist(x = all_posts_data$LC[all_posts_data$subreddit == "vegan"],
     main = "vegan", xlab = "LC")
hist(x = all_posts_data$LC[all_posts_data$subreddit == "AntiVegan"],
     main = "AntiVegan", xlab = "LC")
```




```r
t.test(all_posts_data$LC[all_posts_data$subreddit == "vegan"],
       all_posts_data$LC[all_posts_data$subreddit == "AntiVegan"])
```

```
    Welch Two Sample t-test

data:  all_posts_data$LC[all_posts_data$subreddit == "vegan"] and all_posts_data$LC[all_posts_data$subre
t = 0.30406, df = 52.812, p-value = 0.7623
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.05065464  0.06875489
sample estimates:
   mean of x    mean of y
 0.004793267 -0.004256854
```

Based on the histogram, there is a different pattern of “AntiVegan” sentiment using sentimentGI and sentimentLC.

The statistical test (t-test) for sentimentGI and sentimentLC showed a different result. SentimentGI showed that there is a significant difference between “vegan” sentiment and “AntiVegan” sentiment, with the significance level of 95%. While sentimentLC showed not significant result.

Question 2: What are subreddits having a highest average of sentiment score and lowest average of sentiment score?

```
# table of subreddit and sentimentsGI, sort
subreddits_sentiments_GI <- aggregate(all_posts_data$SentimentGI ~ all_posts_data$subreddit,
                                      all_posts_data, mean)
sort_subreddits_sentiments_GI <- subreddits_sentiments_GI[
  order(subreddits_sentiments_GI$"all_posts_data$SentimentGI", decreasing = TRUE),
  ]
names(sort_subreddits_sentiments_GI) <- c("subreddit", "SentimentGI")
sort_subreddits_sentiments_GI
```

```
              subreddit  SentimentGI
34          vegancheese  0.333333333
52           VeganIndia  0.333333333
1          1200isplenty  0.231990232
35     vegancheesemaking  0.202614379
43          VeganFashion  0.196581197
42    VeganEtFrancophone  0.194444444
80           vegetarian  0.187500000
21            teenagers  0.168128026
57             veganita  0.166666667
58             Veganity  0.140540016
69             veganr4r  0.125000000
76           vegantravel  0.125000000
61           VeganLobby  0.098095238
27          vegan_travel  0.081730769
36           VeganChill  0.080772006
16                memes  0.080000000
71         veganrecipes  0.073998114
5             AskReddit  0.072077922
28        VeganActivism  0.070772283
26           Vegan_Food  0.069106026
47     Veganforbeginners  0.068752255
20      Showerthoughts  0.062500000
33          VeganBeauty  0.061309524
59           Veganivore  0.055555556
24                vegan  0.055059492
45            VeganFood  0.053147922
7                Baking  0.051360544
74         VeganSeattle  0.050000000
10      EatCheapAndVegan  0.045085470
67        veganparenting  0.043386243
2       52weeksofcooking  0.040755595
29    VeganAntinatalists  0.035714286
37       vegancirclejerk  0.034963567
30         veganarchism  0.033333333
13       glutenfreevegan  0.032738095
48 VeganForCircleJerkers  0.032716049
14           highvegans  0.031250000
77              veganuk  0.029455470
46         VeganFoodPorn  0.028193284
38   vegancirclejerkchat  0.025000000
```

```
3          AmItheAsshole  0.022348485
44          veganfitness  0.019928915
73          veganscience  0.016806723
19   ShittyVeganFoodPorn  0.011936468
18        shittyfoodporn  0.009557110
8              dankmemes  0.000000000
22 thatveganteachersucks  0.000000000
25          Vegan__Sensei  0.000000000
41              VeganDiet  0.000000000
49            vegangaming  0.000000000
50         vegangifrecipes  0.000000000
53           veganinjapan  0.000000000
54           Veganinspire  0.000000000
55           VeganIreland  0.000000000
60              veganketo  0.000000000
64           veganmexican  0.000000000
66         vegannutrition  0.000000000
70             VeganRamen  0.000000000
78         VeganWeightGain  0.000000000
11               exvegans -0.005739884
40                VeganDE -0.008688512
9            DebateAVegan -0.015476190
56               Veganism -0.019097222
12               FoodPorn -0.022222222
32            VeganBaking -0.027777778
23       unpopularopinion -0.027777778
63              veganmemes -0.050000000
6               AskVegans -0.051190476
31                VeganAT -0.051282051
62           veganmealprep -0.052631579
4               AntiVegan -0.054336219
81          Vegetarianism -0.055555556
72           veganrunners -0.062500000
17          PlantBasedDiet -0.097718254
39            vegancooking -0.111111111
65                VeganNL -0.166666667
68              veganpets -0.166666667
15                  Jokes -0.216666667
79         VeganWeightLoss -0.250000000
51   vegangranolamomsnark -0.333333333
75    VeganTeacherHateClub -0.333333333
```

```
subreddits_sentiments_LC <- aggregate(all_posts_data$LC ~ all_posts_data$subreddit,
                                      all_posts_data, mean)
sort_subreddits_sentiments_LC <- subreddits_sentiments_LC[
  order(subreddits_sentiments_LC$"all_posts_data$LC", decreasing = TRUE),
  ]
names(sort_subreddits_sentiments_LC) <- c("subreddit", "LC")
sort_subreddits_sentiments_LC
```

```
              subreddit          LC
79      VeganWeightLoss  0.250000000
42    VeganEtFrancophone  0.194444444
52           VeganIndia  0.166666667
```

16	memes	0.150000000
1	1200isplenty	0.133089133
35	vegancheesemaking	0.130718954
69	veganr4r	0.125000000
76	vegantravel	0.125000000
80	vegetarian	0.125000000
58	Veganity	0.113636364
27	vegan_travel	0.112980769
34	vegancheese	0.111111111
74	VeganSeattle	0.100000000
78	VeganWeightGain	0.100000000
29	VeganAntinatalists	0.098214286
47	Veganforbeginners	0.093901064
43	VeganFashion	0.088319088
26	Vegan_Food	0.087416684
10	EatCheapAndVegan	0.086965812
36	VeganChill	0.066119991
45	VeganFood	0.064072962
44	veganfitness	0.063149411
21	teenagers	0.062887113
14	highvegans	0.062500000
71	veganrecipes	0.054701995
67	veganparenting	0.040740741
61	VeganLobby	0.038095238
7	Baking	0.037074830
46	VeganFoodPorn	0.034865561
13	glutenfreevegan	0.032738095
72	veganrunners	0.031250000
28	VeganActivism	0.027815934
5	AskReddit	0.025000000
77	veganuk	0.023661521
20	Showerthoughts	0.020833333
2	52weeksofcooking	0.020379000
17	PlantBasedDiet	0.018849206
19	ShittyVeganFoodPorn	0.014281581
32	VeganBaking	0.009259259
24	vegan	0.004793267
8	dankmemes	0.000000000
12	FoodPorn	0.000000000
22	thatveganteachersucks	0.000000000
25	Vegan__Sensei	0.000000000
41	VeganDiet	0.000000000
49	vegangaming	0.000000000
50	vegangifrecipes	0.000000000
51	vegangranolamomsnark	0.000000000
53	veganinjapan	0.000000000
54	Veganinspire	0.000000000
55	VeganIreland	0.000000000
57	veganita	0.000000000
59	Veganivore	0.000000000
60	veganketo	0.000000000
62	veganmealprep	0.000000000
64	veganmexican	0.000000000
65	VeganNL	0.000000000

```
66        vegannutrition  0.000000000
68             veganpets  0.000000000
70            VeganRamen  0.000000000
81         Vegetarianism  0.000000000
4              AntiVegan -0.004256854
9           DebateAVegan -0.004259259
37       vegancirclejerk -0.005874279
63            veganmemes -0.008333333
11              exvegans -0.009951883
23      unpopularopinion -0.012301587
18        shittyfoodporn -0.019230769
40               VeganDE -0.032339831
73          veganscience -0.042016807
31               VeganAT -0.051282051
30          veganarchism -0.051893939
38   vegancirclejerkchat -0.058333333
6              AskVegans -0.066666667
48 VeganForCircleJerkers -0.069135802
33            VeganBeauty -0.071329365
56              Veganism -0.074652778
3          AmItheAsshole -0.078333333
15                 Jokes -0.166666667
39          vegancooking -0.222222222
75  VeganTeacherHateClub -0.333333333
```

Based on sentimentGI, subreddits having highest average of sentiment are "vegancheese" and "VeganIndia" with the average score of 0.333333333. Subreddits having lowest average of sentiment are "vegangranolamomsnark" and "VeganTeacherHateClub" with score of -0.333333333.

However, LC sentiment shows a different result. Subreddit having highest average of sentiment is "VeganWeightLoss" with the average score of 0.25. Subreddit having lowest average of sentiment is "VeganTeacherHateClub" with score of -0.333333333.

Question 3: What kind of posts received more comments?

```
subset_all_posts_data <- all_posts_data[, c(5, 6, 7, 8)]
sort_subset_all_posts_data <- subset_all_posts_data[
  order(subset_all_posts_data$comments, decreasing = TRUE),
  ]
head(sort_subset_all_posts_data[, c("subreddit", "comments")], 10)
```

```
         subreddit comments
1218 AmItheAsshole     2764
1217 AmItheAsshole     1644
713  AmItheAsshole     1279
182          vegan     1046
165          vegan     1037
997          vegan      872
1003         vegan      869
202          vegan      630
193          vegan      538
1219 AmItheAsshole      535
```

Top 10 posts receiving the most comments were posted in either `r/AmItheAsshole` or `r/vegan`, in which those posted in `r/AmItheAsshole` are mainly about conflicts between vegans and their non-vegan families or friends and one from `r/vegan` has a similar topic. It is sensible that these posts are more popular than others since they were published in two large subreddits and involve interpersonal issues that may raise empathy.

Question 4: Do the comments of a text post follow similar sentiment as the main post?

```r
# sentiment analysis
sentiments.comments <- analyzeSentiment(iconv(as.character(data.comments$comment),
                                              to = "UTF-8"))
head(sentiments.comments)
```

```
  WordCount SentimentGI NegativityGI PositivityGI SentimentHE NegativityHE
1        33 -0.03030303   0.03030303            0           0            0
2        17 -0.05882353   0.05882353            0           0            0
3        13  0.00000000   0.00000000            0           0            0
4        26  0.00000000   0.00000000            0           0            0
5        28 -0.07142857   0.07142857            0           0            0
6        19 -0.05263158   0.05263158            0           0            0
  PositivityHE SentimentLM NegativityLM PositivityLM RatioUncertaintyLM
1            0           0            0            0                  0
2            0           0            0            0                  0
3            0           0            0            0                  0
4            0           0            0            0                  0
5            0           0            0            0                  0
6            0           0            0            0                  0
  SentimentQDAP NegativityQDAP PositivityQDAP
1    0.03030303     0.00000000     0.03030303
2   -0.05882353     0.05882353     0.00000000
3    0.00000000     0.00000000     0.00000000
4    0.00000000     0.00000000     0.00000000
5    0.00000000     0.03571429     0.03571429
6   -0.05263158     0.05263158     0.00000000
```

```r
# calculate LC sentiments
tokenized_comments <- tokens_lookup(tokens(data.comments$comment),
                                    dictionary = data_dictionary_LSD2015,
                                    exclusive = FALSE)
sentiments.comments$LCpos <- sapply(tokenized_comments, function(x) {
  sum(x == "POSITIVE") - sum(x == "NEG_POSITIVE") + sum(x == "NEG_NEGATIVE")
  })
sentiments.comments$LCneg <- sapply(tokenized_comments, function(x) {
  sum(x == "NEGATIVE") - sum(x == "NEG_NEGATIVE") + sum(x == "NEG_POSITIVE")
  })
sentiments.comments$LC <- (sentiments.comments$LCpos - sentiments.comments$LCneg) /
  sentiments.comments$WordCount
```

```r
# merge data.comments and sentiments.comments
all_posts_comments <- cbind(data.comments, sentiments.comments)
```

```r
# merge data posts and comments
subset.posts <- all_posts_data[, c("date_utc","timestamp", "title", "text",
                                   "subreddit", "comments", "url", "WordCount",
                                   "SentimentGI", "LC")]

subset.comments <- all_posts_comments[, c("url", "date", "timestamp", "comment",
                                          "WordCount", "SentimentGI", "LC"),]

data.merge.posts.comments <- merge(x = subset.posts, y = subset.comments,
                                   by.x = c("url", "date_utc"), by.y = c("url", "date"),
                                   all.x = TRUE)

# clean dataset from missing sentiment and -inf value
complete.data.merge.posts.comments <- data.merge.posts.comments[
  complete.cases(data.merge.posts.comments[ , c("LC.x", "LC.y")]),
  ]

complete.data.merge.posts.comments2 <- subset(complete.data.merge.posts.comments,
                                              LC.y >= -1 & LC.y <= 1)
```

```r
# check relationship of GI sentiment from posts and comments
cor.test(complete.data.merge.posts.comments2$SentimentGI.x,
         complete.data.merge.posts.comments2$SentimentGI.y,
         method = "pearson")
```

```
    Pearson's product-moment correlation

data:  complete.data.merge.posts.comments2$SentimentGI.x and complete.data.merge.posts.comments2$Sentim
t = 5.7179, df = 8762, p-value = 1.114e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.04008590 0.08180366
sample estimates:
       cor
0.06097141
```

There is a significant correlation of sentimentGI between posts and comments, with significance level of 95%. The correlation score is 0.06 (weak and positive correlation).

```r
# check relationship of LC sentiment from posts and comments
cor.test(complete.data.merge.posts.comments2$LC.x,
         complete.data.merge.posts.comments2$LC.y,
         method = "pearson")
```

```
    Pearson's product-moment correlation

data:  complete.data.merge.posts.comments2$LC.x and complete.data.merge.posts.comments2$LC.y
t = 8.7682, df = 8762, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
```

```
 0.07246862 0.11397791
sample estimates:
       cor
0.09326379
```

There is a significant relationship of LC sentiments between posts and comments, with significance level of 95%. The correlation score is 0.09 (weak and positive correlation).

SentimentGI and sentimentLC showed a similar results.