Twitter Sentiment Mining

An Analytical report on public sentiments for Samsung Mobile.

Muhammad Khizar Hayat Tahir

MGIS - 650

Sentiment Mining Project

Overview:

Manufacturers and service providers have always been concerned about how their products are perceived by their audience. Traditionally people have used surveys and feedback forms to understand the *sentiment* of public. With increasing usage of services globally, the traditional ways have become inconvenient and ineffective. We realize that most opinion is shared on the internet in an *unstructured* and informal way. **Text Mining** and **Sentiment Analysis** are modern methods to gauge public sentiment from unstructured data.

Objective:

To pick a specific company and use modern methods and tools to understand the sentiment public has for them on Twitter platform.

Executive Summary:

In this project, we will use R and its packages to perform Sentiment Analysis of **Samsung Mobile** using **Twitter** as our data source. We will try to answer questions like is the overall sentiment about Samsung positive or negative; when is the sentiment more positive or negative; what terms are people mostly using when referring to or discussing the company under discussion, etc. We will incorporate visualizations where they help to understand relationships better.

Sentiment analysis in this project will be performed in 3 steps:

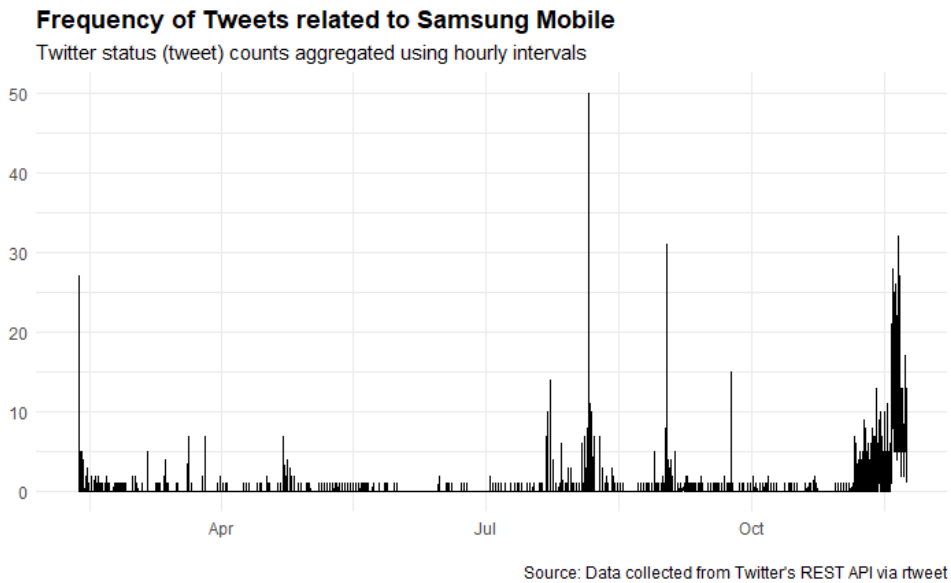
1. Collecting data from Twitter API
2. Performing Sentiment analysis
3. Text mining

STEP 1 - Collecting data from Twitter API:

As a first step, we fetch relevant data from Twitter which we will use in our analysis. We utilize publicly exposed Twitter API to get the following types of Tweets:

- Tweets from any source containing the keyword “Samsung Mobile”.
- Tweets made by the official Samsung mobile using their handle @SamsungMobile.
- Tweets made by the official Samsung support using their handle @SamsungSupport.

From each type, we select about 1000 tweets. We do not include retweets as a cleaning method to avoid duplication of results. We restrict our data to English language only. We combine the three types of tweets into a single data frame, and remove possible duplicates. Then, we do a frequency plot of tweets per hour as shown on the right:



Insight:

We can see that apart from some spikes, the number of tweets for most months has remained around 10 per

hour, **except** more recently when it clearly appears to be high – greater density of longer spikes near the end of October and in November. This could be due to:

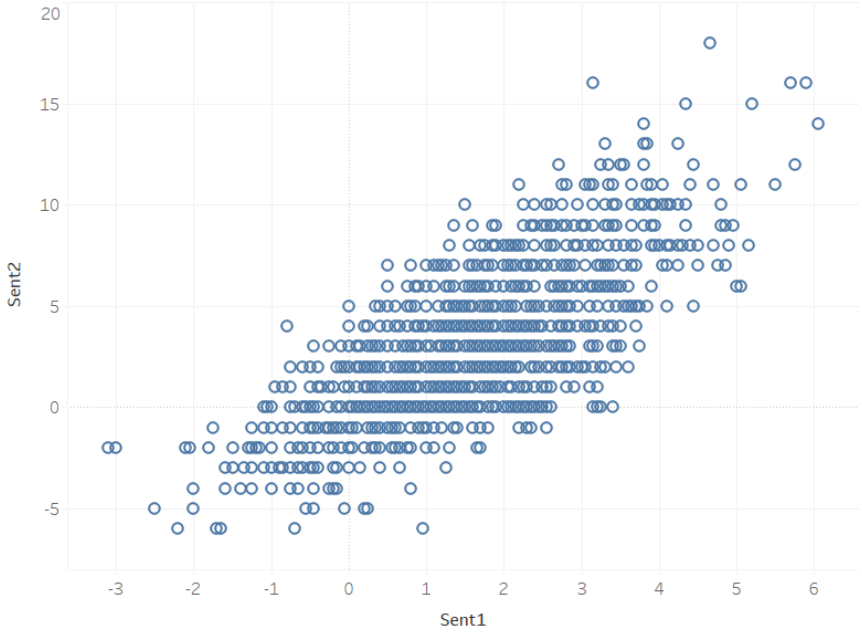
- The recent launch of new range of smartphones by Samsung.
- People who were the first to buy are likely to post reviews and/or complaints.
- The recent launch of new range of cell phones by their biggest competitor Apple fueling the forever old debate of which is better.

STEP 2 - Sentiment Analysis

2.1 Sentiment Scores

In this step, we use R package function on our dataset from Step 1 to calculate the **Sentiment score** for each tweet. This is a score given by the function signifying the emotion/opinion expressed in the tweet. If the sentiment is good, the score given will be positive and if it's bad, a negative score will be given. The better the sentiment, the more positive the score and vice versa.

We use two widely used methods for calculating sentiment scores to verify that the score is indeed correct or not. A comparison plot of the two sentiment scores is given on the right.



Sent1 vs. Sent2.

Insight:

We observe that the points show a **Positive** trend meaning both sentiment scores are positively correlated. Both assign scores to tweets similarly. These assure us that scores are indeed correct and any one method can be used for further analysis. We will use 'sent1' method for the rest.

2.2 Good Tweets

Next, we use our combined data from Step 1 and use R function to add a binary column (is_good) to it which is **True** if the tweet contains positive words: 'good', 'fast', 'reliable', 'trust', or 'support'; **False** otherwise.

Insight:

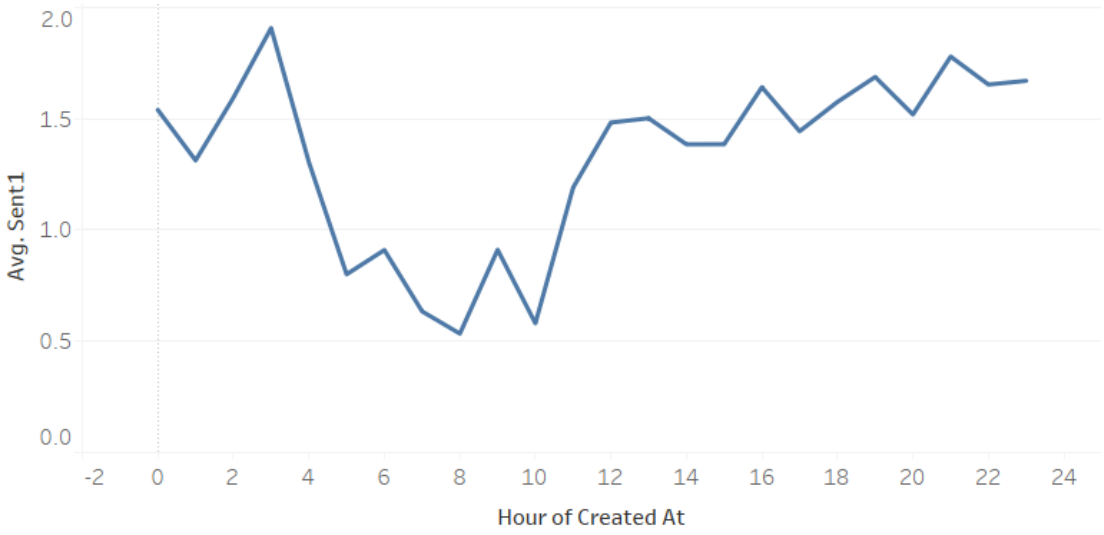
We calculate the total number of Tweets with **True** and see there are 435 tweets which is not bad! These are the tweets containing very positive words as mentioned above.

2.3 Sentiment with respect to time of day

We want to see if the sentiment score has any relation with *when* a tweet is made. We plot a graph of Average sentiment score against time of day in a 24-hour clock setting as shown below:

Insight:

We observe that sentiment score is low from 4 AM till 10 AM and generally high at other times. This could be due to:



- Maximum number of people are asleep at that time – low engagement.
- Although people are up by 7, they probably do not use social media until 10 AM (work maybe).
- Samsung Mobile official accounts refrain from news and new events at this time because of low interactivity. They publicize and market more during late hours, hence greater/more positive sentiment scores then.
- People get free to use Twitter as day goes by with maximum engagement late at night.
- At night, people use Twitter on their free will, they are likely to want to lift their mood and what they post reflects it as we see more positive sentiment scores late at night.

2.4 When do people favorite or retweet more?

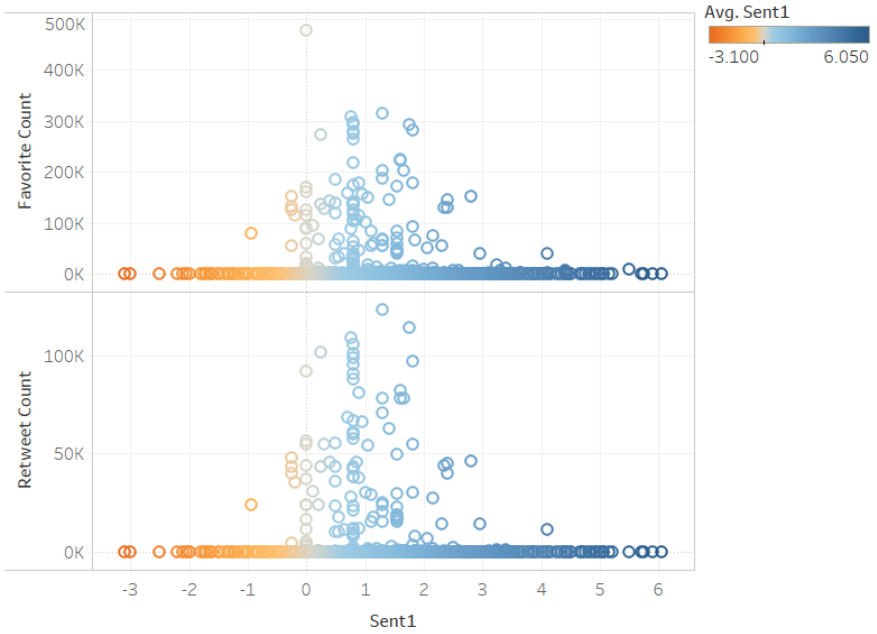
We determine which *kind* of tweets are most engaged with and shared by the audience. We plot two graphs: favorite count vs sentiment score, and retweet count vs sentiment score. Results below:

Insight:

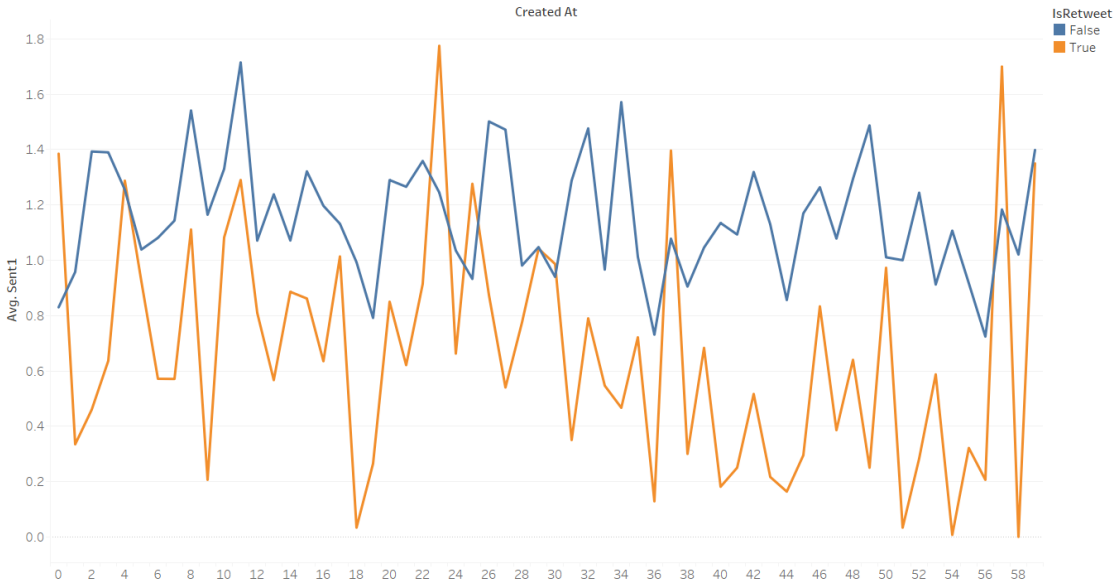
We observe that people tend to favorite or retweet more those tweets that are neutral and do not show any extreme emotion be it positive or negative.

This may be because people interact and share those tweets comfortably that are informative and sharing them does **Not** make them have a strong opinion on the matter. They may feel reluctant to share something very positive or negative as they may not want to take any specific hard stance or be a part of it.

We also observe from the color and key that most tweets have a positive sentiment score.



A graph below containing separate plots for retweeted or not-retweeted data reaffirms our observation above by showing that tweets with slightly positive sentiment score were retweeted and those with very high sentiment scores were not.



2.5 What is the General sentiment about Samsung Mobile.

The mean sentiment score from ALL tweets comes out as 1.1 meaning the general sentiment about Samsung mobiles from the sentiment scores **only** is slightly **Positive**.

2.6 Are Positive sentiments only by Android fans?

To determine and rule out this possibly biased attitude, we filter the data to only retain 3 sources of tweets:

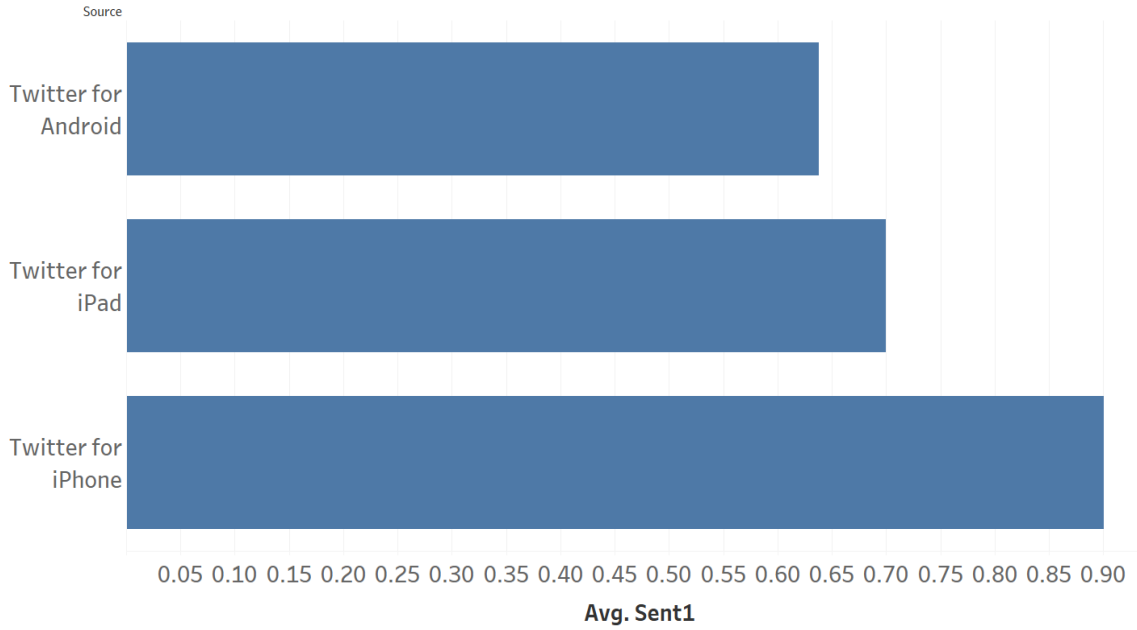
1. Twitter for Android
2. Twitter for iPad
3. Twitter for iPhone

We plot each of these categories against their respective **average** sentiment score.

Insight:

To our surprise, we observe that of these three categories, iPhone users' tweets had the Highest sentiment score for Samsung Mobile, followed by iPad users, and then Android.

This means that Samsung has actually done really well to not just keep their OS users positive about their product, but even the users of their rival mobile company, Apple.



STEP 3 – Text mining

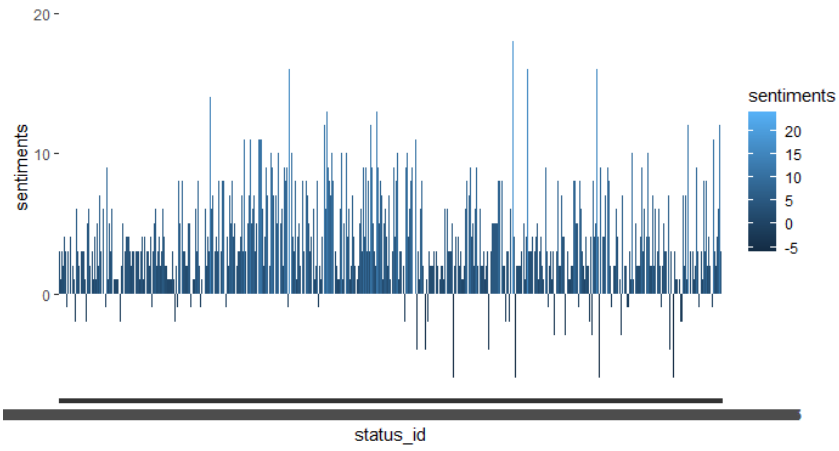
3.1 Visualizing sentiment scores of tweets further

We use a different technique to double-check our results. Instead of calculating sentiment scores for each full tweet, we break down the tweet into individual words (**tokens**) and do a lexicon analysis to give each word a score which we can later sum up for each tweet/status_id. Then, we plot to get an idea of the sentiment again.

Insight:

We again see that for most tweets, the sentiment scores are positive.

The extreme values on the negative side are less than the extreme values on the positive side too. And there are very few negative sentiment values.



3.2 Creating Wordcloud

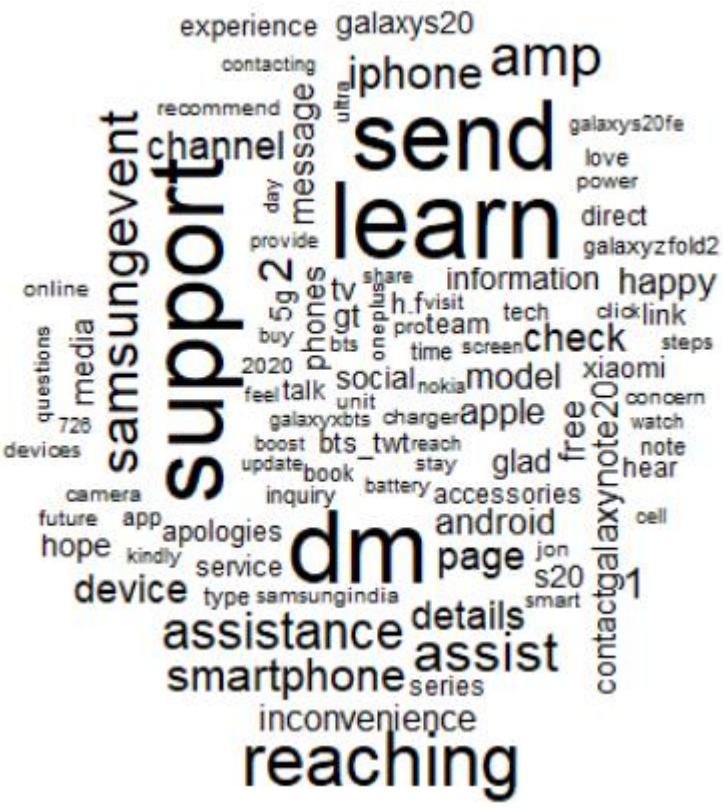
In this part, we create a wordcloud from our tokens. We count the occurrence of each word and create it. As a cleaning method, we maintain a stop-word list which does not get displayed. We include meaningless words into that list so our wordcloud only contains words that explain the sentiments of the audience as best as it can.

Insight:

The words with higher frequency are larger.

We notice that:

- Words like Support, Assistance, Learn, Reaching are the largest, meaning these positive terms were used the most. Hence it is not a surprise why the sentiment scores were generally positive.
- A lot of these words tell us that Samsung mobile has provided great support and assistance to their customers. Most words resonate with asking for help and Samsung resolving it
- We notice that competitor companies were also mentioned quite a few times like iPhone, Apple, Xiaomi, etc. This might be due to comparisons being made of these companies' products and services.



For further analysis, we create a wordcloud of word with a frequency range of 50 to 750 – a range we expect most popular words to be in; a way to remove words occasionally used. We use Color to depict the sentiment score with Negatives given Red, and Positives given Green. We center at the sentiment score of 0.

From the resulting plot below, we see that **ALL** the meaningful, frequently used words are Positive. The negative sentiment words do exist but their frequencies are very less and get dropped by our filter.

Conclusion

With all these analyses and visualizations, we conclude that Samsung Mobile has quite a **Positive overall sentiment** among public with Twitter as our data source. People used positive words for Samsung Mobile generally, even the users of competitor sell phones.

Through the procedures described and performed in this project, we demonstrated how Text Mining and Sentiment Analysis can be effectively used to determine what the people think and feel about a company or their product/services.

Appendix

```
# STEP 0 - Libraries


library(readxl)

library(rtweet)

library(dplyr)

library(ggplot2)

library(syuzhet)

library(magrittr)

library(wordcloud)

library(writexl)

library(tidytext)



# STEP 1 - Collecting data from Twitter API


searchdata <- search_tweets("samsung mobile", n = 1000, include_rts = FALSE, lang = "en")

View(searchdata)

samsung_tweets <- get_timeline(c("@SamsungMobile", "@SamsungSupport"), n = 1000)

View(samsung_tweets)


alldata <- rbind(searchdata,samsung_tweets)


dup <- duplicated(alldata$status_id)

table(dup) # Returned 0 duplicates so further subsetting not required but still done in case running later returns a duplicate.

alldata <- alldata[!duplicated(alldata), ]


View(alldata)

dim(alldata)
```

```
library(ggplot2)
alldata %>%
  ts_plot(by = "hour") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold")) +
  labs(
    x = NULL, y = NULL,
    title = "Frequency of Tweets related to Samsung Mobile",
    subtitle = "Twitter status (tweet) counts aggregated using hourly intervals",
    caption = "\nSource: Data collected from Twitter's REST API via rtweet"
  )


# STEP 2 - Sentiment Analysis

sent1 <- get_sentiment(alldata$text,method="syuzhet")
sent2 <- get_sentiment(alldata$text,method="afinn")


sent_data <- cbind(alldata, sent1, sent2)
names(sent_data)
head(sent_data)

attach(sent_data)
plot(sent1, sent2, type = "p")


#grepl() function searchs for matches of a string or string vector. It returns TRUE if a string contains the pattern, otherwise FALSE
is_good <- grepl("good|fast|reliable|trust|support",alldata$text,ignore.case=TRUE) #see if the text contains good words or not
```

```
finaldata <- cbind(sent_data,is_good)

View(finaldata)

good_data <- finaldata[finaldata[, "is_good"] == "TRUE",]

dim(good_data)

View(good_data)


# model <- lm(retweet_count ~ display_text_width + sent2 + is_good, data = finaldata) # Regression perfomed but was not of much use.

# summary(model)

sum(finaldata$sent1)

mean(finaldata$sent1)

write_xlsx(finaldata, "C:/RIT Courses/MGIS 650 - Data Analytics & BI/R/finaldata_samsung.xlsx")



# STEP 3 - Text Mining


attach(finaldata)

tokens <- finaldata %>%

  unnest_tokens(word, text)


samsung_sentiment <- tokens %>%

  inner_join(get_sentiments("afinn")) %>%

  group_by(status_id) %>%

  summarize(sentiments =  sum(value))


samsung_sentiment %>%

  ggplot() +

  geom_col(mapping = aes(status_id, sentiments,fill=sentiments))


tokens %>%

  anti_join(stop_words) %>%
```

```
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))

Df_newwords <- data.frame(c("https", "t.co", "samsung", "mobile", "galaxy", "x1ikdlbvk5", "phone"),
                          c("SMART", "onix", "SMART", "SMART", "SMART", "SMART", "SMART"))

# Naming the above data frame
names(Df_newwords) <- c("word", "lexicon")

# adding observations using rbind() function
new_stop_words <- rbind(stop_words, Df_newwords)

tokens %>%
  anti_join(new_stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100, scale = c(3,0.2)))

write_xlsx(tokens, "C:/RIT Courses/MGIS 650 - Data Analytics & BI/R/finaldata_samsung2.xlsx")
```