Final Project - Data Wrangling

Swarup K Rakshit

04/26/2021

Final Project Description

Project is about a recent incident that happened around Game Stop Short Squeeze. Idea is to extract data from wikipedia (using the rvest package), twitter (using the rtweet package), reddit's wallstreetbets group data (using RedditExtractoR package), wrangle datasets and make it ready for the next phase of analysis. Eventually these data will be utilized to perform sentiment analysis, quantitative analysis and topic modelling using various techniques available in the field of natural language processing.

As per the review paper by Maurizio Naldi, 2019 around sentiment analysis there are 4 such packages by which sentiment analysis can be performed using R language,

1. Syuzhet;
2. Rsentiment;
3. SentimentR;
4. SentimentAnalysis.

First part of this project is to perform sentiment analysis on Twitter and Reddit's data extracted in the previous phase and leverage above packages to perform this analysis. Each package covers various aspects of sentiment analysis. Significant features of each package will be explored to provide meaningful insight of sentiment analysis on game stop data. Second part of this project is to perform Quantitative analysis using the quanteda package. Third part of this project is to perform Topic modelling using Latent Dirichlet allocation (LDA), Bag of Words, TF-IDF, Word2Vec etc. methodologies.

References:-

1. Game Stop Short Squeeze.
2. Rvest package.
3. Rtweet package.
4. RedditExtractorR package.
5. A review of sentiment computation methods with R packages by Maurizio Naldi
6. Syuzhet package.
7. Rsentiment package.
8. SentimentR package.
9. SentimentAnalysis package.
10. Quanteda package.
11. Topicmodels package.

Install necessary packages

```
#install.packages("rvest")
#install.packages("tidyverse")
#install.packages("stringr")
#install.packages("sjmisc")
#install.packages("lubridate")
#install.packages("RedditExtractoR")
#install.packages("tm")
#install.packages("syuzhet")
#install.packages("pander")
#install.packages("rlist")
#install.packages("sentimentr")
#install.packages("magrittr")
#install.packages("stringi")
#install.packages("pacman")
#install.packages("textcorpus")
#install.packages("textshape")
#install.packages("textreadr")
#install.packages("textclean")
#install.packages("numform")
#install.packages("xml2")
#install.packages("tidytext")
#install.packages("purrr")
#install.packages(rtweet")
#install.packages(RColorBrewer")
#install.packages(wordcloud")
#install.packages("data.table")
```

Load necessary packages

```
library(rvest)
library(tidyverse)
library(stringr)
library(sjmisc)
library(lubridate)
library(RedditExtractoR)
library(tm)
library(syuzhet)
library(pander)
library(rlist)
library(sentimentr)
library(magrittr)
library(stringi)
library(pacman)
library(textcorpus)
library(textshape)
library(textreadr)
library(textclean)
library(numform)
library(xml2)
library(data.table)
library(tidytext)
library(purrr)
library(rtweet)
library(RColorBrewer)
library(wordcloud)
```

Setting Working Directory

```
setwd("~/Documents/MSDS - Rutgers/Spring-2021/16-954-597-01-DATA-WRANGLING/final-proj
ect/msds-data-wragling-project")
```

Extracting Game Stop Short Squenze data from Wikipedia page.

```
url <- "https://en.wikipedia.org/wiki/GameStop_short_squeeze"
html <- read_html(url)

# read all html tables in a page
html_table_data <- html %>% html_nodes("table") %>% html_table(fill = TRUE)

# stock prices
other_stock_data <- html_table_data[[3]]
```

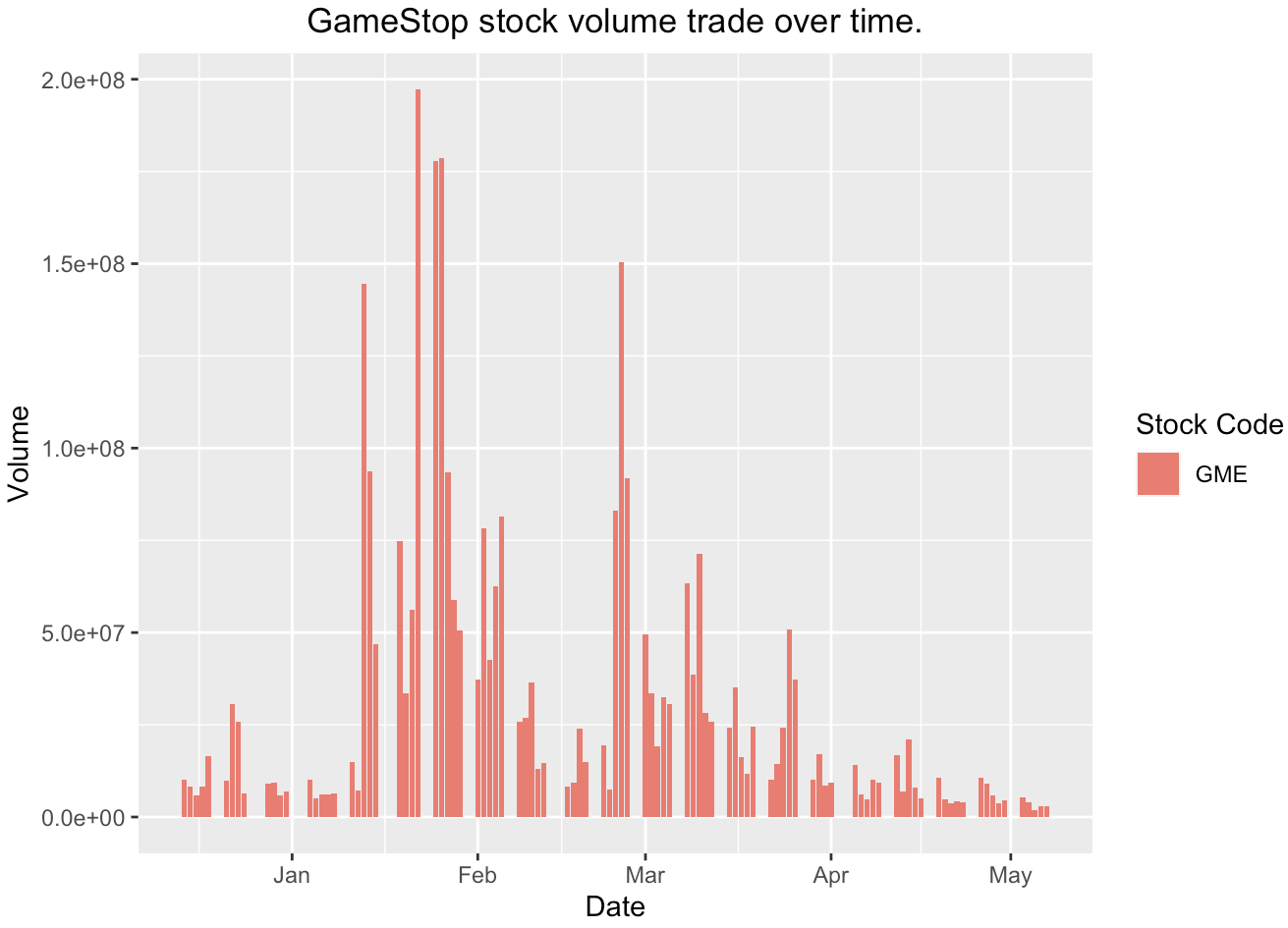
Functions used to extract data from Yahoo finance page and pre-process data.

```
getImpactedStocksData = function(impacted_stocks_df) {

  stock_names <- pull(impacted_stocks_df, `Security (symbol)`)
  #print(stock_names)

  stock_codes <- stock_names %>%
                 str_extract("\\(+([A-Z]{2,4})\\)+") %>%
                 str_extract("[A-Z]{2,4}")
  #print(stock_codes)

  impacted_stock_price_df <- data.frame()

  for ( stock_cd in stock_codes) {
    #print(stock_cd)
    stock_price_df <- getHistoricalStockPriceByStockCode(stock_cd)
    impacted_stock_price_df <- rbind(impacted_stock_price_df, stock_price_df)
  }

  impacted_stock_price_df
}

getHistoricalStockPriceByStockCode = function(stock_code) {
  url <- paste0("https://finance.yahoo.com/quote/", stock_code, "/history/")
  #print(url)
  stock_price_df_list <- url %>%
                      read_html(url) %>%
                      html_nodes("table") %>%
                      html_table(fill = TRUE)

  stock_price_df <- stock_price_df_list[[1]]

  # removed NA records
  stock_price_df <- head(stock_price_df, -1)

  stock_price_df$Date <- mdy(stock_price_df$Date)
  stock_price_df$Open <-as.numeric(stock_price_df$Open)
  stock_price_df$High <-as.numeric(stock_price_df$High)
  stock_price_df$Low <-as.numeric(stock_price_df$Low)
  stock_price_df$`Close*` <-as.numeric(stock_price_df$`Close*`)
  stock_price_df$`Adj Close**` <-as.numeric(stock_price_df$`Adj Close**`)
  stock_price_df$`Volume` <-as.numeric(gsub(",", "", stock_price_df$Volume))
  stock_price_df$`Stock Code` <- stock_code

  stock_price_df
}
```

Plot GameStop price/volume data over time using ggplot library.

```
gme_stock_df <- getHistoricalStockPriceByStockCode("GME")

ggplot(data = gme_stock_df) +
  geom_line(mapping = aes(x = Date, y = `Close*`, color = `Stock Code`)) +
  labs(x = "Date", y = "Price") +
  ggtitle("GameStop stock price over time.") +
  theme(plot.title = element_text(size = 13)) + theme(plot.title = element_text(hjust
= 0.5))
```


```
ggplot(data = gme_stock_df, mapping = aes(x = Date)) +
  geom_col(mapping = aes(y = `Volume`, fill = `Stock Code`)) +
  labs(x = "Date", y = "Volume") +
  ggtitle("GameStop stock volume trade over time.") +
  theme(plot.title = element_text(size = 13)) + theme(plot.title = element_text(hjust
= 0.5))
```



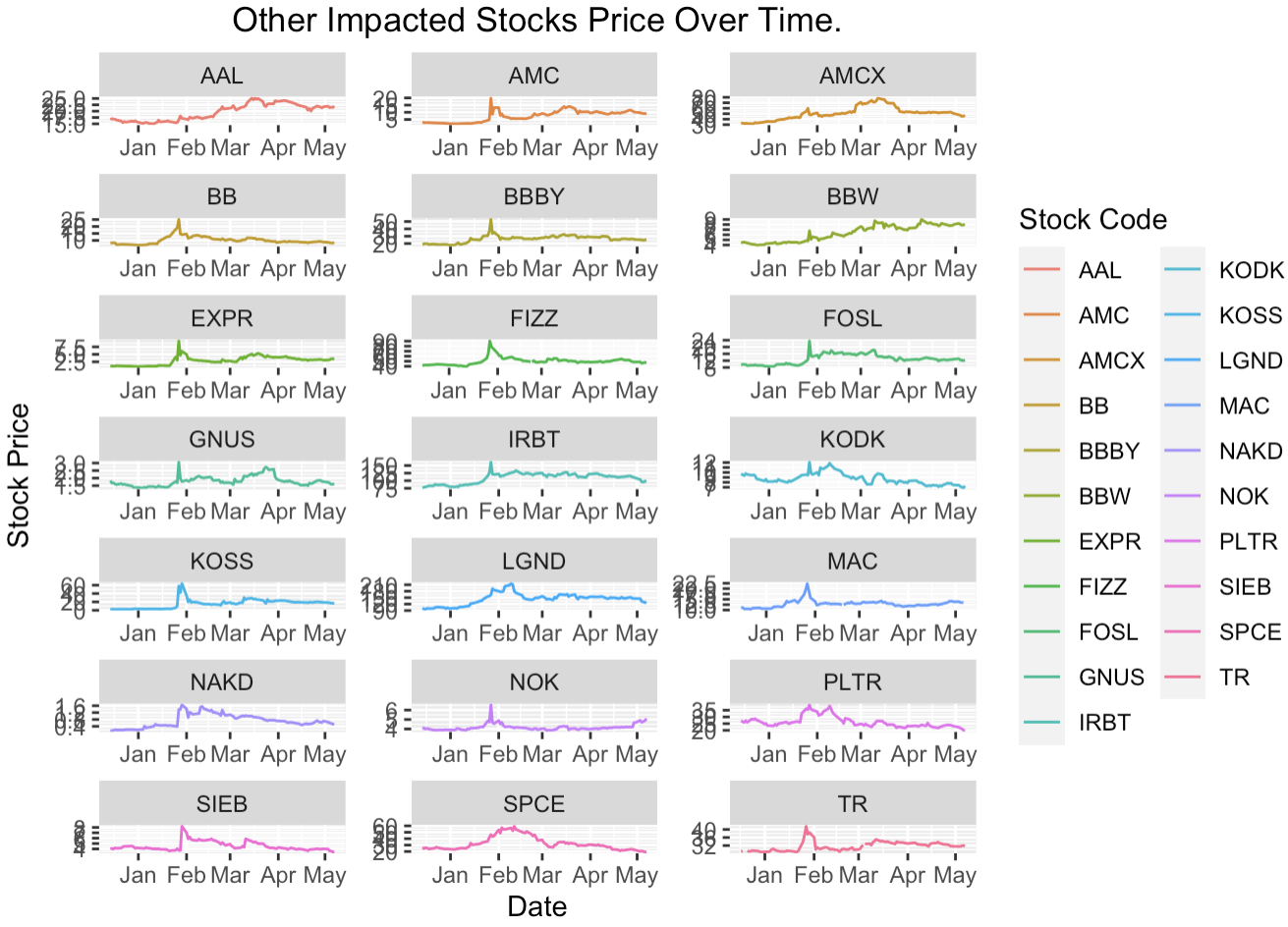
Let's see other impacted stocks along with GameStop.

```
other_stock_data %>%
  select(-`Ref.`)
```

```
## # A tibble: 22 x 4
##    `Security (symbol)`                       `Price high[a]` `Jan 22` `% chg.`
##    <chr>                                              <dbl>    <dbl> <chr>
##  1 AMC Entertainment Holdings, Inc. (AMC)              20.4     3.51 480.1%
##  2 AMC Networks Inc. (AMCX)                            59.8    49.4  21.2%
##  3 American Airlines Group Inc. (AAL)                  21.8    15.8  37.6%
##  4 BB Liquidating Inc. (OTC Pink: BLIAQ)                0.3     0.01 3000%
##  5 Bed Bath & Beyond Inc. (BBBY)                       53.9    30.2  78.4%
##  6 BlackBerry Limited (BB)                             28.8    14.0  104.9%
##  7 Build-A-Bear Workshop, Inc. (BBW)                    8.4     4.52 85.8%
##  8 Eastman Kodak Company (KODK)                        15.2     9.46 60.1%
##  9 Express, Inc. (EXPR)                                14.0     1.79 680.4%
## 10 Fossil Group, Inc. (FOSL)                           28.6     9.87 189.8%
## # … with 12 more rows
```

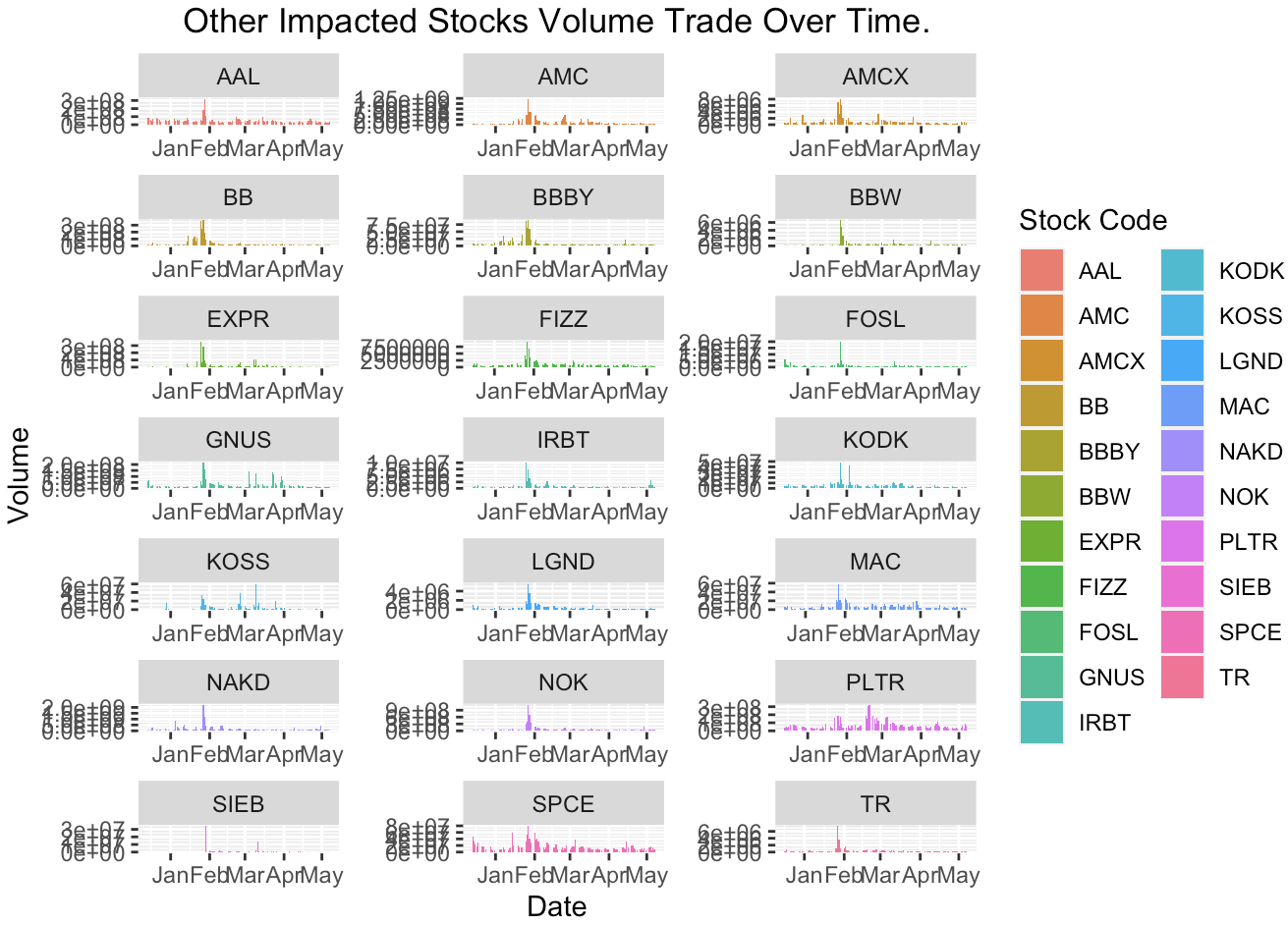
Below plot depicts how stock prices has changes during the course of this event.

```
impacted_stock_price_df <- getImpactedStocksData(other_stock_data)

ggplot(data = impacted_stock_price_df) +
  geom_line(mapping = aes(x = Date, y = `Close*`, color = `Stock Code`)) +
  labs(x = "Date", y = "Stock Price") +
  facet_wrap(~`Stock Code`, ncol = 3, scales = "free") +
  ggtitle("Other Impacted Stocks Price Over Time.") +
  theme(plot.title = element_text(size = 13)) + theme(plot.title = element_text(hjust
= 0.5))
```



Below plot represents how many stocks are being traded during the course of this event.

```
ggplot(data = impacted_stock_price_df) +
  geom_col(mapping = aes(x = Date, y = `Volume`, fill = `Stock Code`)) +
  labs(x = "Date", y = "Volume") +
  facet_wrap(~`Stock Code`, ncol = 3, scales = "free") +
  ggtitle("Other Impacted Stocks Volume Trade Over Time.") +
  theme(plot.title = element_text(size = 13)) + theme(plot.title = element_text(hjust
= 0.5))
```



Extracting wallstreetbets Reddit group data using RedditExtractorR package.

```
reddit_wallstreetbets <- get_reddit(subreddit = "wallstreetbets", page_threshold = 1,
sort_by = "relevance")
```

```
## Cannot connect to the website, skipping...
## Cannot connect to the website, skipping...
## Cannot connect to the website, skipping...
## Cannot connect to the website, skipping...
##
  |
  |                                                                      |   0%
  |
  |===                                                                   |   4%
  |
  |======                                                                |   8%
  |
  |=========                                                             |  12%
  |
  |============                                                          |  17%
  |
  |===============                                                       |  21%
  |
  |==================                                                    |  25%
  |
  |====================                                                  |  29%
  |
  |=======================                                               |  33%
  |
  |===========================                                           |  38%
  |
  |==============================                                        |  42%
  |
  |================================                                      |  46%
  |
  |===================================                                   |  50%
  |
  |=======================================                               |  54%
  |
  |==========================================                            |  58%
  |
  |==============================================                        |  62%
  |
  |================================================                      |  67%
  |
  |====================================================                  |  71%
  |
  |======================================================                |  75%
  |
  |==========================================================            |  79%
  |
  |==============================================================        |  83%
  |
  |================================================================      |  88%
  |
  |====================================================================  |  92%
```

```
  |
  |=================================================================    |  96%
  |
  |======================================================================| 100%
```

```
# sample reddit wallstreetbets data.
reddit_wallstreetbets %>% select(post_date, author, subreddit, comment) %>% head()
```

```
##   post_date    author      subreddit
## 1  07-05-21 WhichEdge wallstreetbets
## 2  07-05-21 WhichEdge wallstreetbets
## 3  07-05-21 WhichEdge wallstreetbets
## 4  07-05-21 WhichEdge wallstreetbets
## 5  07-05-21 WhichEdge wallstreetbets
## 6  07-05-21 WhichEdge wallstreetbets
##
comment
## 1 I'm not sure why anyone would want in on mortgage company stocks right now.   In
terest rates are slated to rise after being at lifetime lows for the last few years,
while everyone who wanted to refinance has done so already, and housing inventory is
at an all time low.  Mortgage demand pretty much has no where to go but down at this
point.
## 2
So you are saying I\031m screwed on my UWMC calls? I still have 2000 shares.
## 3
No. Hold those. UWMC is a far better company than RKT and the only people who don't r
ealize that are people who don't/have never owned a home.   \n\n\nPositions - 400 UWM
C, 300 LDI
## 4
P/e of 3 even if rates rise unless they lose two thirds of their revenue this valuati
on is unjustified
## 5
The whole casino bet, also I think a lot thought after the huge dip it would pick up
not further double down on that.\n\nI completely agree with your assessment though, i
t's refreshing to see some good DD here again finally.
## 6
Damn I'm just going to hold till I die then
```

```
#reddit_wallstreetbets_250 <- get_reddit(subreddit = "wallstreetbets", page_threshold
= 250, sort_by = "relevance", wait_time = 60)
#write_csv(x = reddit_wallstreetbets_250,
#          file = "/Users/swaruprakshit/Documents/MSDS - Rutgers/Spring-2021/16-954-5
97-01-DATA-WRANGLING/Final Project/final-project-submission/data/reddit_wallstreetbet
s_data_250.csv",
#          na = "NA")
```

Loading data from previously saved csv file. Let's take a look at Wallstreetbets

data.

```
# reading it from previously saved data.
reddit_wallstreetbets <- read.csv("data/reddit_wallstreetbets_data_150.csv", header =
TRUE)

reddit_wallstreetbets %>% select(post_date, author, subreddit, comment) %>% head()
```

```
##   post_date   author      subreddit
## 1  08-04-21 shdhkxjc wallstreetbets
## 2  08-04-21 shdhkxjc wallstreetbets
## 3  08-04-21 shdhkxjc wallstreetbets
## 4  08-04-21 shdhkxjc wallstreetbets
## 5  08-04-21 shdhkxjc wallstreetbets
## 6  08-04-21 shdhkxjc wallstreetbets
##
comment
## 1
What bells are you gonna hear in April when you got January calls, lol
## 2
Not a yolo, this exp next year not in 2 days
## 3 Why the hell are people expecting Disney to do 30% when it\031s currently 50% hi
gher than pre covid level? \n\nI\031ve seen some balance sheet and new openings but i
t\031s def not the mtk value in pricing of 260 per share
## 4
By the next year not totally impossible.
## 5                                                    If they start streaming movi
es direct to the Disney channel then it\031s very possible.\n\nI think the opening up
is mostly priced in so they will need to add new stuff.
## 6
Stream what, their biggest thing woke Star Wars?
```

RedditExtractorR package provides api's to created user network, let's explore and find out how reddit user's are connected?

```
#game_stop_urls <- reddit_urls(search_terms="gamestop", page_threshold = 1) # isolate
some URLs
#write_csv(x = game_stop_urls,
#          file = "/Users/swaruprakshit/Documents/MSDS - Rutgers/Spring-2021/16-954-5
97-01-DATA-WRANGLING/Final Project/final-project-submission/data/game_stop_urls.csv",
#          na = "NA")

# reading data from previously saved file.
game_stop_urls <- read_csv(file = "data/game_stop_urls.csv")
game_stop_df <- game_stop_urls %>% filter(num_comments==max(game_stop_urls$num_commen
ts)) %$% URL %>% reddit_content # get the contents of a small thread
```

```
##
  |
  |                                                                      |   0%
  |
  |======================================================================| 100%
```

```
game_stop_user_network <- game_stop_df %>% user_network(include_author=FALSE, agg=TRU
E) # extract the network
game_stop_user_network$plot # explore the plot
```

User Network

Extract data from Twitter using rtweet package. Twitter api has limitation of 18K records can be downloaded in every 15 min. In order to extract more data for analysis, apply delay. Also we need to make sure that every call gets unique tweet post. In achive that, used max_id attribute in twitter api. For a given extract, min of status_id represents oldest tweet in that extract which is used as max_id and be used as starting point of next extraction.

```
api_key <- "5HzcNSBdFTpgQmQanbNBdA2jL"
api_key_secret <- "4kLa5QZ04P56vIHedLztI1vZKjipwwC0xxGXrpWSOCRn8wU7vE"
access_token <- "1209907587073884160-vLJDQdjQl6NgrYi0gStwltMfOo5HH5"
access_token_secret <- "e3iwUh3xcZ0ZMvExou8NWUe0O2dvQSPm00ENo4vWTAkgU"
app_name <- "MSDS_FINAL_PROJECT_APP"

## authenticate via web browser
token <- create_token(
  app = app_name,
  consumer_key = api_key,
  consumer_secret = api_key_secret,
  access_token = access_token,
  access_secret = access_token_secret)

# delay function taken number of seconds
delay <- function(x) {
  p1 <- proc.time()
  Sys.sleep(x)
  proc.time() - p1 # The cpu usage should be negligible
}

# function is responsible for extract data from twitter.
twitter_data_extrctor <- function() {

  gamestop_short_squeeze_tweet_master_df <- data.frame()
  prev_max_id <- "0"
  for (i in 1:1) {
    gamestop_short_squeeze_tweet <- search_tweets( q = "#$gme OR #shortsqueeze OR #ga
mestopshortsqueeze OR #gmeshortsqueeze OR #thebigshortsqueeze OR #gamestop OR #gme",
                                                   n = 18000,
                                                   type = "mixed",
                                                   include_rts = TRUE,
                                                   geocode = NULL,
                                                   max_id = prev_max_id,
                                                   token = bearer_token(),
                                                   #retryonratelimit = TRUE,
                                                   lang = "en")

    prev_max_id <- as.character(min(gamestop_short_squeeze_tweet$status_id))
    gamestop_short_squeeze_tweet_master_df <- rbind(gamestop_short_squeeze_tweet_mast
er_df, gamestop_short_squeeze_tweet)
    # twitter api has rate limit to 18K records can be extracted in every 15 min.
    # delay(900)
  }
  return (gamestop_short_squeeze_tweet_master_df)
}

# twitter data frame
gamestop_short_squeeze_tweet_master_df <- twitter_data_extrctor()

# sample twitter data.
```

```
gamestop_short_squeeze_tweet_master_df %>%
  select(user_id, status_id, created_at, screen_name, text) %>%
  head()
```

```
## # A tibble: 6 x 5
##   user_id  status_id    created_at          screen_name text
##   <chr>    <chr>        <dttm>              <chr>       <chr>
## 1 1584450… 13903005880… 2021-05-06 13:41:05 BetterMark… "Today's @FSCDems 3rd h…
## 2 1584450… 13899633909… 2021-05-05 15:21:11 BetterMark… "The @FSCDems holds its…
## 3 1885094… 13907229600… 2021-05-07 17:39:26 jm_corba    "SEC Chair Gary Gensler…
## 4 1885094… 13903402143… 2021-05-06 16:18:33 jm_corba    "Glued to the third hea…
## 5 3367334… 13883654827… 2021-05-01 05:31:40 BTCTN       "Jerome Powell said tha…
## 6 2754259… 13908007770… 2021-05-07 22:48:39 edOfCHRIST  "Don't Want To Say It B…
```

```
# used to save large data into csv format.
#write_as_csv(x = gamestop_short_squeeze_tweet_master_df, file_name = "/Users/swarupr
akshit/Documents/MSDS - Rutgers/Spring-2021/16-954-597-01-DATA-WRANGLING/Final Projec
t/final-project-submission/data/gamestop_short_squeeze_tweet_master_100k.csv", prepen
d_ids = TRUE, fileEncoding = "UTF-8")
```

Loading data from previously saved csv file. Let's take a look at gamestop stop squeeze twitter's tweet data.

```
gamestop_short_squeeze_tweet <- read_twitter_csv(file = "data/gamestop_short_squeeze_
tweet_master_100k.csv", unflatten = FALSE)

gamestop_short_squeeze_tweet %>%
  select(user_id, status_id, created_at, screen_name, text) %>%
  head()
```

```
## # A tibble: 6 x 5
##   user_id     status_id     created_at   screen_name text
##   <chr>       <chr>         <chr>        <chr>       <chr>
## 1 118639757   138993398515… 2021-05-05 … kokid951    "I Never Bought any $GME b…
## 2 158445020   138996339090… 2021-05-05 … BetterMark… "The @FSCDems holds its 3r…
## 3 3367334171  138836548277… 2021-05-01 … BTCTN       "Jerome Powell said that t…
## 4 1354472301… 139027016095… 2021-05-06 … gmerockets… "This is a really good bre…
## 5 1354472301… 138958153617… 2021-05-04 … gmerockets… "$GME DFV on May 4th\n\n#M…
## 6 1354472301… 138929550337… 2021-05-03 … gmerockets… "Part III of the $GME Cong…
```

Let's clean reddit user's comments and twitter's tweet for further analysis.

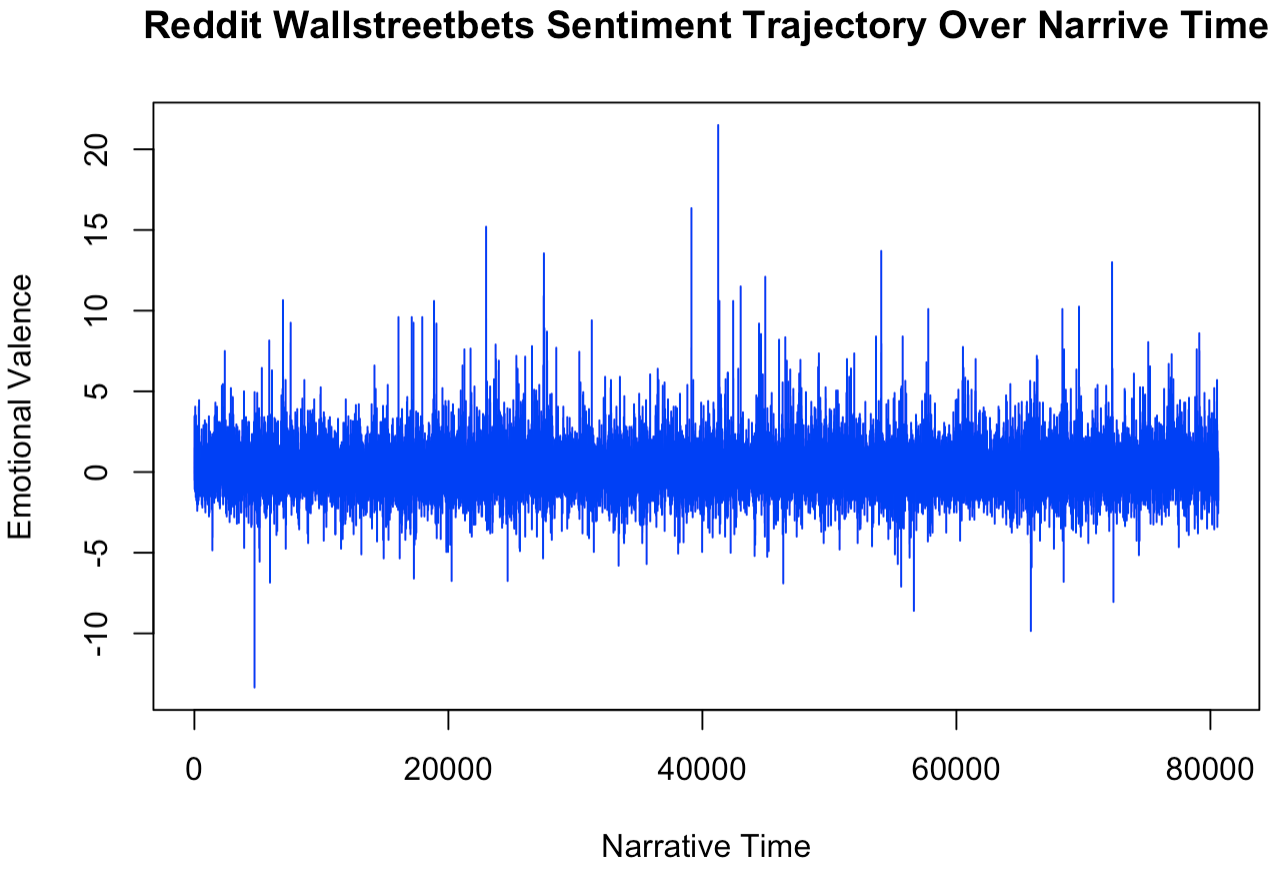
```
# Function for data cleaning
f_gsub_clean_data <- function (data) {

  # remove at people
  clean_data = gsub('@\\w+', '', data)
  # remove punctuation
  clean_data = gsub('[[:punct:]]', '', clean_data)
  # remove numbers
  clean_data = gsub('[[:digit:]]', '', clean_data)
  # remove html links
  clean_data = gsub('http\\w+', '', clean_data)
  # remove unnecessary spaces
  clean_data = gsub('[ \t]{2,}', '', clean_data)
  clean_data = gsub('^\\s+|\\s+$', '', clean_data)
  # remove emojis or special characters
  clean_data = gsub('<.*>', '', enc2native(clean_data))
  # to lowercase
  clean_data = tolower(clean_data)
  # change character encoding
  clean_data = iconv(clean_data, to="utf-8-mac")

  clean_data
}

reddit_wallstreetbets_comments_clean <- f_gsub_clean_data(reddit_wallstreetbets$comme
nt)
gamestop_short_squeeze_tweet_clean <- f_gsub_clean_data(gamestop_short_squeeze_twee
t$text)
```

Let's try to leverage R package Syuzhet to do sentiment analysis.

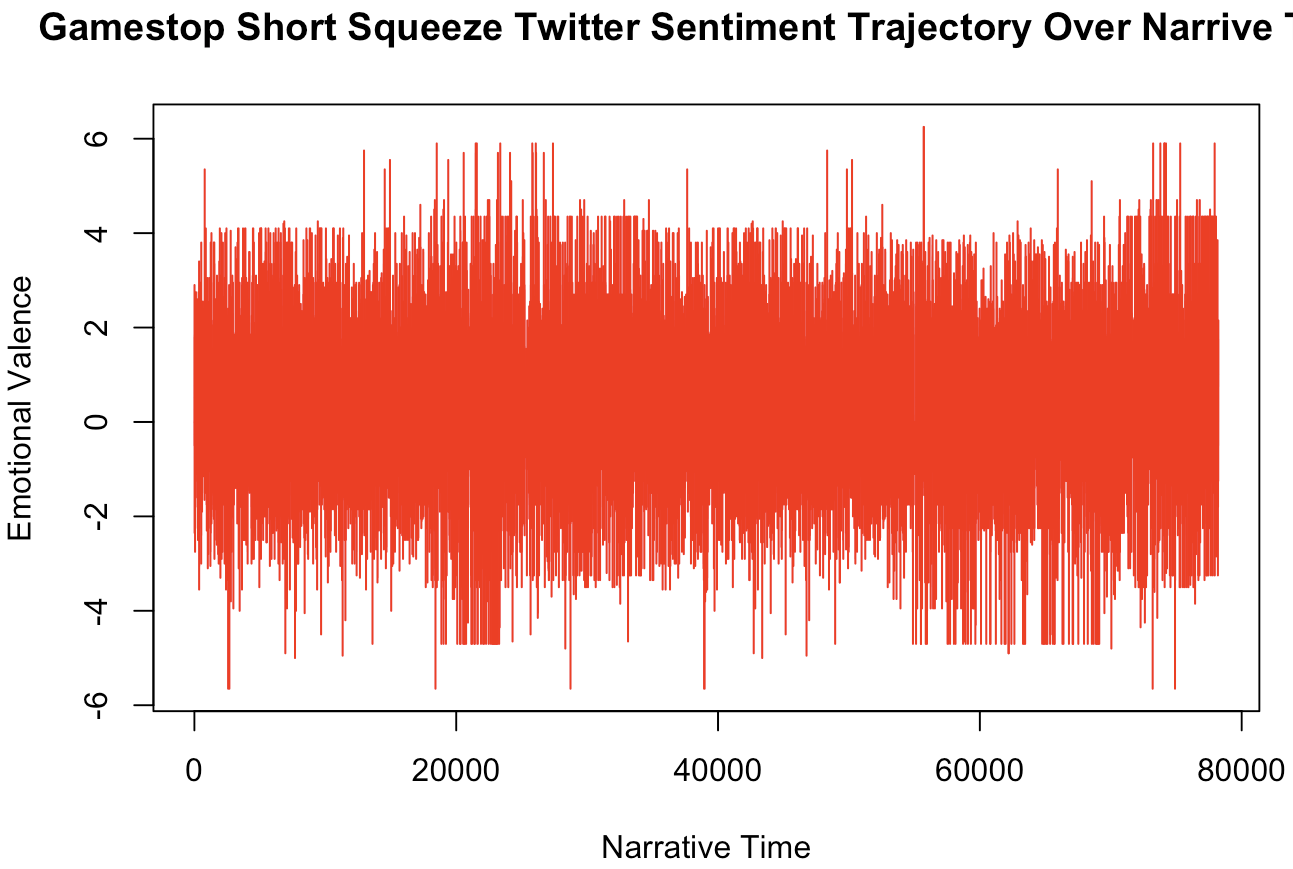
The package comes with four sentiment dictionaries and provides a method for accessing the robust, but computationally expensive, sentiment extraction tool developed in the NLP group at Stanford. Use of this later method requires that you have already installed the coreNLP package (see http://nlp.stanford.edu/software/corenlp.shtml (http://nlp.stanford.edu/software/corenlp.shtml)).

Let's explore how sentiment trajectory looks like over narrative time.

```
reddit_wallstreetbets_comments_clean_sentiment <- syuzhet::get_sentiment(reddit_walls
treetbets_comments_clean)
gamestop_short_squeeze_tweet_clean_sentiment <- syuzhet::get_sentiment(gamestop_short
_squeeze_tweet_clean)

plot(
  reddit_wallstreetbets_comments_clean_sentiment,
  type = "l",
  main = "Reddit Wallstreetbets Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "blue"
)
```

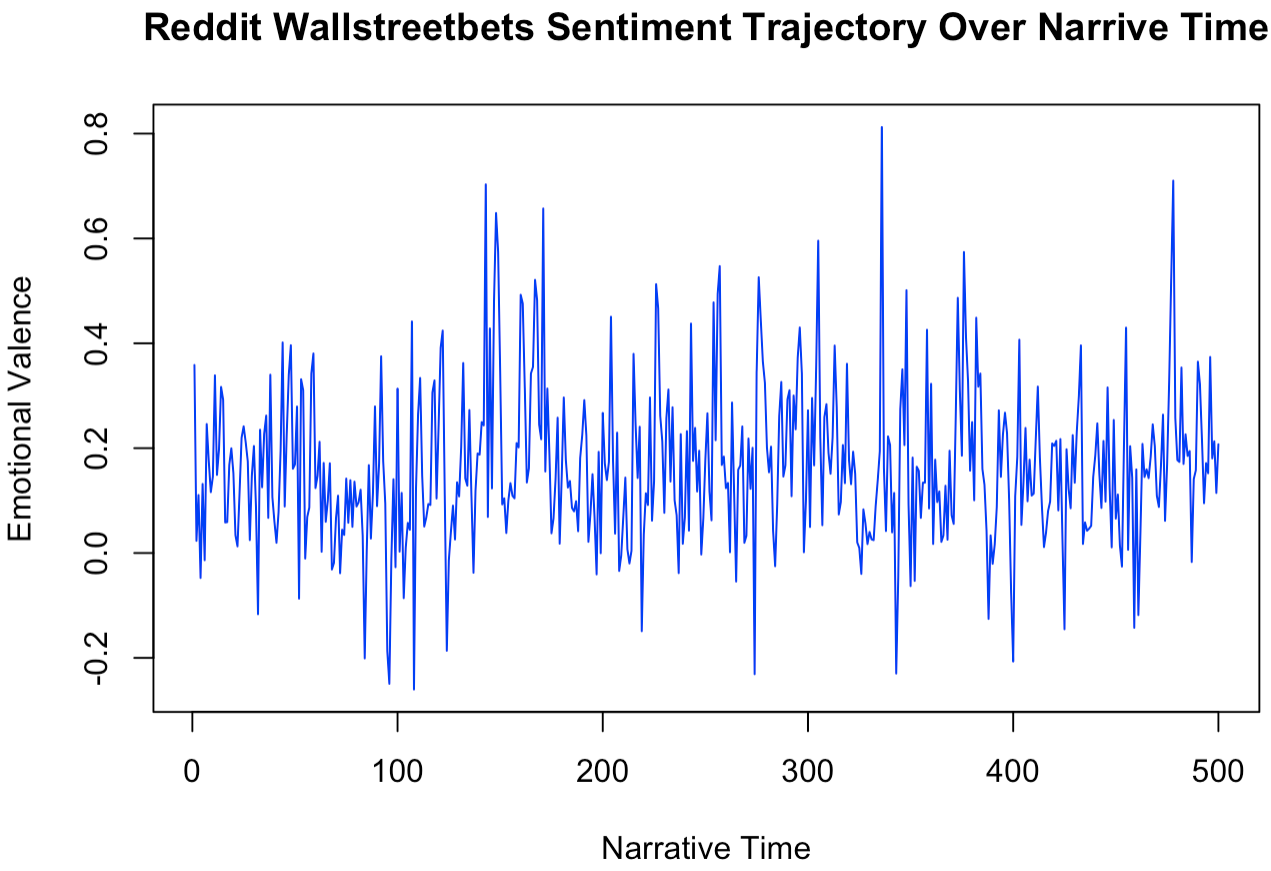



```
plot(
  gamestop_short_squeeze_tweet_clean_sentiment,
  type = "l",
  main = "Gamestop Short Squeeze Twitter Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "red"
)
```

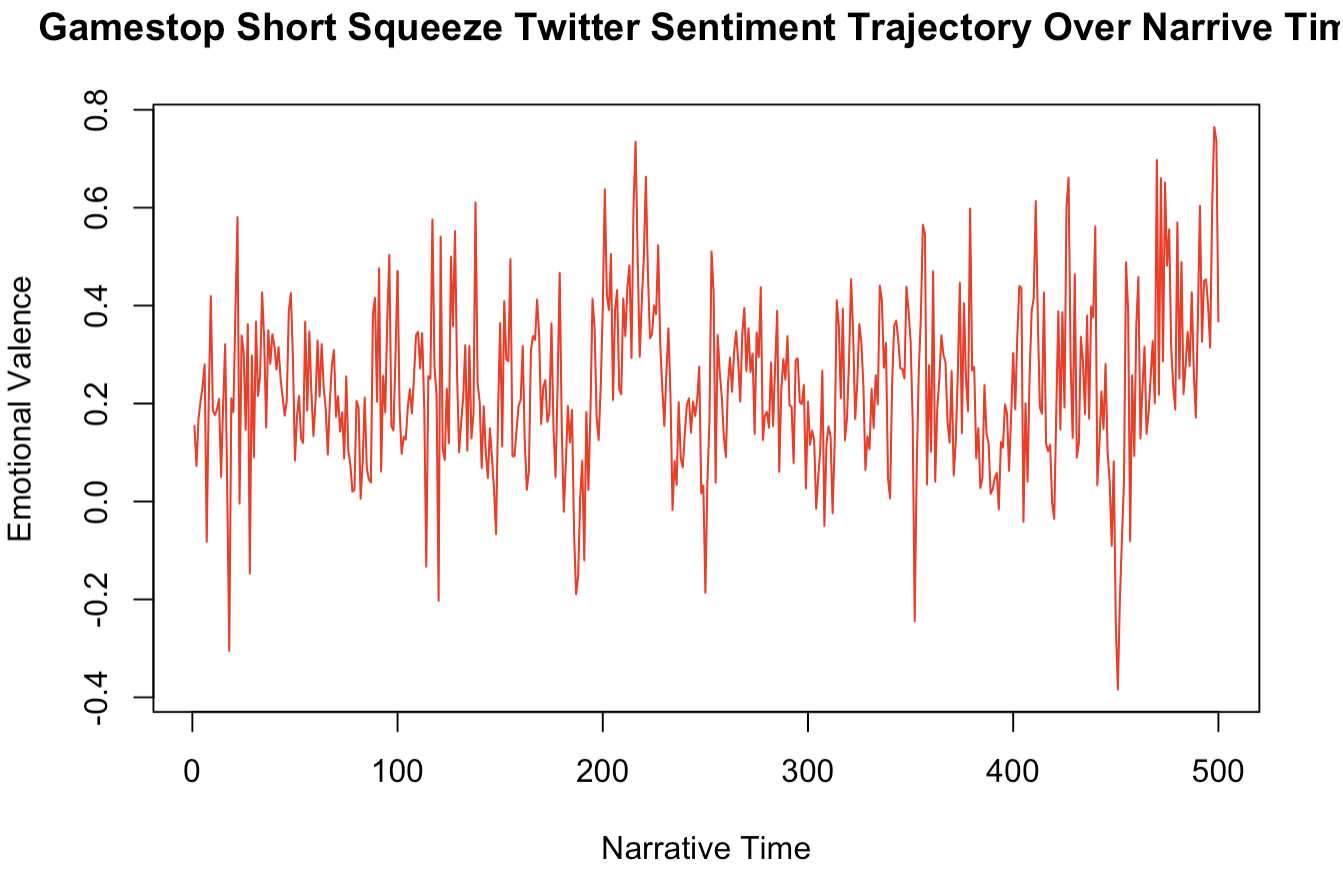


As we see from above plot that very difficult to interpret the polarity of sentiment. syuzhet packages provides another function get_percentage_values(...) which divides text into equal number of chunks and then calculates the mean sentiment valence for each. In this plot used bin = 500 represents chunk size.

```
reddit_wallstreetbets_comments_clean_vector <- syuzhet::get_sentiment(reddit_wallstre
etbets_comments_clean, method = "syuzhet")
gamestop_short_squeeze_tweet_clean_vector <- syuzhet::get_sentiment(gamestop_short_sq
ueeze_tweet_clean, method = "syuzhet")

reddit_wallstreetbets_comments_clean_percentage_sentiment <-  syuzhet::get_percentage
_values(reddit_wallstreetbets_comments_clean_vector, bins = 500)

gamestop_short_squeeze_tweet_clean_percentage_sentiment <-  syuzhet::get_percentage_v
alues(gamestop_short_squeeze_tweet_clean_vector, bins = 500)


plot(
  reddit_wallstreetbets_comments_clean_percentage_sentiment,
  type = "l",
  main = "Reddit Wallstreetbets Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "blue"
)
```

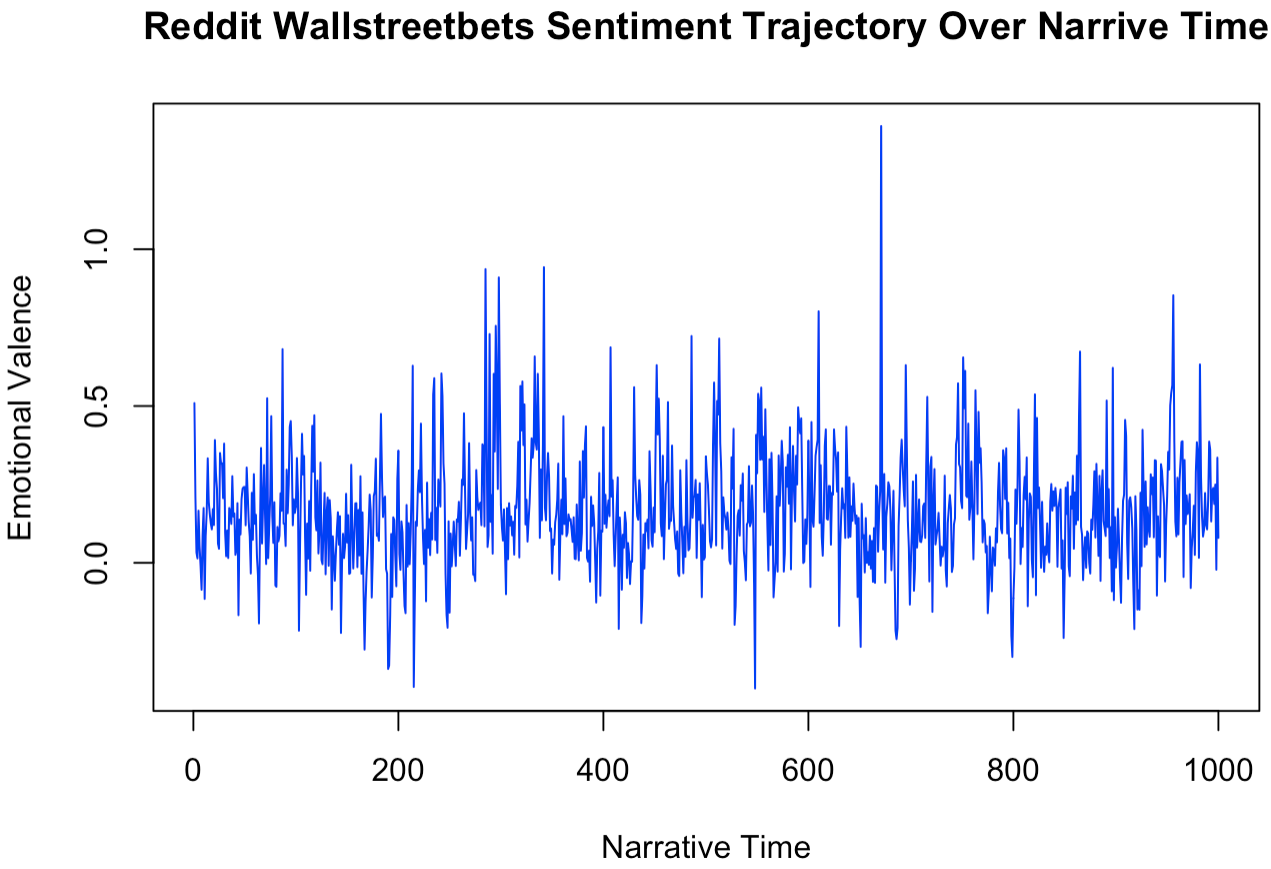


```
plot(
  gamestop_short_squeeze_tweet_clean_percentage_sentiment,
  type = "l",
  main = "Gamestop Short Squeeze Twitter Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "red"
)
```

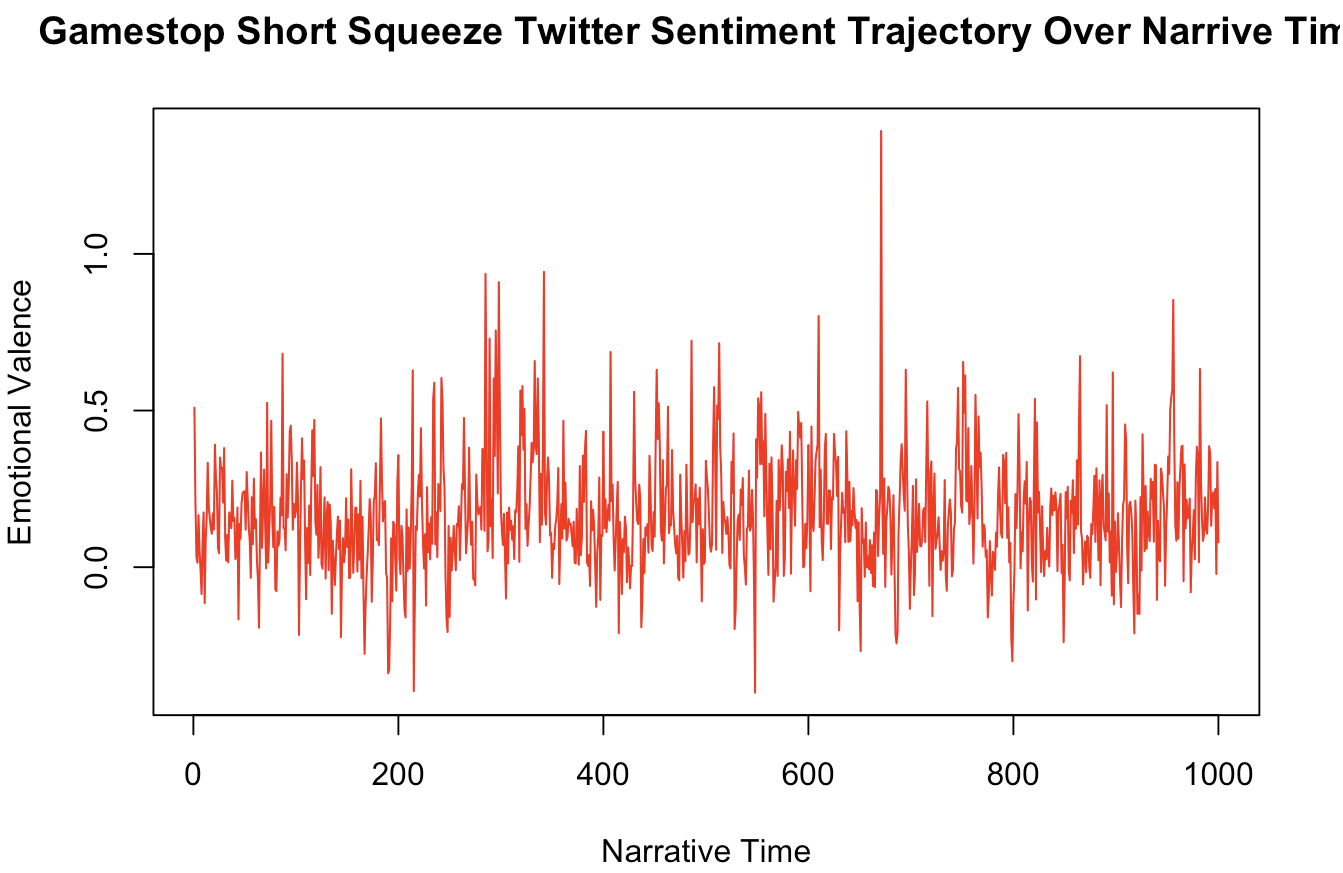


As per plot using chunk size as 500, it's hard to interpretate sentiment tracjectory. Let's try chunk value as 1000 and see if plot is getting any better?

```
reddit_wallstreetbets_comments_clean_percentage_sentiment <- syuzhet::get_percentage_
values(reddit_wallstreetbets_comments_clean_vector, bins = 1000)

gamestop_short_squeeze_tweet_clean_percentage_sentiment <- syuzhet::get_percentage_va
lues(gamestop_short_squeeze_tweet_clean_vector, bins = 1000)

plot(
  reddit_wallstreetbets_comments_clean_percentage_sentiment,
  type = "l",
  main = "Reddit Wallstreetbets Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "blue"
)
```



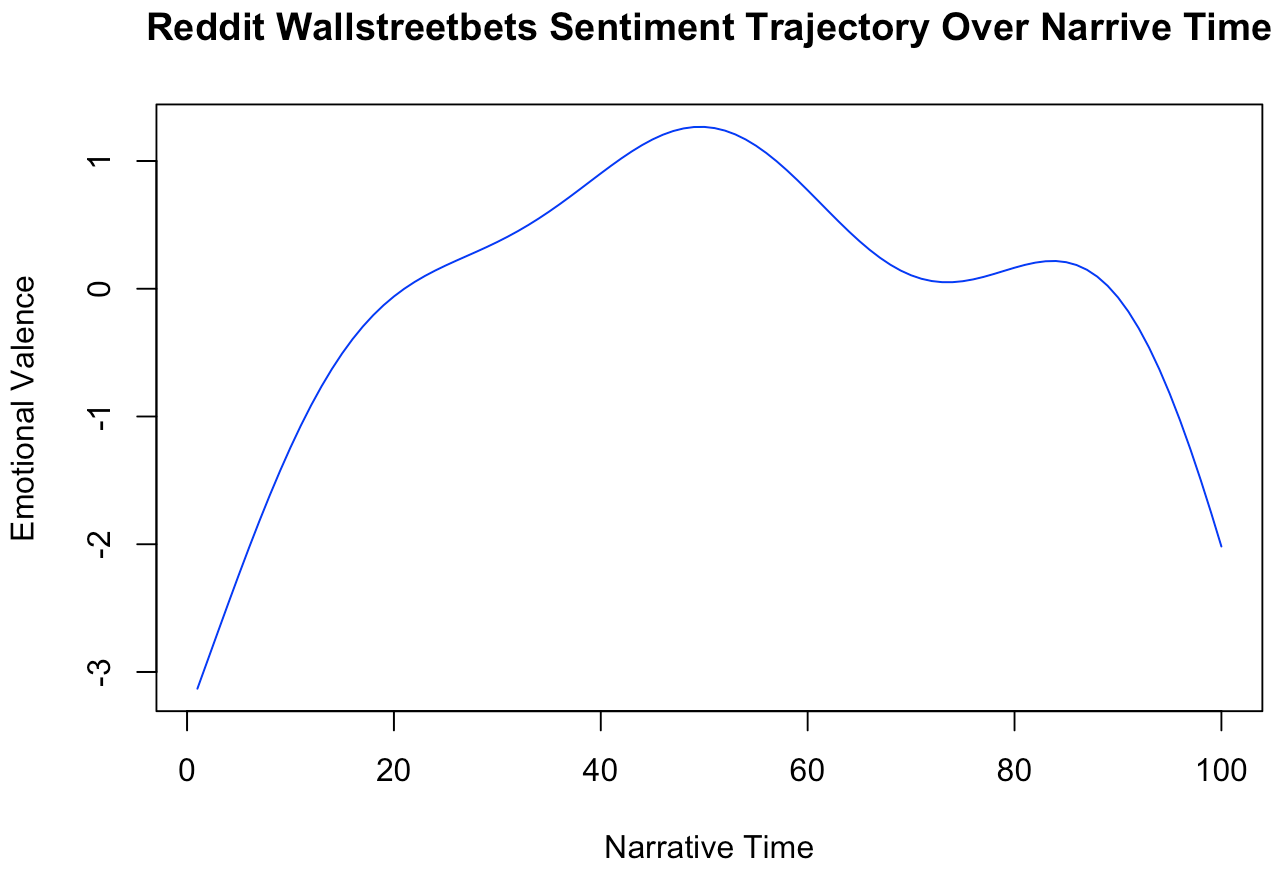

```
plot(
  reddit_wallstreetbets_comments_clean_percentage_sentiment,
  type = "l",
  main = "Gamestop Short Squeeze Twitter Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "red"
)
```



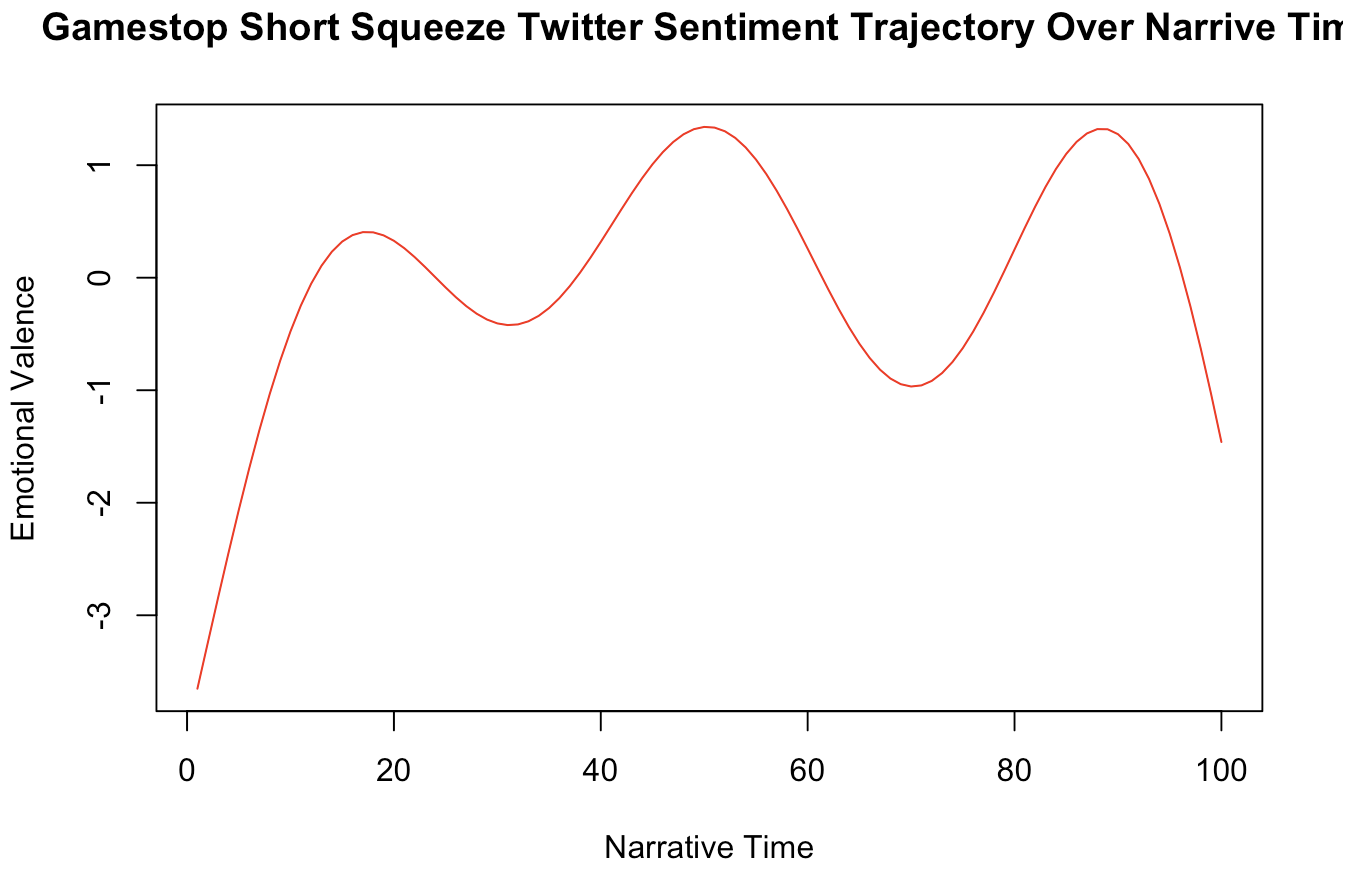
Explanation:- Unfortunately, percentage value approach does not explain emotional valance trajectory due to following reason, 1. Combining larger chunk (i.e. 500 or 1000 sentence) contains wide range of emotion values than 100 sentence chunk. . Indeed, the means of longer passages tend to converge toward 0.
2. In addition to that, emotion valance changes corpus to corpus. Grouping corpus might be get the sentiment trajectory. Syuzhet package provides two alternatives to percentage-based comparison using either the Fourier or Discrete Cosine Transformations in combination with a low pass filter.

Emotional Valance analysis using Fourior Transformation technique (i.e.get_transformed_values(...))

```
reddit_wallstreetbets_comments_clean_vector <- syuzhet::get_sentiment(reddit_wallstre
etbets_comments_clean, method = "syuzhet")

reddit_wallstreetbets_comments_clean_vector_ft_values <- syuzhet::get_transformed_val
ues(reddit_wallstreetbets_comments_clean_vector,
                                                                    low_pass_size = 3,
                                                                    x_reverse_len = 10
0,
                                                                    padding_factor =
2,
                                                                    scale_vals = TRUE,
                                                                    scale_range = FALS
E
                                                                   )

gamestop_short_squeeze_tweet_clean_vector_ft_values <- syuzhet::get_transformed_value
s(gamestop_short_squeeze_tweet_clean_vector,
                                                                    low_pass_size = 3,
                                                                    x_reverse_len = 10
0,
                                                                    padding_factor =
2,
                                                                    scale_vals = TRUE,
                                                                    scale_range = FALS
E
                                                                   )


plot(
  reddit_wallstreetbets_comments_clean_vector_ft_values,
  type = "l",
  main ="Reddit Wallstreetbets Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "blue"
)
```

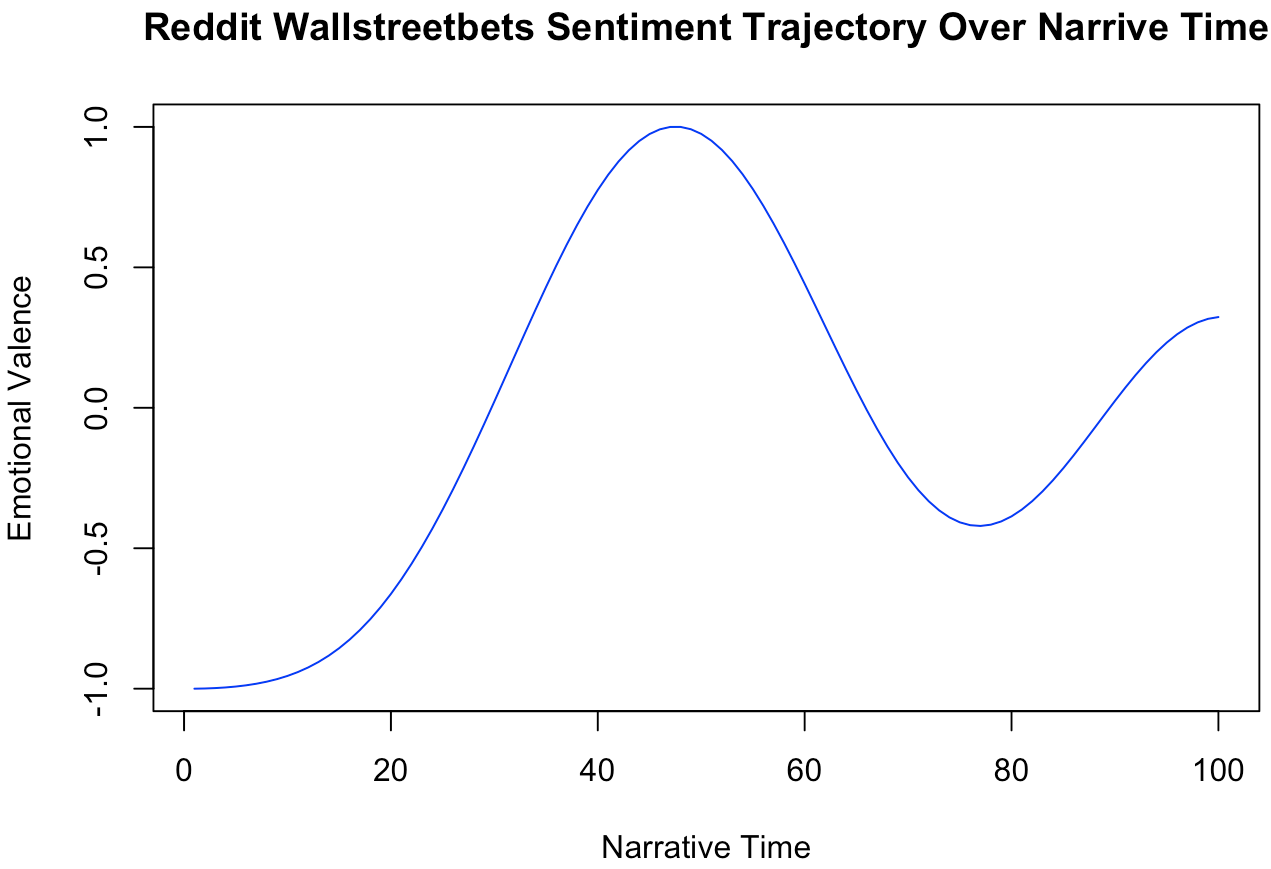



```
plot(
  gamestop_short_squeeze_tweet_clean_vector_ft_values,
  type = "l",
  main ="Gamestop Short Squeeze Twitter Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "red"
)
```

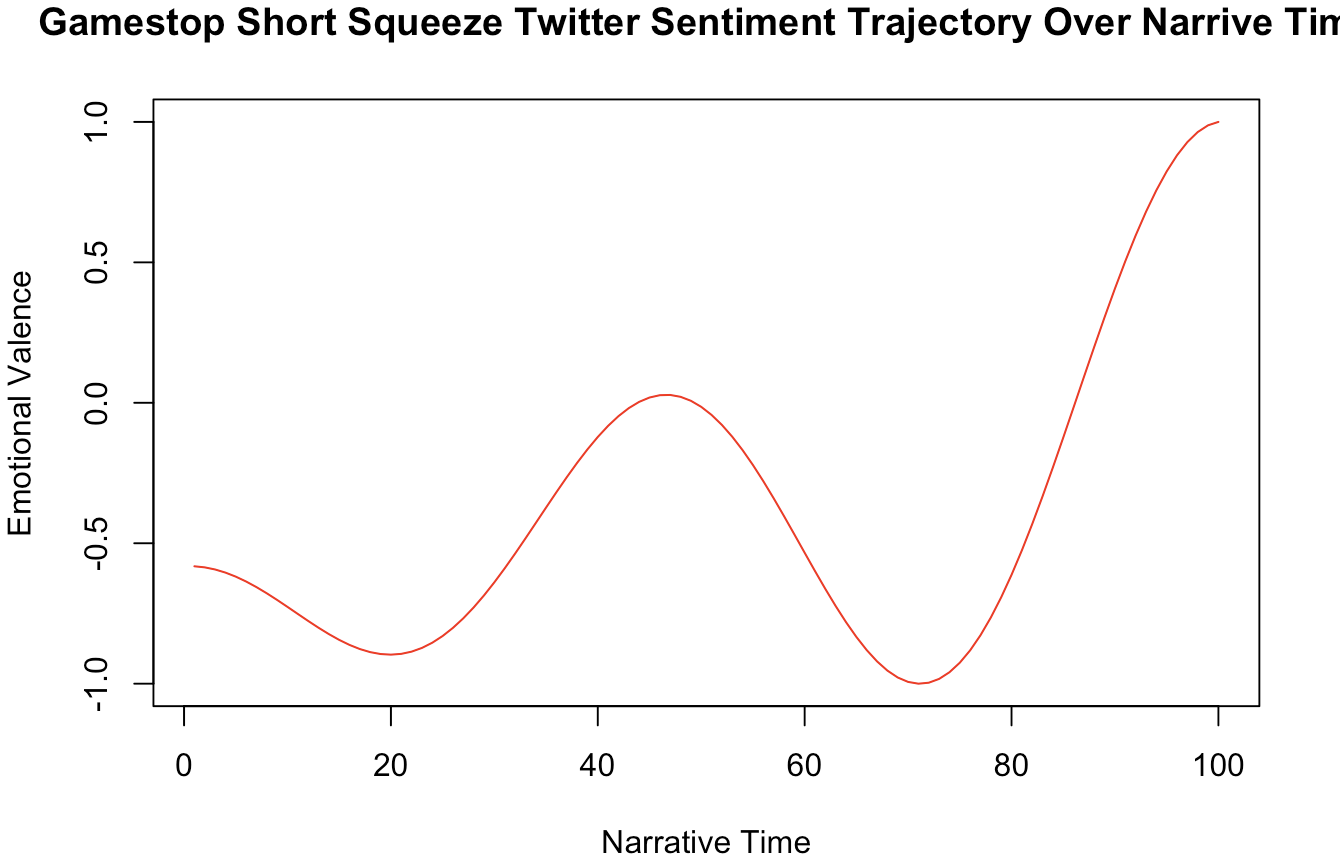


Emotional Valance analysis using Discrete Cosine Transformations technique (i.e.get_dct_transform(...))

```
reddit_wallstreetbets_comments_clean_vector_dct_values <- syuzhet::get_dct_transform
(reddit_wallstreetbets_comments_clean_vector,
                                                                    low_pass_size = 5,
                                                                    x_reverse_len = 10
0,
                                                                    scale_vals = FALS
E,
                                                                    scale_range = TRUE
                                                                   )

gamestop_short_squeeze_tweet_clean_vector_dct_values <- syuzhet::get_dct_transform(ga
mestop_short_squeeze_tweet_clean_vector,
                                                                    low_pass_size = 5,
                                                                    x_reverse_len = 10
0,
                                                                    scale_vals = FALS
E,
                                                                    scale_range = TRUE
                                                                   )

plot(
  reddit_wallstreetbets_comments_clean_vector_dct_values,
  type = "l",
  main = "Reddit Wallstreetbets Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "blue"
)
```



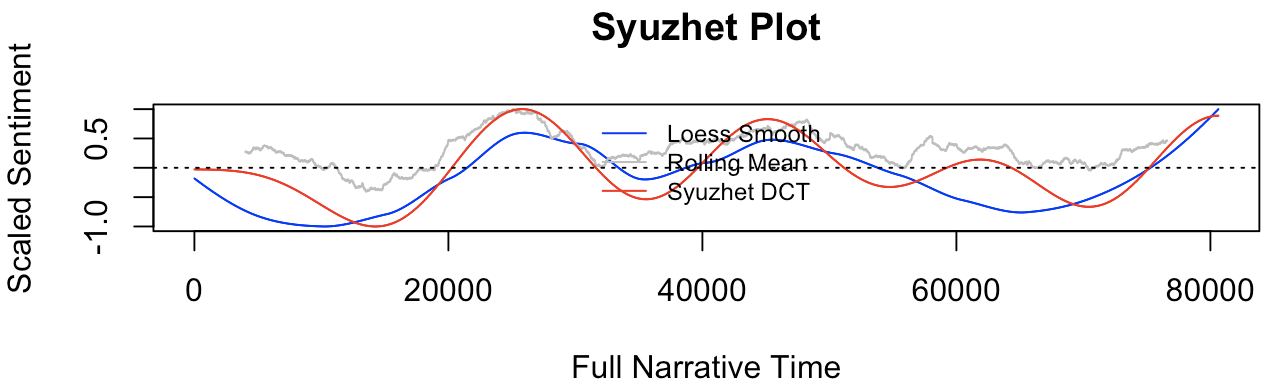

```
plot(
  gamestop_short_squeeze_tweet_clean_vector_dct_values,
  type = "l",
  main = "Gamestop Short Squeeze Twitter Sentiment Trajectory Over Narrive Time",
  xlab = "Narrative Time",
  ylab = "Emotional Valence",
  col = "red"
)
```

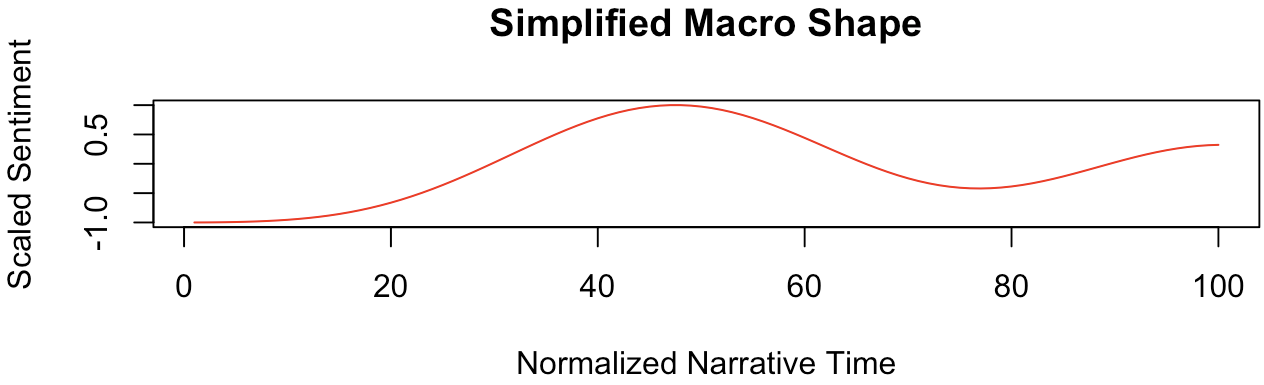


Explanation:- Main advantage is in its better representation of edge values in the smoothed version of the sentiment vector.

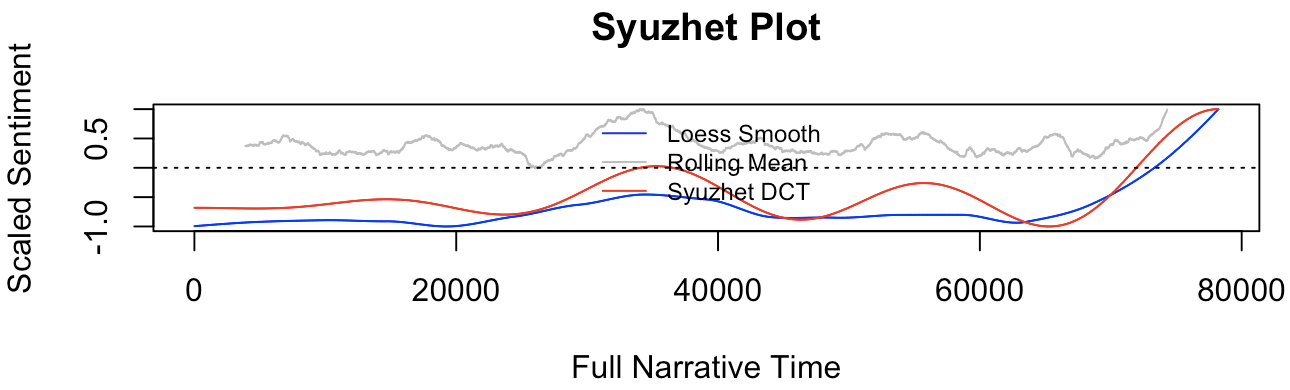
The simple_plot function takes a sentiment vector and applies three smoothing methods. The smoothers include a moving average, loess, and discrete cosine transformation. This function produces two plots stacked. The first shows all three smoothing methods on the same graph. The second graph shows only the DCT smoothed line, but does so on a normalized time axis. The shape of the DCT line in both the top and bottom graphs are identical.

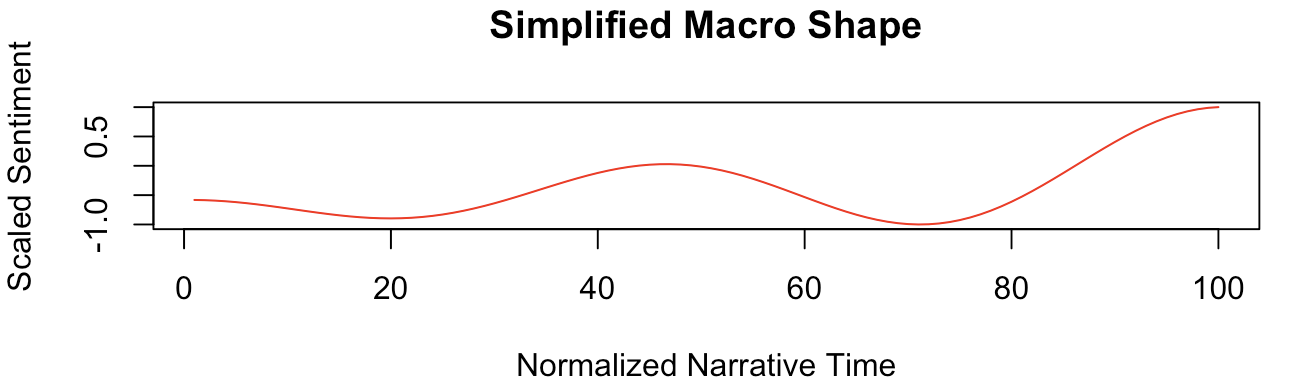
```
# Reddit Wallstreetbets
syuzhet::simple_plot(reddit_wallstreetbets_comments_clean_vector)
```





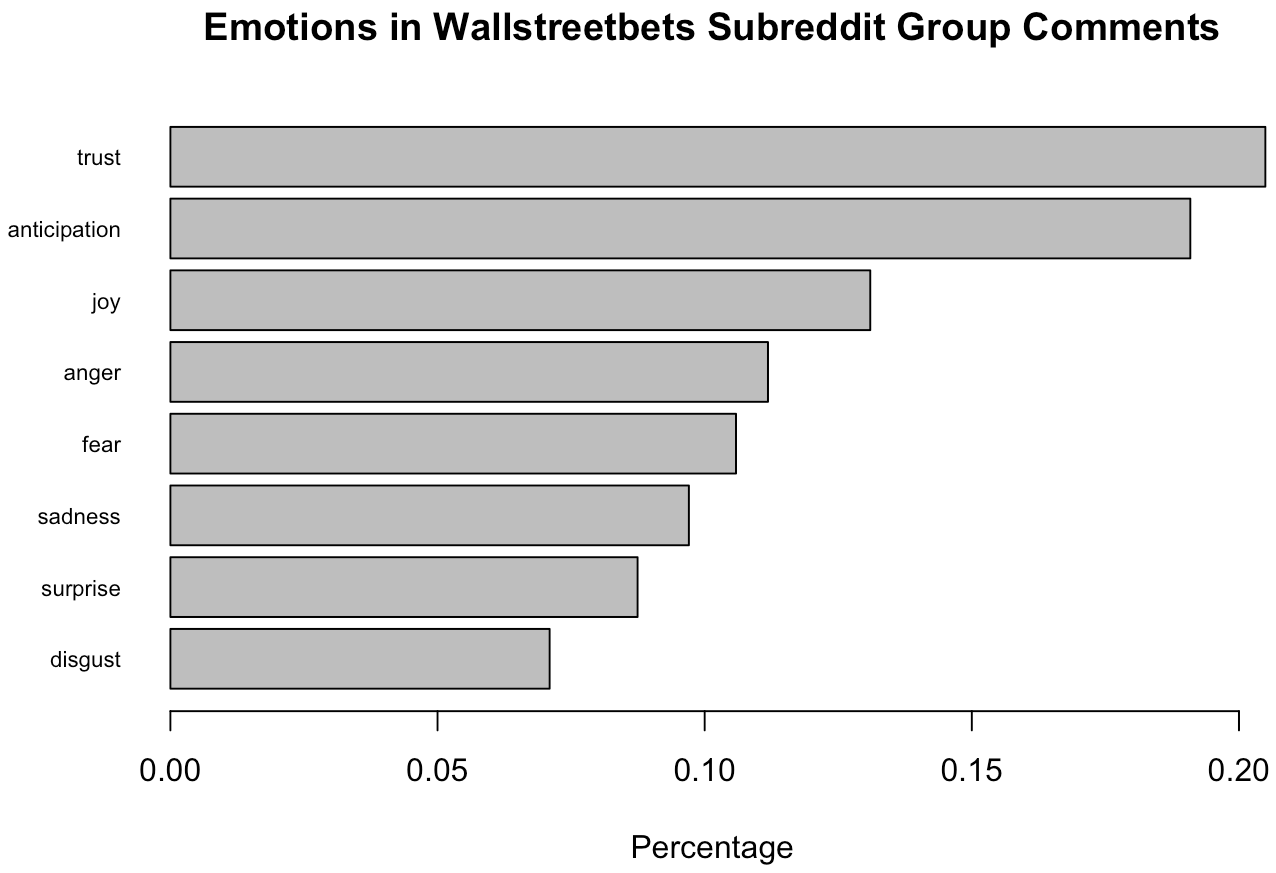

```
# Gamestop short squeeze twitter sentiment
syuzhet::simple_plot(gamestop_short_squeeze_tweet_clean_vector)
```



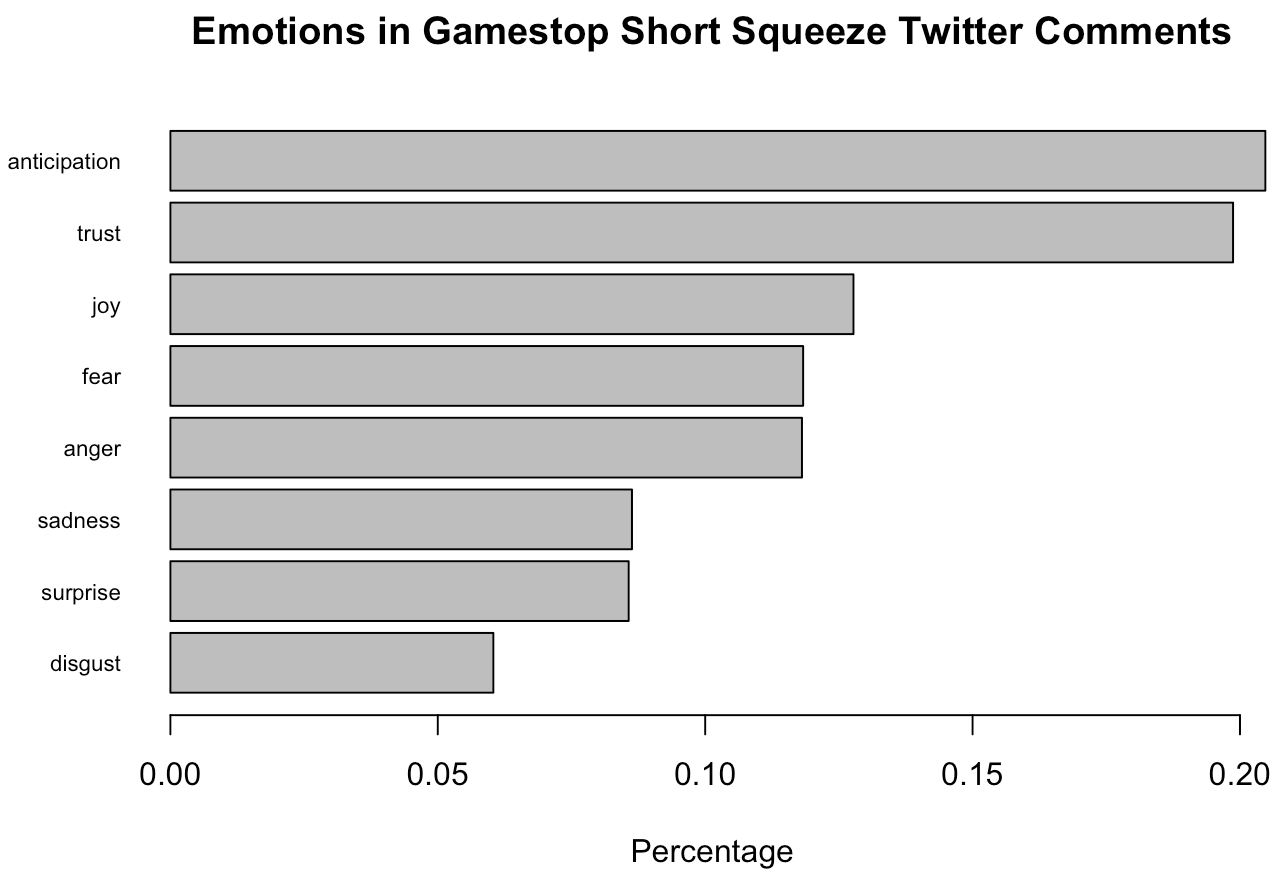


Emotional Valance using NRC lexicon dictianary.

```
reddit_wallstreetbets_comments_nrc_sentiment <- reddit_wallstreetbets_comments_clean
%>%
                                                syuzhet::get_sentences() %>%
                                                syuzhet::get_nrc_sentiment()

gamestop_short_squeeze_tweet_nrc_sentiment <- gamestop_short_squeeze_tweet_clean %>%
                                                syuzhet::get_sentences() %>%
                                                syuzhet::get_nrc_sentiment()

barplot(
  sort(colSums(prop.table(reddit_wallstreetbets_comments_nrc_sentiment[, 1:8]))),
  horiz = TRUE,
  cex.names = 0.7,
  las = 1,
  main = "Emotions in Wallstreetbets Subreddit Group Comments",
  xlab = "Percentage"
)
```




```
barplot(
  sort(colSums(prop.table(gamestop_short_squeeze_tweet_nrc_sentiment[, 1:8]))),
  horiz = TRUE,
  cex.names = 0.7,
  las = 1,
  main = "Emotions in Gamestop Short Squeeze Twitter Comments",
  xlab = "Percentage"
)
```



Explanation:- trust and anticipation top 2 sentiment category in Reddit Wallstreetbets comments, gamestop short queeze twitter tweets. In case of Reddit Wallstreetbets group comments, more than 20% comments are trust related whereas gamestop short squeeze tweets are also more than 20% related to trust and anticipation.

Comparitive study of Sentiment Analysis using lexicon dictionary from syuzhet package.

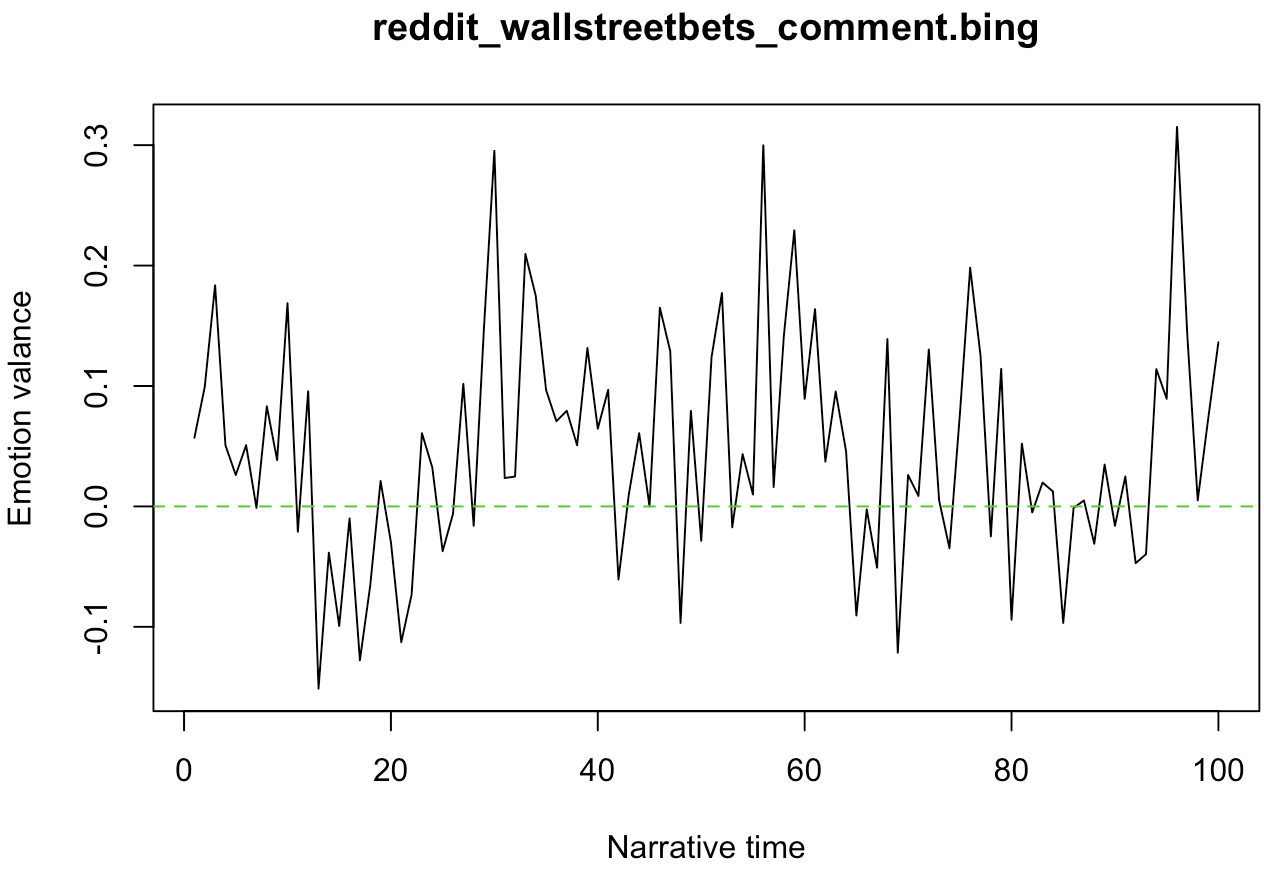
```
reddit_wallstreetbets_comments_sentiment_text <-
  list(reddit_wallstreetbets_comment = reddit_wallstreetbets_comments_clean) %>%
                    lapply(syuzhet::get_sentences)

gamestop_short_squeeze_tweet_sentiment_text <-
  list(gamestop_short_squeeze_tweet = gamestop_short_squeeze_tweet_clean) %>%
                    lapply(syuzhet::get_sentences)

syuzhet_multiple_sentiment <- function(sentences) {
  list(
    bing = syuzhet::get_sentiment(sentences, method = "bing"),
    afinn = syuzhet::get_sentiment(sentences, method = "afinn"),
    nrc = syuzhet::get_sentiment(sentences, method = "nrc"),
    syuzhet = syuzhet::get_sentiment(sentences, method = "syuzhet")
  )
}

reddit_wallstreetbets_comments_sentiment <- reddit_wallstreetbets_comments_sentiment_
text %>%
  lapply(syuzhet_multiple_sentiment)

gamestop_short_squeeze_tweet_sentiment <- gamestop_short_squeeze_tweet_sentiment_text
%>%
                                          lapply(syuzhet_multiple_sentiment)

sum_up_sentiment <- function(x) {
  apply_sentiment <- function(vec) {
    list(sum = sum(vec),
         mean = mean(vec),
         summary = summary(vec))
  }

  if(is.list(x))
    lapply(x, apply_sentiment)
  else
    apply_sentiment(x)
}

reddit_wallstreetbets_comments_sentiment %>%
  lapply(sum_up_sentiment) %>%
  list.unzip()
```

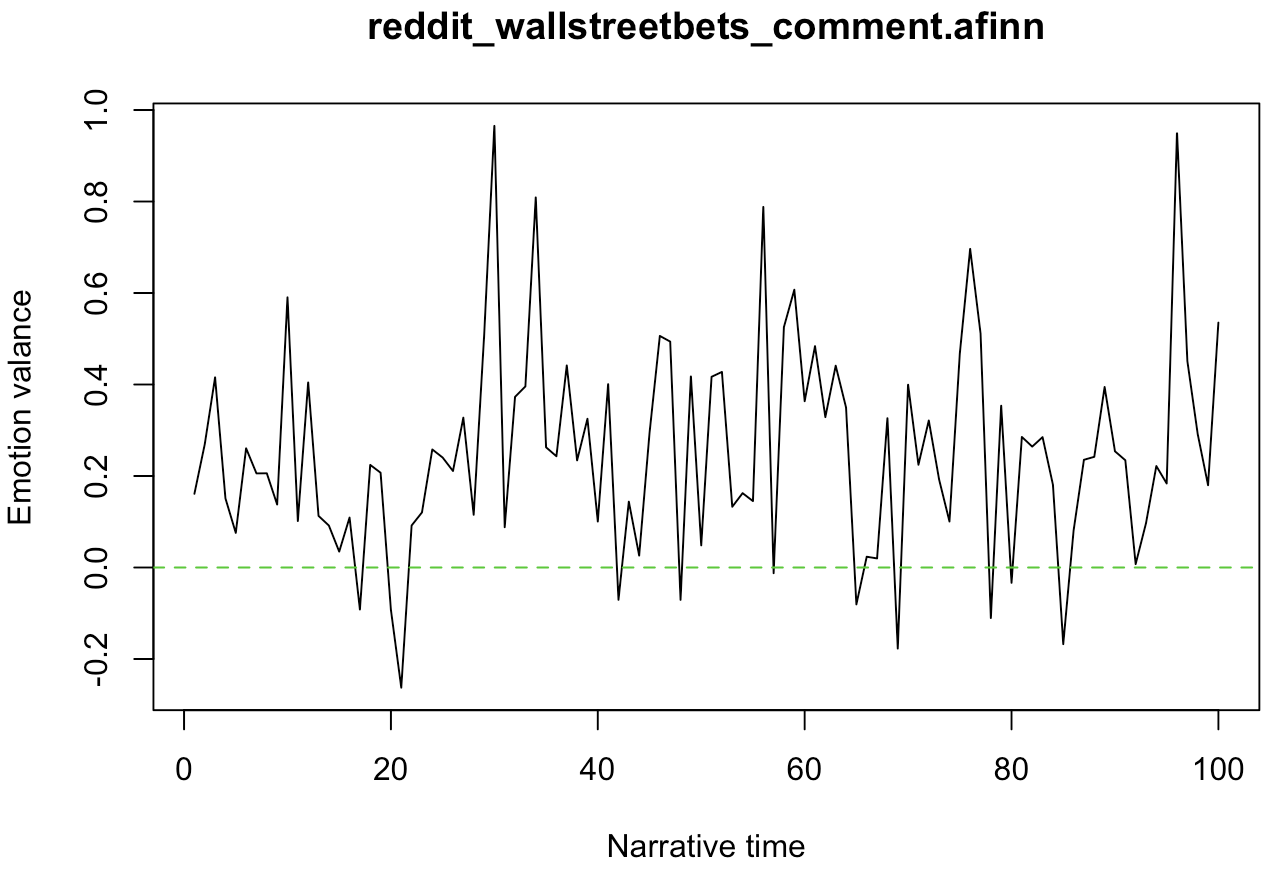
```
## $bing
##         reddit_wallstreetbets_comment
## sum     3749
## mean    0.04649634
## summary Numeric,6
##
## $afinn
##         reddit_wallstreetbets_comment
## sum     20325
## mean    0.2520774
## summary Numeric,6
##
## $nrc
##         reddit_wallstreetbets_comment
## sum     14500
## mean    0.1798338
## summary Numeric,6
##
## $syuzhet
##         reddit_wallstreetbets_comment
## sum     13340.25
## mean    0.1654502
## summary Numeric,6
```

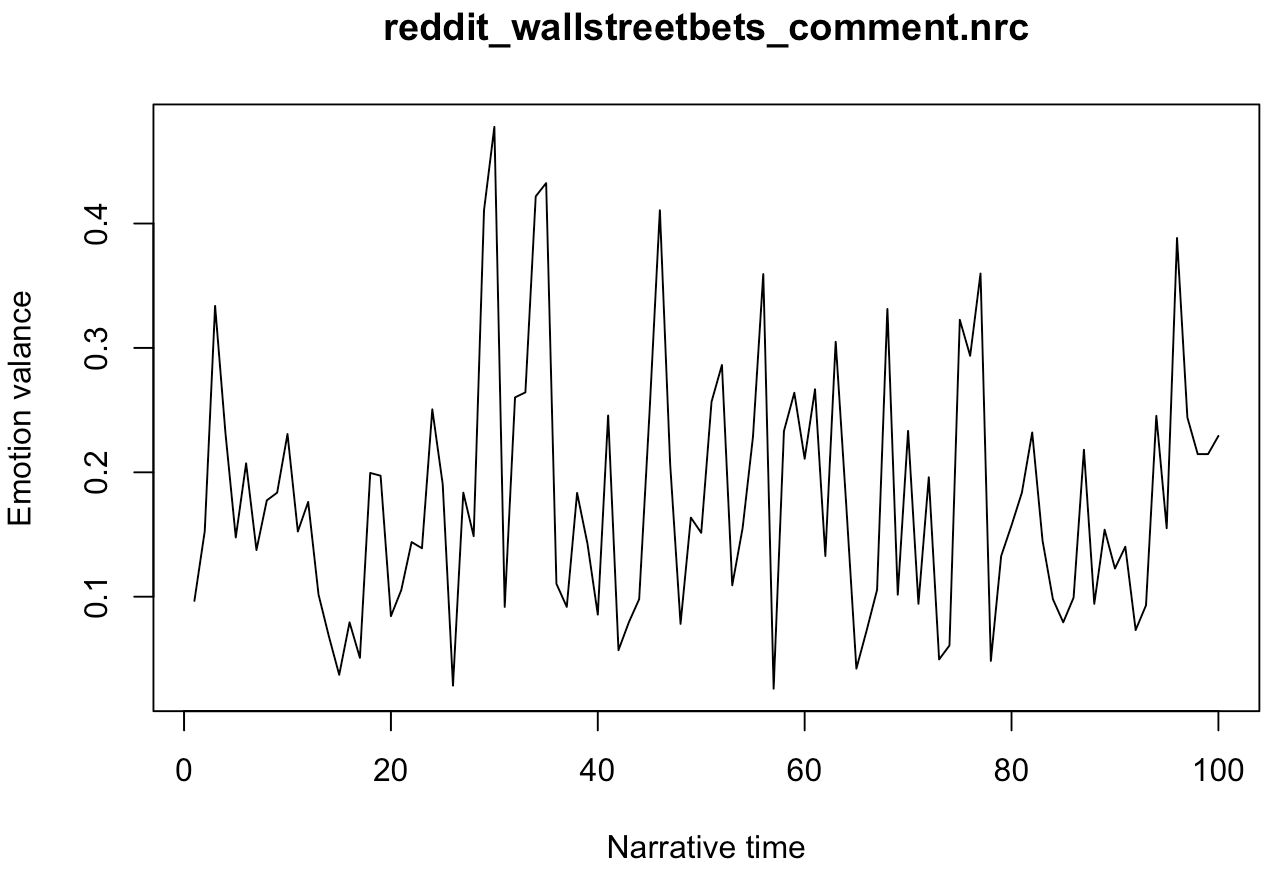
```
gamestop_short_squeeze_tweet_sentiment %>%
  lapply(sum_up_sentiment) %>%
  list.unzip()
```

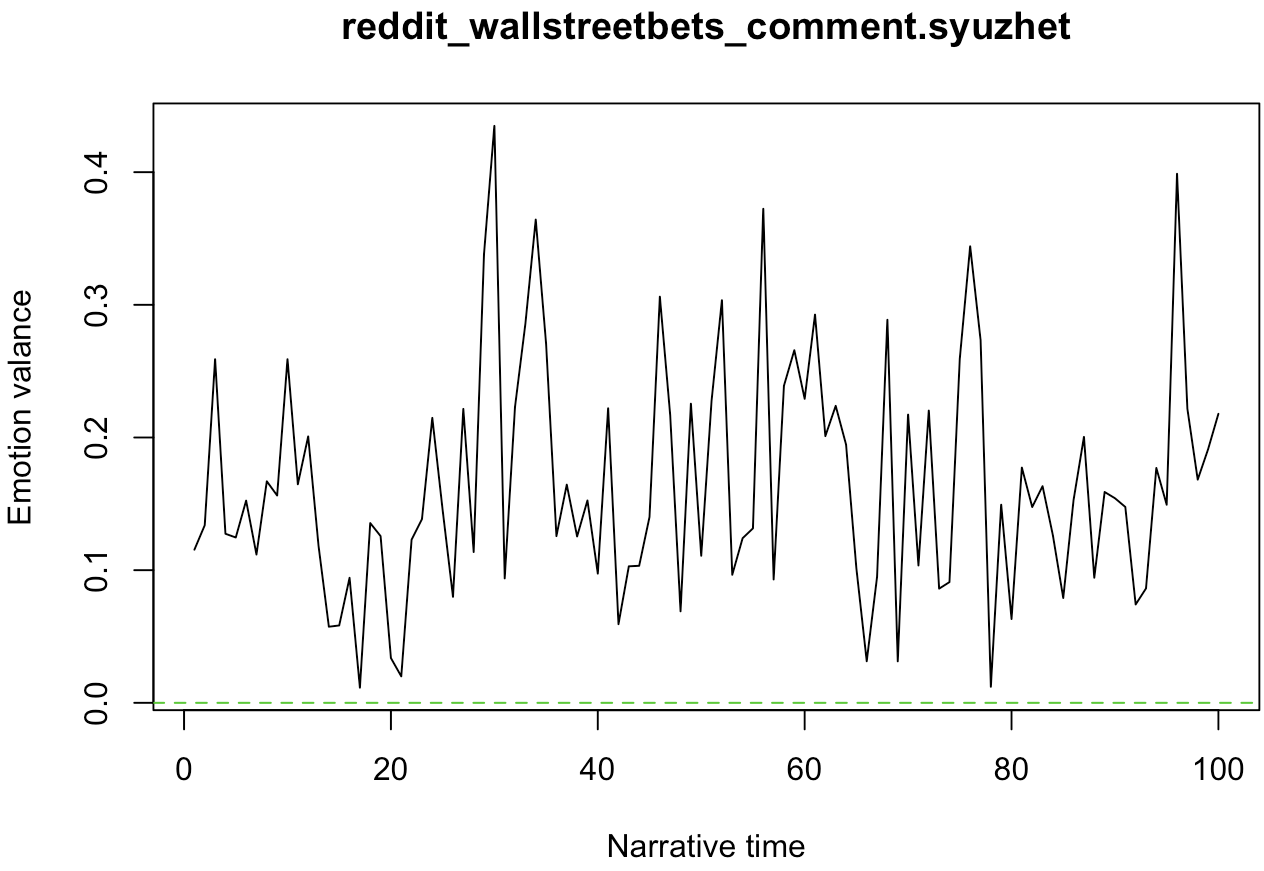
```
## $bing
##         gamestop_short_squeeze_tweet
## sum     6971
## mean    0.08911929
## summary Numeric,6
##
## $afinn
##         gamestop_short_squeeze_tweet
## sum     36099
## mean    0.4615001
## summary Numeric,6
##
## $nrc
##         gamestop_short_squeeze_tweet
## sum     14150
## mean    0.1808977
## summary Numeric,6
##
## $syuzhet
##         gamestop_short_squeeze_tweet
## sum     18307.35
## mean    0.2340465
## summary Numeric,6
```

```
plot_sentiment <- function(x, title) {
  plot(x,
       type = "l",
       main = title,
       xlab = "Narrative time",
       ylab = "Emotion valance",
       # ylim = c(-1.5, 3.25) # roughly the min and the max
       )
  abline(h = 0, col = 3, lty = 2) # neutral sentiment
}

reddit_wallstreetbets_comments_sentiment %>%
  list.flatten() %>%
  lapply(syuzhet::get_percentage_values) %>%
  Map(plot_sentiment, ., names(.))
```


reddit_wallstreetbets_comment.bing
Emotion valance
-0.1
0.0
0.1
0.2
0.3
0
20
40
60
80
100
Narrative time

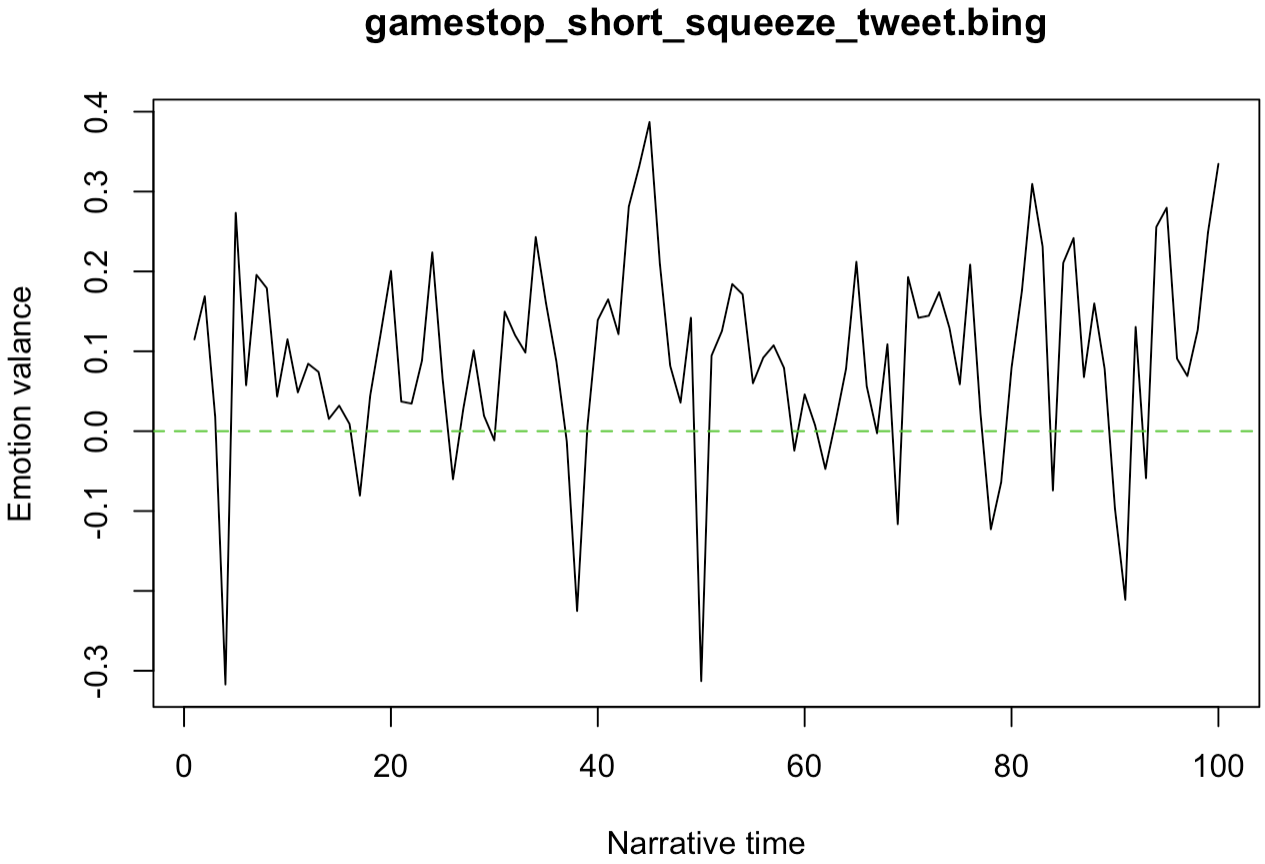

reddit_wallstreetbets_comment.afinn
Emotion valance
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Narrative time

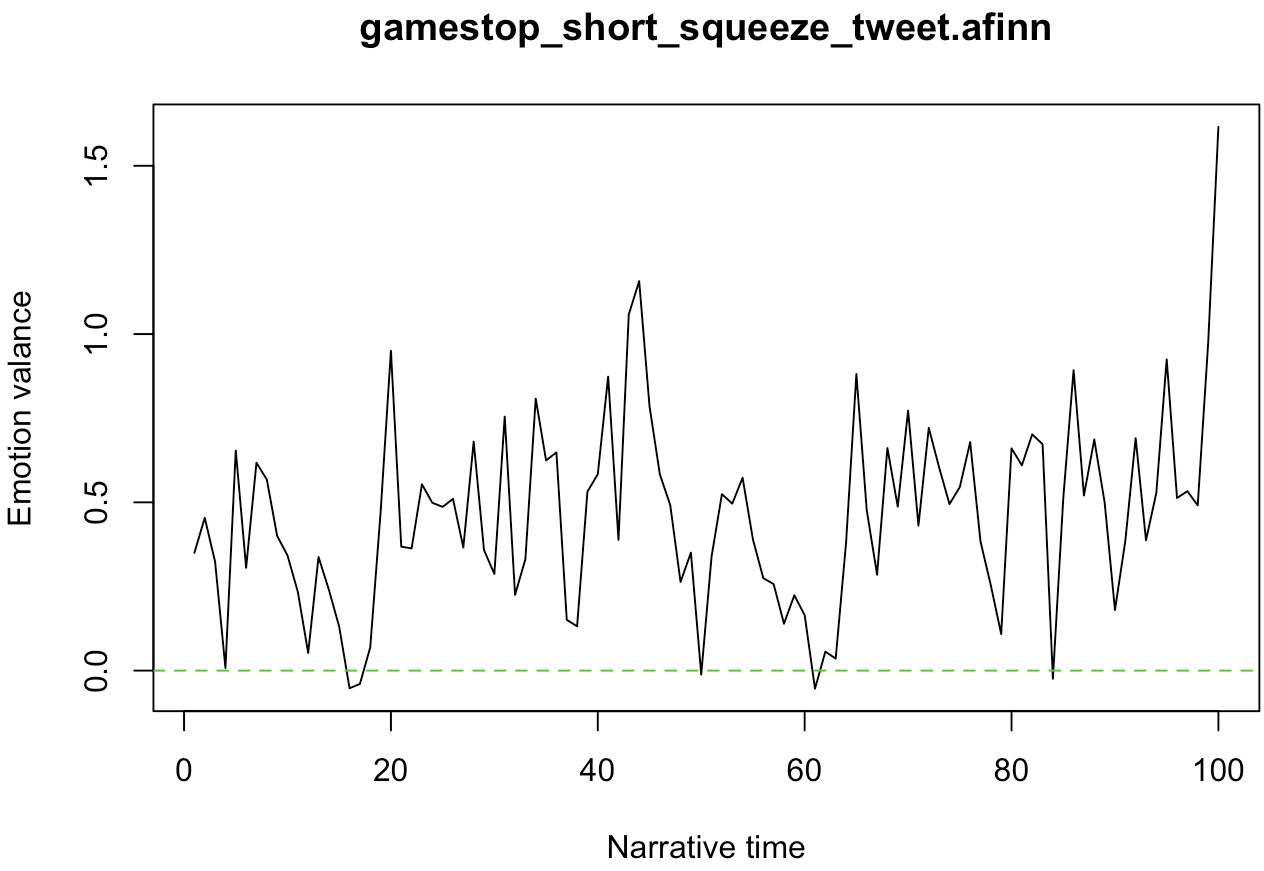

reddit_wallstreetbets_comment.nrc
Emotion valance
0.1
0.2
0.3
0.4
0
20
40
60
80
100
Narrative time

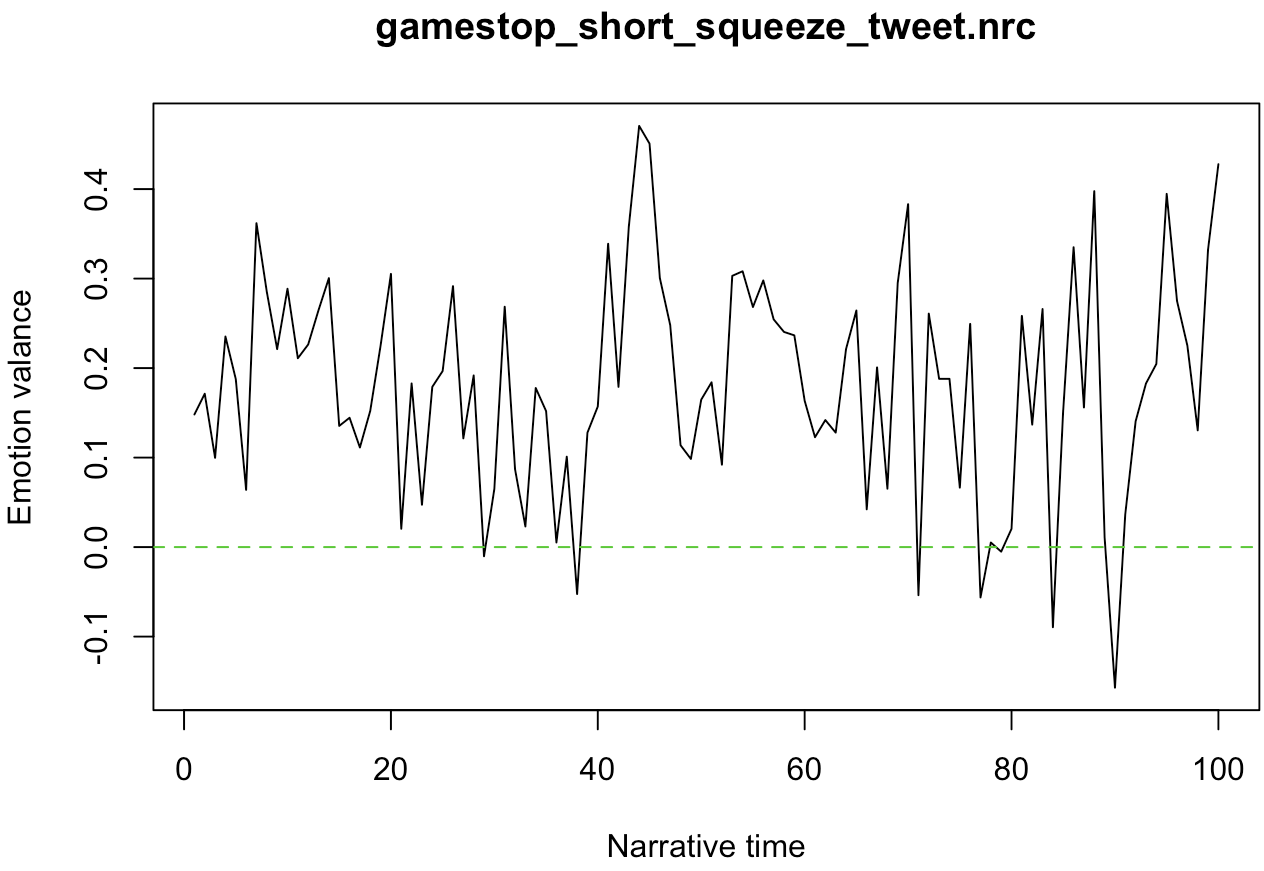

reddit_wallstreetbets_comment.syuzhet
Emotion valance
0.0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
Narrative time

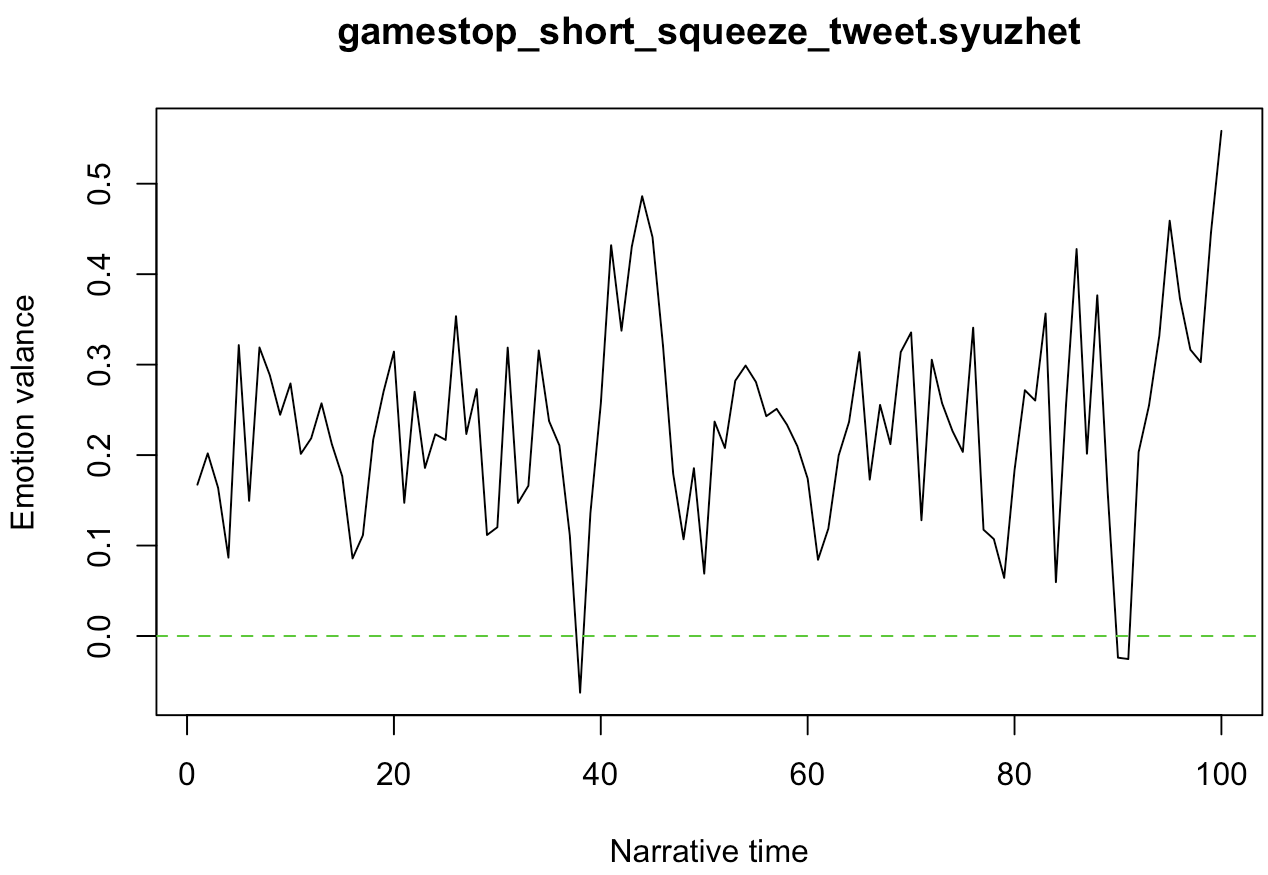
```
## $reddit_wallstreetbets_comment.bing
## NULL
##
## $reddit_wallstreetbets_comment.afinn
## NULL
##
## $reddit_wallstreetbets_comment.nrc
## NULL
##
## $reddit_wallstreetbets_comment.syuzhet
## NULL
```

```
gamestop_short_squeeze_tweet_sentiment %>%
  list.flatten() %>%
  lapply(syuzhet::get_percentage_values) %>%
  Map(plot_sentiment, ., names(.))
```




gamestop_short_squeeze_tweet.afinn
Emotion valance
0.0
0.5
1.0
1.5
0
20
40
60
80
100
Narrative time


gamestop_short_squeeze_tweet.nrc
Emotion valance
-0.1
0.0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
Narrative time

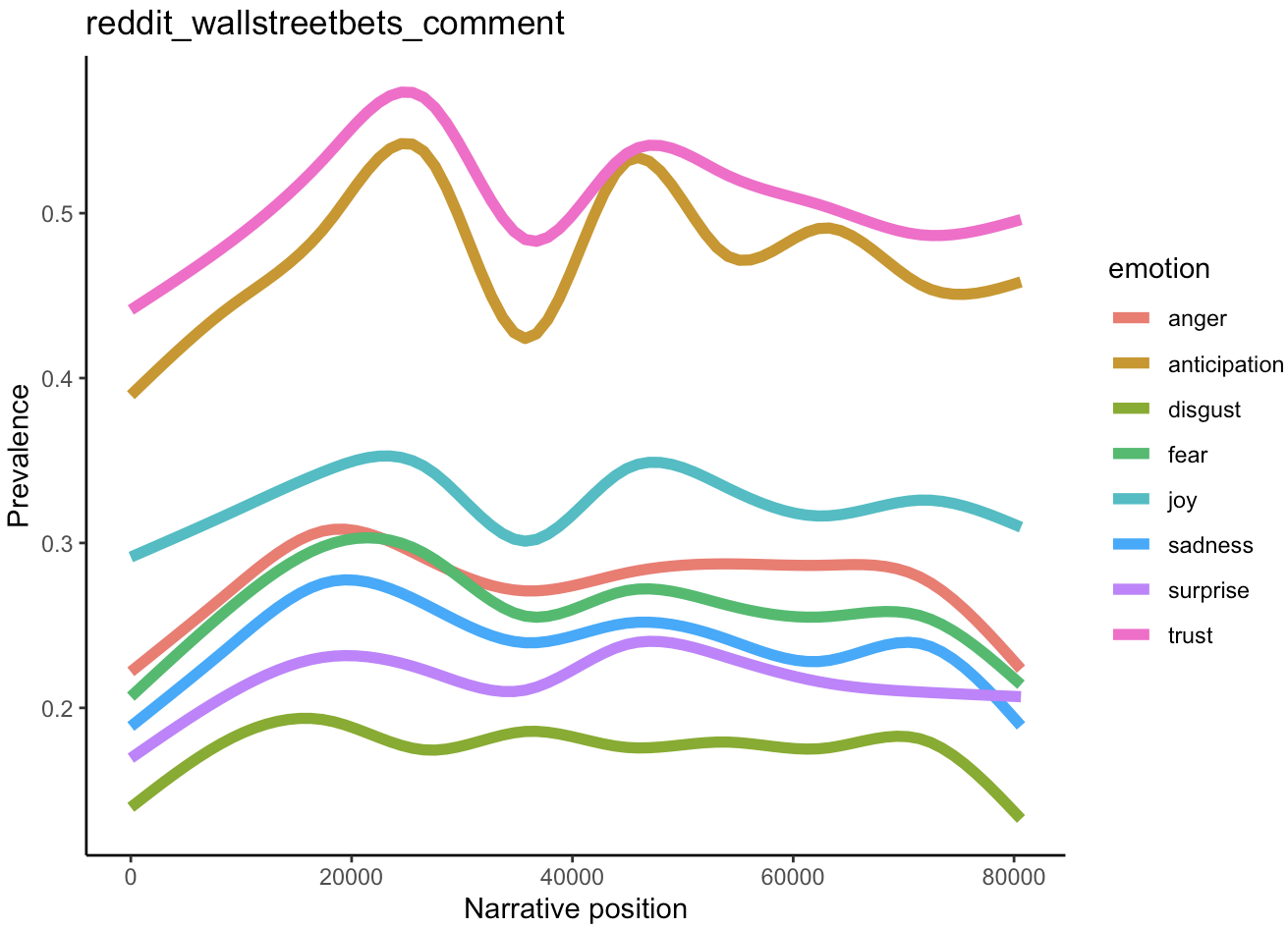



```
## $gamestop_short_squeeze_tweet.bing
## NULL
##
## $gamestop_short_squeeze_tweet.afinn
## NULL
##
## $gamestop_short_squeeze_tweet.nrc
## NULL
##
## $gamestop_short_squeeze_tweet.syuzhet
## NULL
```

Sentiment Analysis Using Syuzhet's get_nrc_sentiment(...).

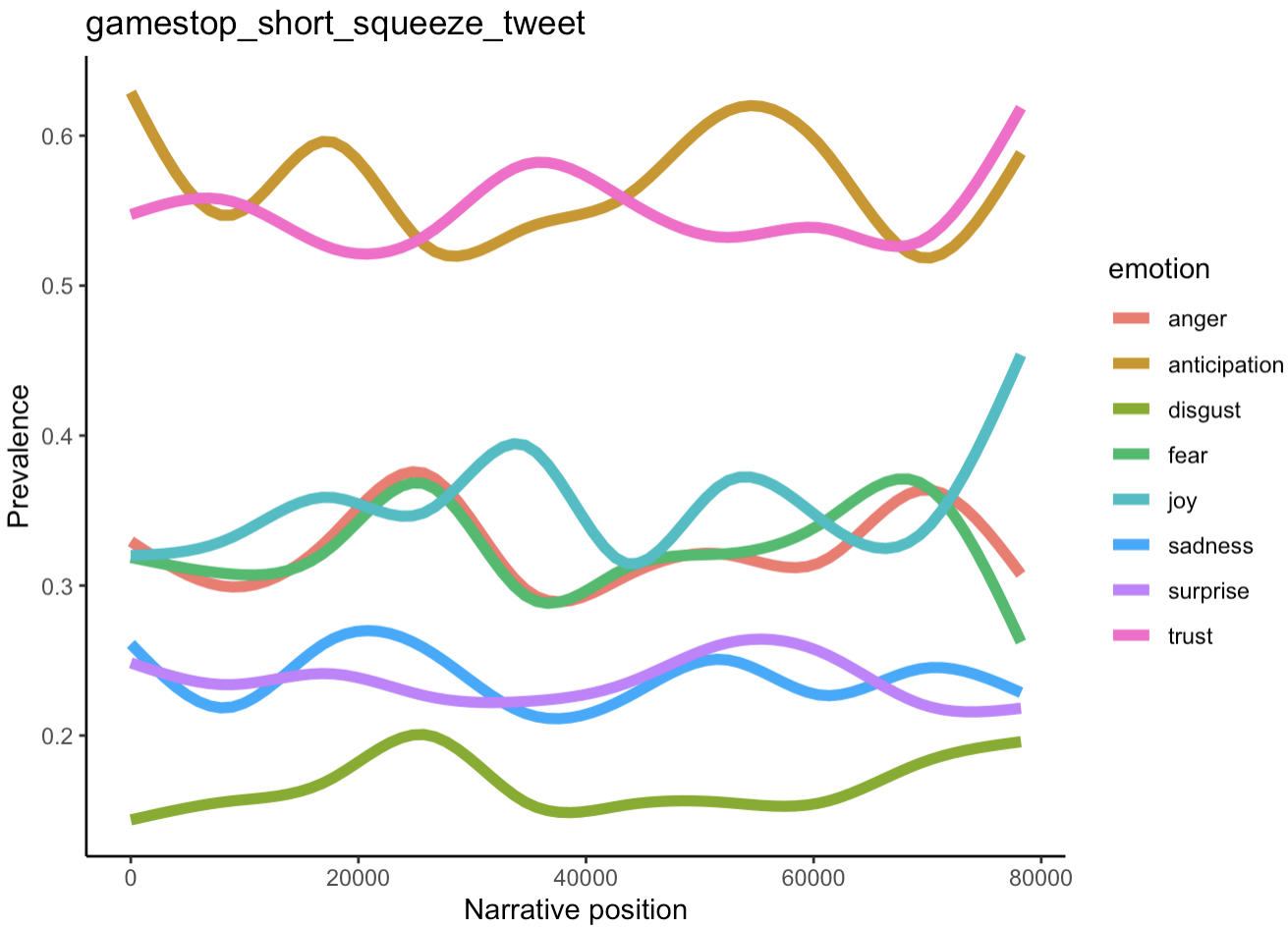
```
bind_pos <- function(df) {
  pos <- data.frame(position = 1:nrow(df))
  cbind(df, pos)
}

plot_nrc <- function(df, title) {
  ggplot(df, aes(x = position, y = value, color = emotion)) +
    geom_smooth(size = 2, se = FALSE) +
    xlab("Narrative position") +
    ylab("Prevalence") +
    theme_classic() +
    ggtitle(title)
}

reddit_wallstreetbets_comments_sentiment_text %>%
  lapply(syuzhet::get_nrc_sentiment) %>%
  lapply(bind_pos) %>%
  lapply(gather, emotion, value, -position, -negative, -positive) %>%
  Map(plot_nrc, ., names(.))
```

```
## $reddit_wallstreetbets_comment
```



```
gamestop_short_squeeze_tweet_sentiment_text %>%
  lapply(syuzhet::get_nrc_sentiment) %>%
  lapply(bind_pos) %>%
  lapply(gather, emotion, value, -position, -negative, -positive) %>%
  Map(plot_nrc, ., names(.))
```

```
## $gamestop_short_squeeze_tweet
```



Explanation:- after processing more than 80K reddit wallstreetbets comments / gamestop short squeeze twitter tweet found that trust and anticipation are top most emotion stand out. Above chart depicts that over the 80K narrative, how various categories are sentiment changes and trust and anticipantion are top most emotion in that case as well.

---

sentimentR package is designed to quickly calculate text polarity sentiment at the sentence level and optionally aggregate by rows or grouping variable(s).

Limitations of syuzhet package

Matthew Jockers created the syuzhet package that utilizes dictionary lookups for the Bing, NRC, and Afinn methods as well as a custom dictionary. He also utilizes a wrapper for the Stanford coreNLP which uses much more sophisticated analysis. Jocker's dictionary methods are fast but are more prone to error in the case of valence shifters.

So what does sentimentr do that other packages don't and why does it matter?

sentimentr attempts to take into account valence shifters (i.e., negators, amplifiers (intensifiers), de-amplifiers (downtoners), and adversative conjunctions) while maintaining speed. Simply put, sentimentr is an augmented dictionary lookup. The next questions address why it matters.

So what are these valence shifters?

- A negator flips the sign of a polarized word (e.g., "I do not like it."). See lexicon::hash_valence_shifters[y==1] for examples.
- An amplifier (intensifier) increases the impact of a polarized word (e.g., "I really like it."). See lexicon::hash_valence_shifters[y==2] for examples.
- A de-amplifier (downtoner) reduces the impact of a polarized word (e.g., "I hardly like it."). See lexicon::hash_valence_shifters[y==3] for examples.
- An adversative conjunction overrules the previous clause containing a polarized word (e.g., "I like it but it's not worth it."). See lexicon::hash_valence_shifters[y==4] for examples.

Do valence shifters really matter?

Well valence shifters affect the polarized words. In the case of negators and adversative conjunctions the entire sentiment of the clause may be reversed or overruled. So if valence shifters occur fairly frequently a simple dictionary lookup may not be modeling the sentiment appropriately. You may be wondering how frequently these valence shifters co-occur with polarized words, potentially changing, or even reversing and overruling the clause's sentiment. The table below shows the rate of sentence level co-occurrence of valence shifters with polarized words across a few types of texts.

Let's take an example to understand how valance shifter works in sentimentr package.

```
c("I do not like it.", "I really like it.", "I hardly like it.", "I like it but it's
not worth it.") %>%
  get_sentences() %>%
  sentiment()
```

```
##    element_id sentence_id word_count  sentiment
## 1:          1           1          5 -0.2236068
## 2:          2           1          4  0.4500000
## 3:          3           1          4  0.0500000
## 4:          4           1          9 -0.5623333
```

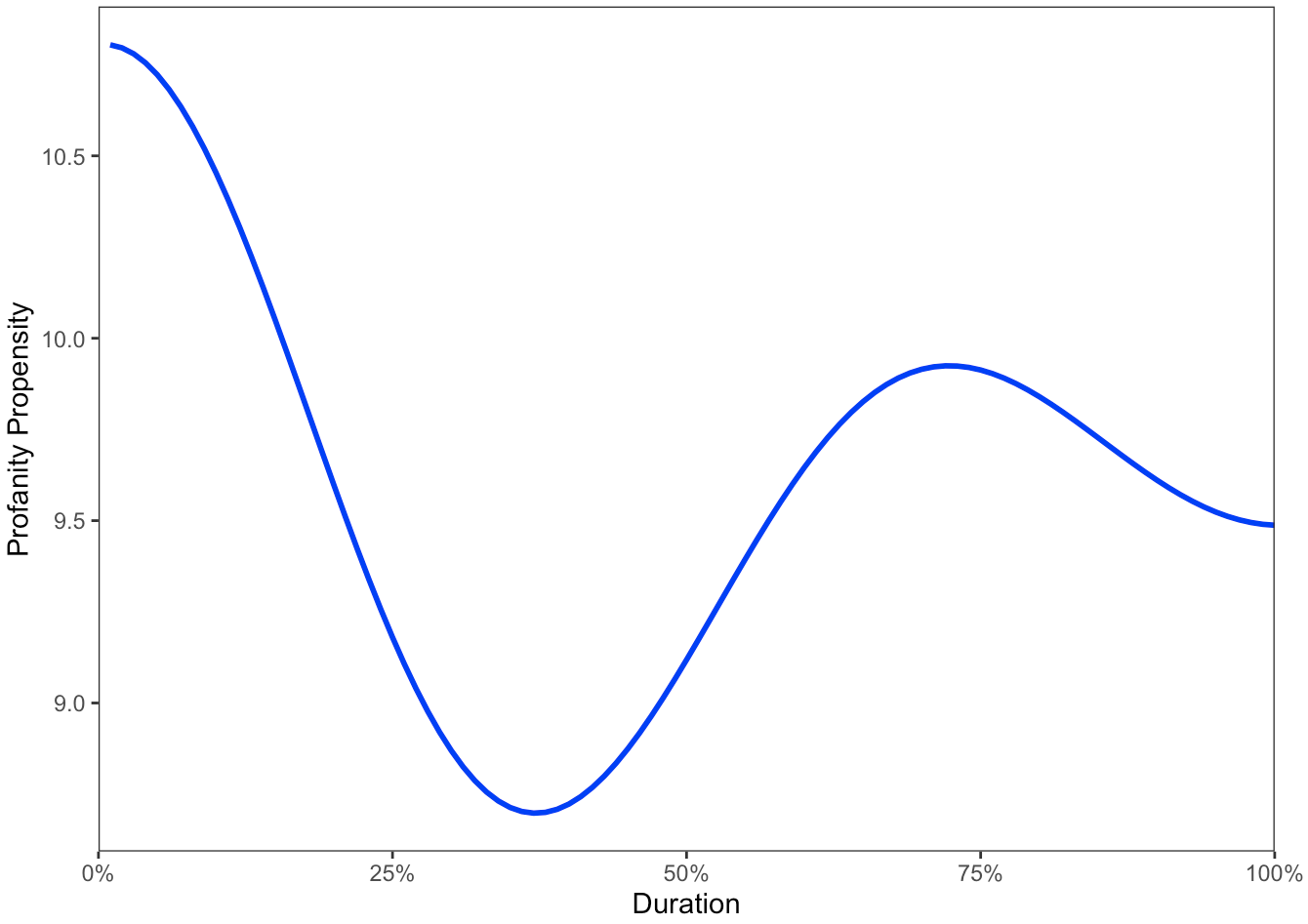
Impact of Valance Shifter in Reddit Wallstreetbets subreddit group's comments and Gamestop short squeeze twitter comments.

```
comments_attributes_rate <- list(
  sentiment_attributes(reddit_wallstreetbets_comments_clean),
  sentiment_attributes(gamestop_short_squeeze_tweet_clean)
) %>%
  lapply(function(y){
    x <- y[['Polarized_Cooccurrences']]
    data.frame(setNames(as.list(f_prop2percent(x[[2]], 0)), gsub('-', '', x[[1]])),
               stringsAsFacto1rs = FALSE, check.names = FALSE)
  }) %>%
  setNames(c('Reddit Wallstreetbets Comments', 'Gamestop Short Squeeze Twitter Commen
ts')) %>%
  tidy_list('text')

comments_attributes_rate
```

```
##                                       text negator amplifier deamplifier
## 1:          Reddit Wallstreetbets Comments     26%       17%          5%
## 2: Gamestop Short Squeeze Twitter Comments     28%       15%          2%
##    adversative stringsAsFacto1rs
## 1:         13%             FALSE
## 2:          8%             FALSE
```

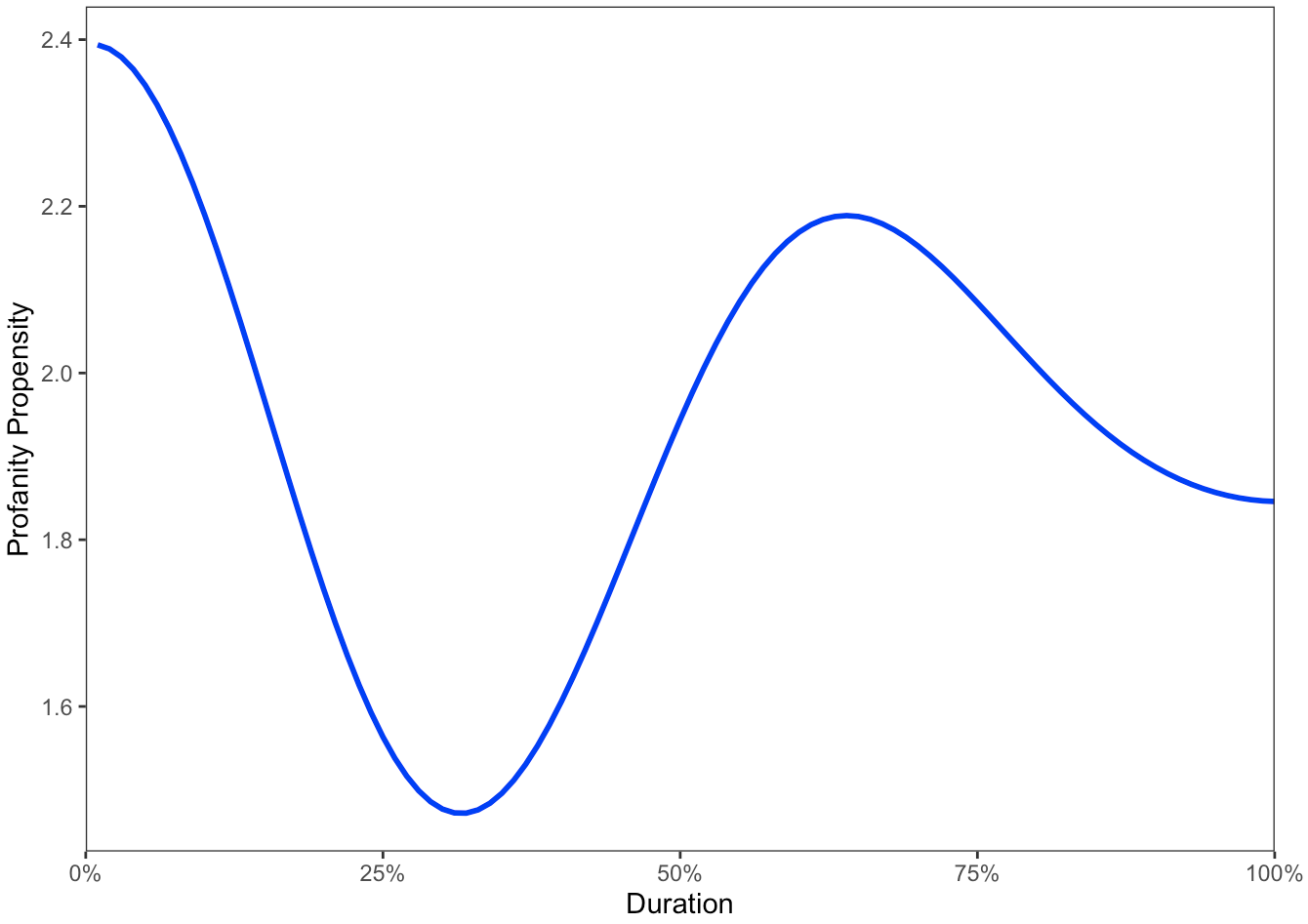
Determine Profanity of Reddit's Wallstreetbets comments using sentimentr package.

```
reddit_wallstreetbets_clean_df <- reddit_wallstreetbets %>%
                                    mutate(clean_comments = f_gsub_clean_data(commen
t))

reddit_wallstreetbets_clean_df$clean_comments %>%
  sentimentr::get_sentences() %>%
  sentimentr::profanity() %>%
  plot()
```



Determine Profanity of Gamestop short squeeze twitter tweets using sentimentr package.

```
gamestop_short_squeeze_tweet_clean_df <- gamestop_short_squeeze_tweet %>%
                                          mutate(clean_tweet = f_gsub_clean_data(tex
t))

gamestop_short_squeeze_tweet_clean_df$clean_tweet %>%
  sentimentr::get_sentences() %>%
  sentimentr::profanity() %>%
  plot()
```



Extact Profanity terms using sentimentr package.

```
reddit_wallstreetbets_profanity_df <- reddit_wallstreetbets_clean_df$clean_comments %
>%
                                      sentimentr::get_sentences() %>%
                                      sentimentr::extract_profanity_terms() %>%
                                      attributes()

reddit_wallstreetbets_profanity_df$counts %>%
  head()
```

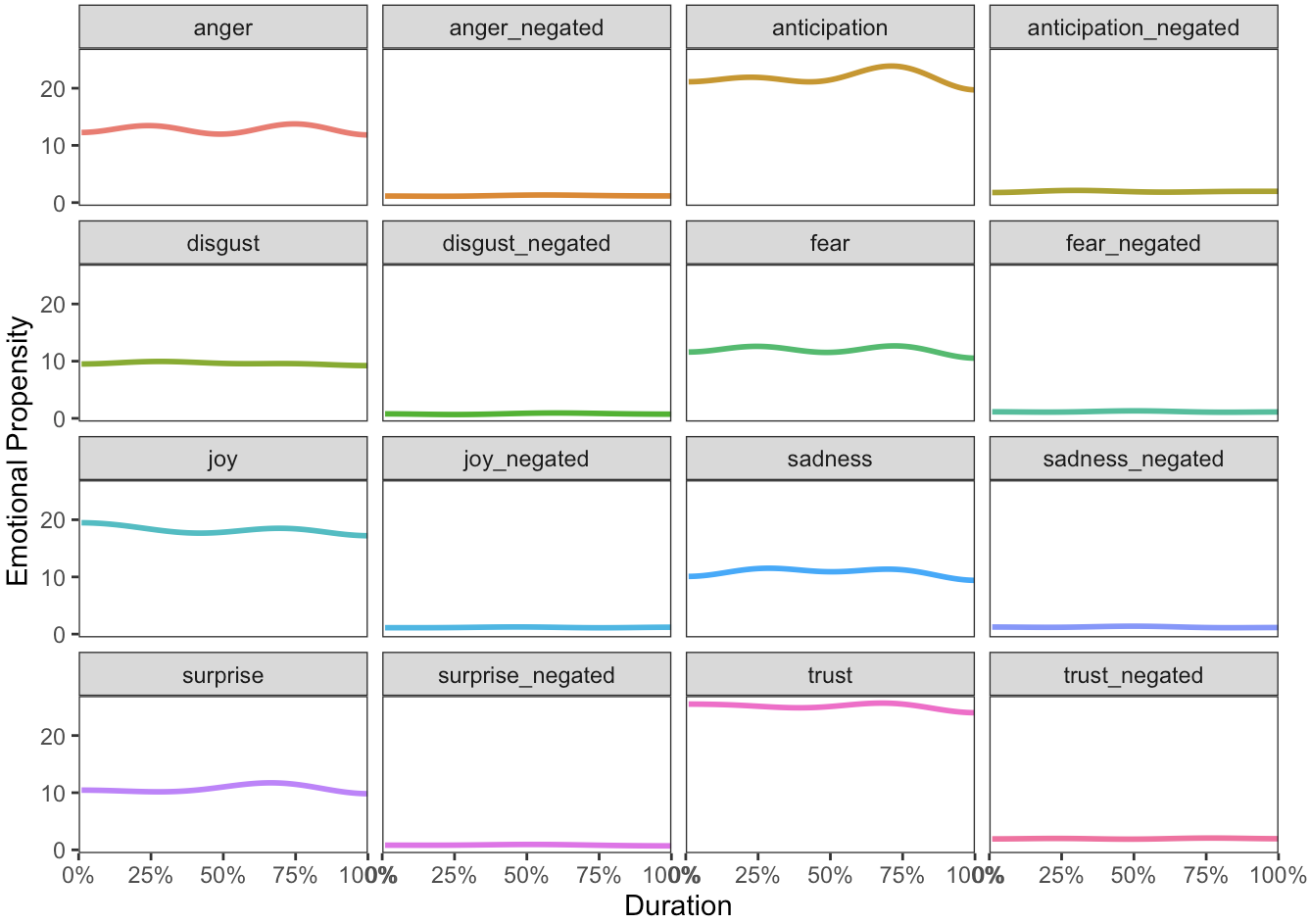
```
##      words profanity    n
## 1:    shit         1 2063
## 2:    fuck         1 2031
## 3: fucking         1 1634
## 4:  retard         1  681
## 5:     ass         1  439
## 6:    hell         1  418
```

```
gamestop_short_squeeze_tweet_profanity_df <- gamestop_short_squeeze_tweet_clean_df$cl
ean_tweet %>%
                                                sentimentr::get_sentences() %>%
                                                sentimentr::extract_profanity_terms() %
>%
                                                attributes()

gamestop_short_squeeze_tweet_profanity_df$counts %>%
  head()
```

```
##      words profanity    n
## 1: fucking         1 1004
## 2:    shit         1  755
## 3:    fuck         1  582
## 4:     ass         1  392
## 5:    hell         1  325
## 6:    tits         1   72
```

Determine emotion valance of reddit wallstreetbets group comments using sentimentr package.

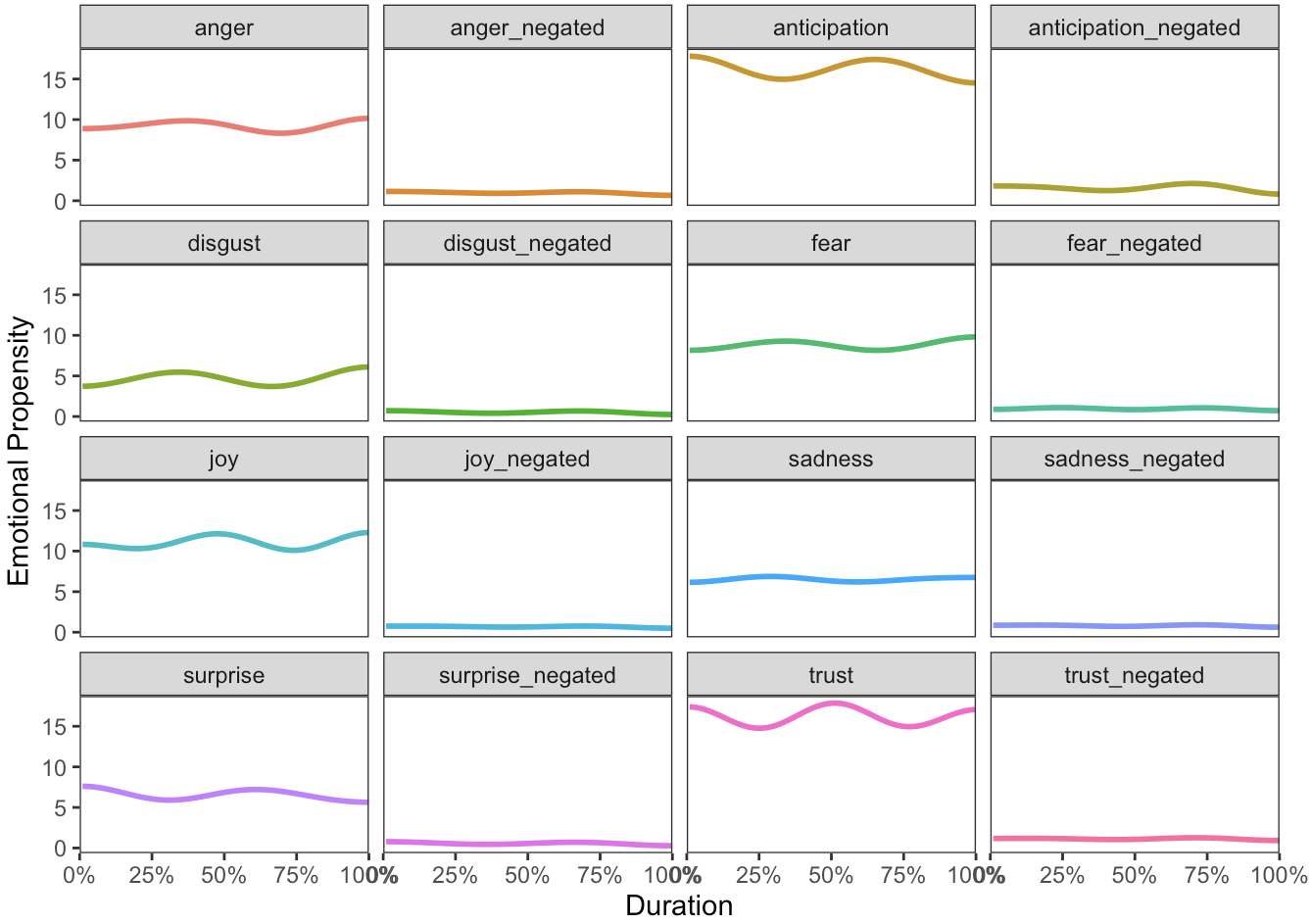
```
reddit_wallstreetbets_clean_df$clean_comments %>%
  sentimentr::get_sentences() %>%
  sentimentr::emotion() %>%
  plot(drop.unused.emotions = TRUE)
```



Explanation:- It happened to be trust and anticipation are top most emotion valance as per sentimentr package as well.

Determine emotion valance of gamestop short squeeze twitter's tweet using sentimentr package.

```
gamestop_short_squeeze_tweet_clean_df$clean_tweet %>%
  sentimentr::get_sentences() %>%
  sentimentr::emotion() %>%
  plot(drop.unused.emotions = TRUE)
```



Explanation:- It happened to be trust and anticipation are top most emotion valance as per sentimentr package as well.

Extract emotion terms using sentimentr package.

```
reddit_wallstreetbets_emotion_df <- reddit_wallstreetbets_clean_df$clean_comments %>%
                                    sentimentr::get_sentences() %>%
                                    sentimentr::extract_emotion_terms() %>%
                                    attributes()

reddit_wallstreetbets_emotion_df$counts %>%
  distinct(words) %>%
  head()
```

```
##    words
## 1: money
## 2:  good
## 3:  time
## 4: calls
## 5:  shit
## 6:   don
```

```
gamestop_short_squeeze_tweet_emotion_df <- gamestop_short_squeeze_tweet_clean_df$clea
n_tweet %>%
                                           sentimentr::get_sentences() %>%
                                           sentimentr::extract_emotion_terms() %>%
                                           attributes()

gamestop_short_squeeze_tweet_emotion_df$counts %>%
  distinct(words) %>%
  head()
```

```
##           words
## 1:         time
## 2:         good
## 3:        money
## 4:          ill
## 5:         love
## 6: manipulation
```

Highlights sentiment using sentimentr package.

```
reddit_wallstreetbets_clean_df$clean_comments %>%
  sentimentr::get_sentences() %>%
  sentimentr::sentiment_by() %>%
  sentimentr::highlight(file = "/Users/swaruprakshit/Documents/MSDS - Rutgers/Spring-
2021/16-954-597-01-DATA-WRANGLING/Final Project/final-project-submission/reddit_walls
treetbets_comment_sentimentr_package_highlight.html")

gamestop_short_squeeze_tweet_clean_df$clean_tweet %>%
  sentimentr::get_sentences() %>%
  sentimentr::sentiment_by() %>%
  sentimentr::highlight(file = "/Users/swaruprakshit/Documents/MSDS - Rutgers/Spring-
2021/16-954-597-01-DATA-WRANGLING/Final Project/final-project-submission/gamestop_sho
rt_squeeze_twitter_tweet_sentimentr_package_highlight.html")
```

Explanation:- Highlight files should have been created in current working directory. Reddit Wallstreetbets sentiment highlighted in reddit_wallstreetbets_comment_sentimentr_package_highlight.html. Twitter's tweet sentiment highlighted in gamestop_short_squeeze_twitter_tweet_sentimentr_package_highlight.html.