# Analyzing the Sentiment Towards Ridesharing Services and Public Transportation in Washington D.C.

GitHub: https://github.com/mbmk2020/FOCD2021_Final_Project

Mohammad Al-khasawneh

2021-12-13

## Contents

# Introduction

The growth of the population and their mobility needs require other alternative transportation systems such as ride-sharing services. The ride-sharing services are more promising for people's needs in terms of cost and time since they are app-based use smartphone technology in their functionality. They can be considered a potential solution to meet the needs of passengers in terms of flexibility and cost without expanding the service frequency of public transportation. Both of these ride services are highly demanding in the metropolitan where people tend to avoid driving or having their cars to avoid delays or parking difficulties accompanied by the dense population in the area. The first impression about these ride services is that Public transportation is usually cheaper than ride-sharing services. On the other hand, ride-sharing services may be faster and easier to access. The main objective of this research project is to answer these questions through a sentiment analysis for people's Twitter posts to extract their feelings about the available alternatives of public transportation and ride-sharing services. The Washington D.C. metropolitan area is selected as the geographical area for this study. Three services included Uber, Lyft, and the normal Taxi are selected to study people's opinions about ride-sharing services, whereas the Amtrak, Metro, and Metrobus were selected to study people's opinions about the public transportation services. Social media has become more popular and is widely used by people of all ages. Twitter is one of the most popular social media with millions of active users monthly. Many organizations, business companies specifically, use this social media to gain some feedback for their business.

# Motivation

The motivation for doing this research project is to answer the questions of what people prefer to use: public transportation or ridesharing services through a sentiment analysis using Twitter posts. Both Twitter data and sentiment analysis are motivated keys for the project. Several research studies have been done in the field of transportation engineering using sentiment analysis. Sonia et al. (2016) have used Twitter data to measure and compare customer satisfaction between two companies that provide online transportation services. They used the sentiment analysis technique and found that most of the Twitter posts were for bad experiences with these two companies. Another study by Eddendy et al. (2016) conducted sentiment analysis for Twitter posts about the use of public transportation in the big cities in Indonesia. These research studies have provided good examples of how Twitter data and sentiment analysis can be used in the field of transportation engineering. Therefore, I was motivated to do a comprehensive comparison to include both the public transportation and the ridesharing services in one comparison. Another reason that motivates me to choose twitter data is that people get the chance to speak more honestly there about their opinions.

# Installing Scripts

## Required packages

The following are the packages used in this research project. The “twitterR” package was used for search twitter data with specific geographical location and search terms. The “tidytext” and “dplyer” packages were used for data manipulation. Finally, the packages of “ggmap” and “gridExtra” were used for data visualization and plot organizations.

```r
install.packages(c("mnormt", "psych", "SnowballC",
                   "hunspell","broom", "tokenizers", "janeaustenr"))
install.packages("twitteR")
install.packages("tidytext")
install.packages("ggmap")
install.packages("tcltk")
install.packages("gridExtra")
install.packages("dplyr")
install.packages("kableExtra")
install.packages("sentimentr")
```

## Required libraries

The following are the specific libraries used in this project.

```r
library(twitteR)
library(tidyverse)
library(tidytext)
library(ggmap)
library(tcltk)
library(gridExtra)
library(scales)
library(dplyr)
library(kableExtra)
library(sentimentr)
```

## Setup the twitter API

In order to be able to use the twitter API the following are the keys and tokens requested through twitter developer website https://developer.twitter.com/en in order to be able to use the twitter API.

# Data

The first part of the project is data gathering. The data used in this study was mainly from twitter posts. The data obtained from twitter API using ‘twitterR’ package. The API provides the user with an access token that gives the user access to searching for tweets and information about them and stores them in a data frame. The data includes people twitter posts about the available public transportation: Amtrak, Metro, and Metrobus services and also the ridesharing services including: Uber, Lyft, and the normal cab Taxi. The obtained data covered mainly the area of Washington D.C. and up to Baltimore area as many of these ride services lives in Baltimore as well. Since The twitter API gives only access to the most recent 8 days, the data for this project was combined for four separate searches conducted on following dates: Nov 2nd, Nov 12, Nov 22, and Dec 2nd.

## Searching data using twitter API

The following is the code used to search for Twitter posts about each ride service:

```
##Ridesharing services
#Taxi <- twListToDF(searchTwitter("taxi", geocode = "38.907,-77.036,60mi",
#                                  n = 5000,lang="en"))
#Uber <- twListToDF(searchTwitter("Uber", geocode = "38.907,-77.036,60mi",
#                                  n = 5000,lang="en"))
#Lyft <- twListToDF(searchTwitter("Lyft", geocode = "38.907,-77.036,60mi",
#                                  n = 5000,lang="en"))
#Ridesharing_Services <- rbind(Taxi, Uber, Lyft)
##Public transportation
#Amtrak <- twListToDF(searchTwitter("amtrak", geocode = "38.907,-77.036,
#                                    60mi",n = 5000,lang="en"))
#metrobus <- twListToDF(searchTwitter("metrobus", geocode = "38.907,-77.036,
#                                      60mi",n = 5000,lang="en"))
#metro <- twListToDF(searchTwitter("metro", geocode = "38.907,-77.036,
#                                   60mi",n = 5000,lang="en"))
#Public_Transportation <- rbind(Amtrak, metro, metrobus)
```

## Combining data for one month history

The downloaded searches were stored at the local storage and then were combined into one dataframe for each ride service.

The following table summaries the number of twitter posts obtained for each ride service:

|  | 11.02.2021 | 11.12.2021 | 11.22.2021 | 12.02.2021 | Total |
|---|---|---|---|---|---|
| Uber | 1712 | 1475 | 1339 | 1322 | 5848 |
| Lyft | 564 | 360 | 335 | 282 | 1541 |
| Taxi | 319 | 444 | 415 | 254 | 1432 |
| Amtrak | 354 | 371 | 363 | 243 | 1331 |
| Metro | 2418 | 1763 | 1687 | 1624 | 7492 |
| Metrobus | 47 | 23 | 25 | 21 | 116 |

# Data Analysis

After the required data was gathered and stored in data frames. The data was cleaned out from the uninformative words through some tokenization steps. Different functions were created to remove "stop" words such as "and", "the", "of", "or" as well as twitter specific words such as "rt" etc. The function also splits the tweets into individual words and removes all hashtags and signs.

After tokenizing and cleaning the data, the tweets were fed to specific functions of each package. For the first packages "tidytext", the function takes the tweets and returns a score for each tweet. This score is then classified according to a function coded into positive and negative and their the score which represent the repetitive of each word.

Finally, a data table is created from the positive and negative classification and a graph is created to visualize the results. The following section will have more detailed description for each step in the data analysis.

## Data Pre-proccessing and Tokanization

The following code is for data cleaning and tokenizing:

```r
Tokenization_fun <- function(df){
  df$text = gsub("(f|ht)(tp)(s?)(://)(.*)[.|/](.*)", " ", df$text)

  #removing link
  df$text = gsub("(f|ht)(tp)(s?)(://)(.*)[.|/](.*)", " ", df$text)

  # removing hashtags
  df$text = gsub("#\\w+", " ", df$text)

  # removing @people
  df$text = gsub("@\\w+", " ", df$text)

  #removing punctuations
  df$text = gsub("[[:punct:]]", " ", df$text)
  #removing numbers
  df$text = gsub("[[:digit:]]", " ", df$text)

  #removing emojis
  df$text <- str_replace_all(df$text,"[^[:graph:]]"," ")
  df$text <- str_replace_all(df$text,’https’," ")
  df$text <- str_replace_all(df$text,’amp’," ")

  #removing spaces
  df$text = gsub("[ \t]{2,}", " ", df$text)
  df$text = gsub("^\\s+|\\s+$", "", df$text)

  return(df)
}

Taxi.tweets <- Tokenization_fun(Taxi.tweets)
Uber.tweets <- Tokenization_fun(Uber.tweets)
Lyft.tweets <- Tokenization_fun(Lyft.tweets)
RSS.tweets <- rbind(Taxi.tweets, Uber.tweets, Lyft.tweets)


Amtrak.tweets <- Tokenization_fun(Amtrak.tweets)
metro.tweets <- Tokenization_fun(metro.tweets)
metrobus.tweets <- Tokenization_fun(metrobus.tweets)
PT.tweets <- rbind(Amtrak.tweets, metro.tweets, metrobus.tweets)
```

After data cleaning and tokenizing. The tweets were passed into unnset_tokens() function in order to split them into individual words using:

```
#Ride Sharing Services
Taxi.tweets_stem <-
  Taxi.tweets %>% select(text)%>%unnest_tokens(word, text) %>%
  anti_join(stop_words)
Uber.tweets_stem <-
  Uber.tweets %>% select(text)%>%unnest_tokens(word, text) %>%
  anti_join(stop_words)
Lyft.tweets_stem <-
  Lyft.tweets %>% select(text)%>%unnest_tokens(word, text) %>%
  anti_join(stop_words)
RSS.tweets_stem <- rbind(Taxi.tweets_stem,Uber.tweets_stem,Lyft.tweets_stem)

#Public Transportation
Amtrak.tweets_stem <-
  Amtrak.tweets %>% select(text)%>%unnest_tokens(word, text) %>%
  anti_join(stop_words)
metro.tweets_stem <-
  metro.tweets %>% select(text)%>%unnest_tokens(word, text) %>%
  anti_join(stop_words)
metrobus.tweets_stem <-
  metrobus.tweets %>% select(text)%>%unnest_tokens(word, text) %>%
  anti_join(stop_words)
PT.tweets_stem <- rbind(Amtrak.tweets_stem, metro.tweets_stem,
                        metrobus.tweets_stem)
```

In order to check whether the collected searches have included some relevant words to transportation or not, the most common words in people's twitter posts about ridesharing services and public transportation were counted and ploted as shown in Figure 1.

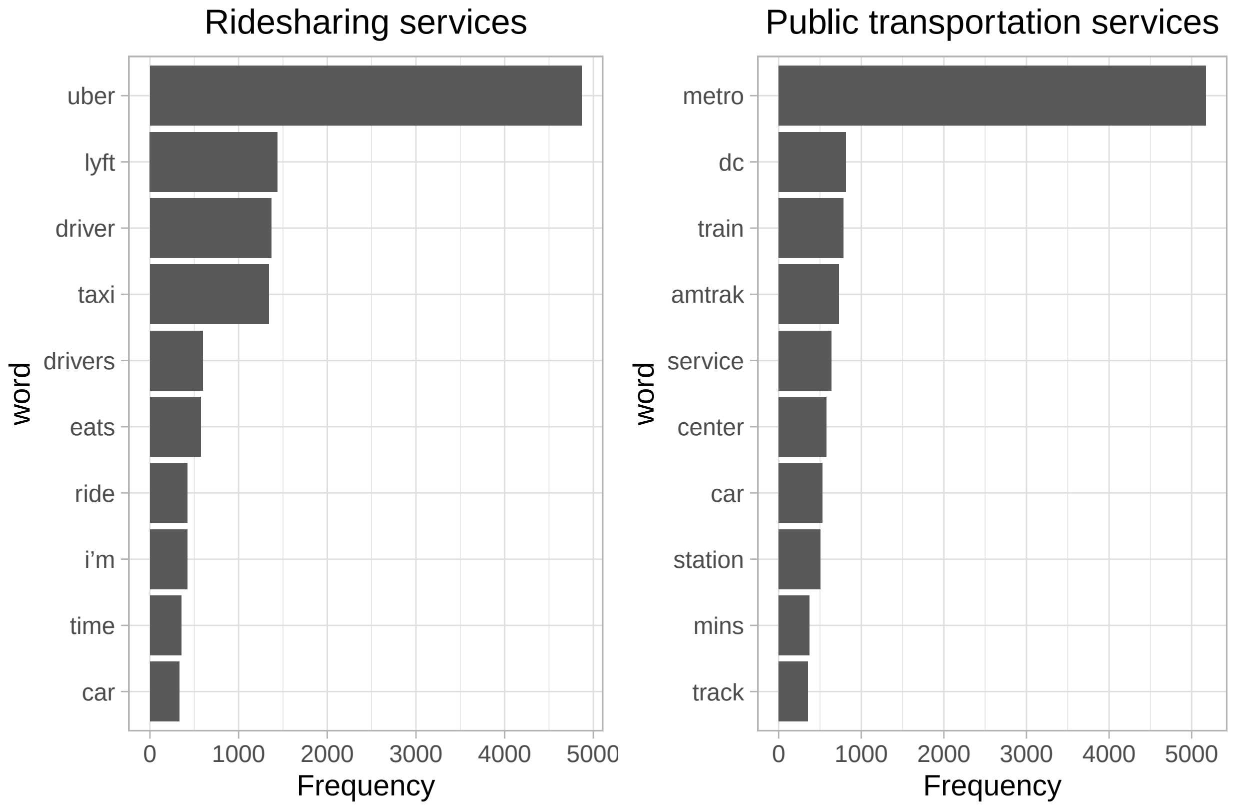

Figure 1. Most common words

## Sentiment lexicons using "tidytext"

There are several ways for evaluating the opinion or emotion in text. The sentiment lexicons provides analysis at the word level by splitting the whole sentence into individual words and then assigns words into different categories or feelings with specific scores. The sentiment lexicons can be done using tidytext package in R. There are also several lexicons such as: bing from (Bing Liu and collaborators, 2004), ncr by (Saif Mohammad and Peter Turney,

2010) , and AFINN from ( Finn Årup Nielsen, 2011). The bing sentiment analysis will be used for this research project for doing the analysis at the word level.

Bing Sentiment Analysis The bing lexicon categorizes words into positive and negative feelings. The get_sentiment(“bing”) from (2016) the function was used to obtain the sentiment dictionary for the list of words prepared by (Bing Liu and collaborators, 2004). Then the inner_hoin() was used to match each word in our data set with the appropriate feelings and sentiment score (-1 or 1). The function sentiment_score_bing() was prepared for this project to to calculate the sentiment score and feeling for each word by by passing the list of cleaned words from twitter posts. Finally each word in our dataset about each ride service will were assigned to “Positive” or “Negative” feelings according to the bing dictionary. This was done using following code:

```r
sentiment_score_bing <- function(tweets_stem){
  tweets_stem %>%
  inner_join(get_sentiments("bing")) %>%
  mutate(value = case_when(
    sentiment=="negative"~-1,
    sentiment=="positive"~1
  ))%>%
  count(word, sentiment,value, sort = TRUE) %>%
  ungroup()
}

Uber.tweets_Sentiment <- sentiment_score_bing(Uber.tweets_stem)
Lyft.tweets_Sentiment <- sentiment_score_bing(Lyft.tweets_stem)
Taxi.tweets_Sentiment <- sentiment_score_bing(Taxi.tweets_stem)
RSS.tweets_Sentiment <- sentiment_score_bing(RSS.tweets_stem)

Amtrak.tweets_Sentiment <- sentiment_score_bing(Amtrak.tweets_stem)
metro.tweets_Sentiment <- sentiment_score_bing(metro.tweets_stem)
metrobus.tweets_Sentiment <- sentiment_score_bing(metrobus.tweets_stem)
PT.tweets_Sentiment <- sentiment_score_bing(PT.tweets_stem)
```

The following Figure 2 and Figure 3 show the most common positive and negative words in the collected twitter posts about the ridesharing services and the public transportation, respectively.

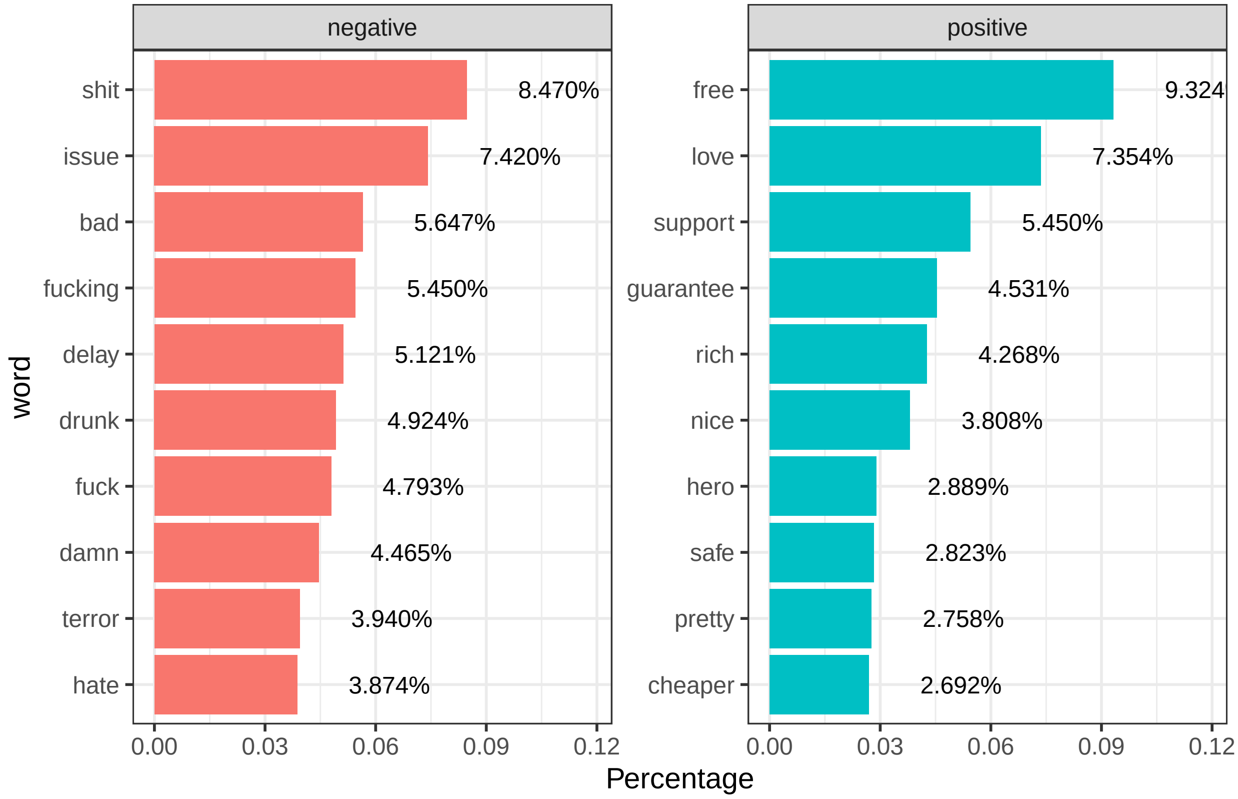

Figure 2. Common negative and Poitive words for the Ridesharing services

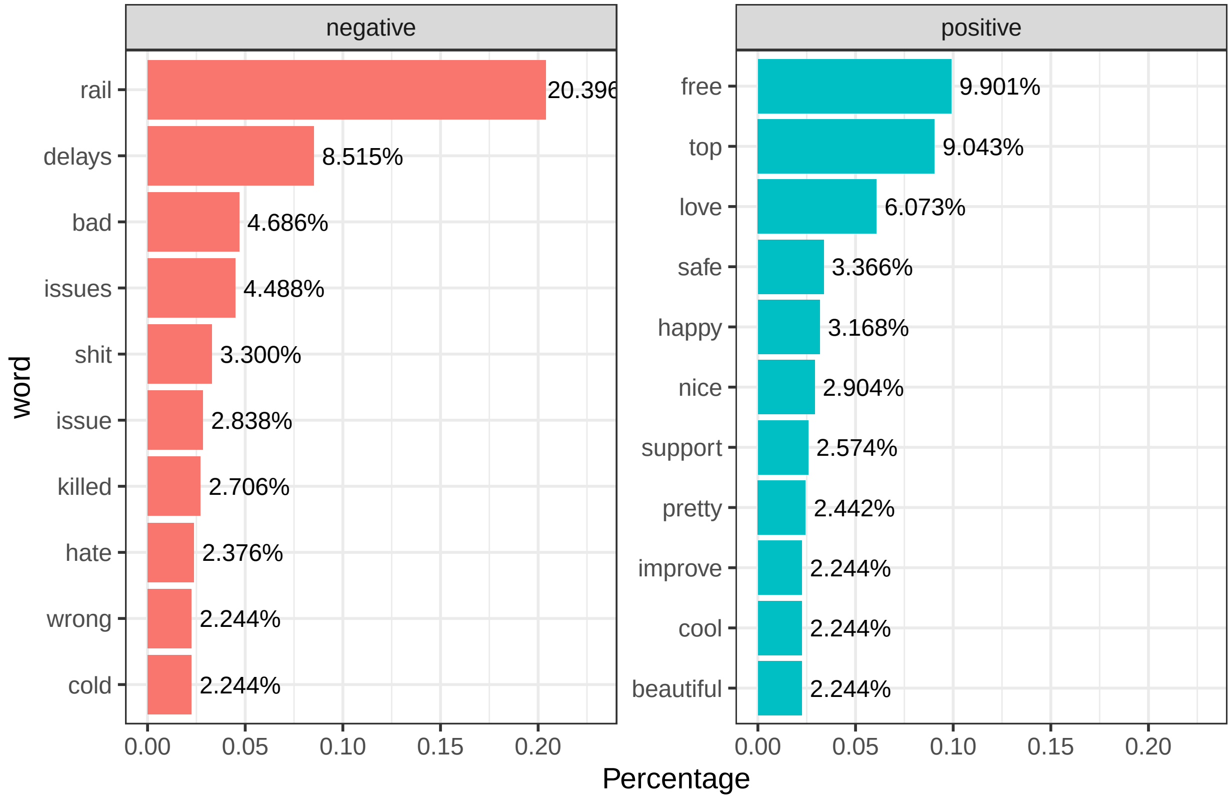

Figure 3. Common negative and Poitive words for the Public Transporatio

The next step is to count up the positive and negative words for each ride service. Then the percentage of each category was calculated out of the total number of words. The following code shows the total_sentiment_score() function used to calculate the percentages of the positive and negative opinions about the ride service in the collected twitter posts:

```
total_sentiment_score <- function(tweets_sentiments, service_name){
  tweet_table<-tweets_sentiments %>%
    mutate(score = n*value)
  sent.score_positive = case_when(
    nrow(tweet_table)==0~0,
    nrow(tweet_table)>0~sum(tweet_table[tweet_table$score>0,]$score)
  )
  sent.score_negative = case_when(
    nrow(tweet_table)==0~0,
    nrow(tweet_table)>0~sum(tweet_table[tweet_table$score<0,]$score)
  )
  positive_percent = sent.score_positive/(sent.score_positive+
                                           abs(sent.score_negative))
  negative_percent = abs(sent.score_negative)/(sent.score_positive+
                                                abs(sent.score_negative))
  results <- data.frame(c(service_name),c("Positive","Negative"),
                        c(positive_percent,negative_percent))
  colnames(results) <- c("Service","Sentiment","Sentiment_Score")
  return(results)
}
```

The sentiment analysis at the word level was performed for each ride service in order to have individual comparison for each of these services. Moreover, to have more aggregated comparison, the six services were reduced to four groups as the following: the Uber and Lyft services were grouped into one single group of App_Based Taxi, and the Amtrak and the Metro were grouped as Train transportation mode. The Figures 4 and 5 illustrate the results from the bing sentiment analysis for the six different services and the aggregated four groups.

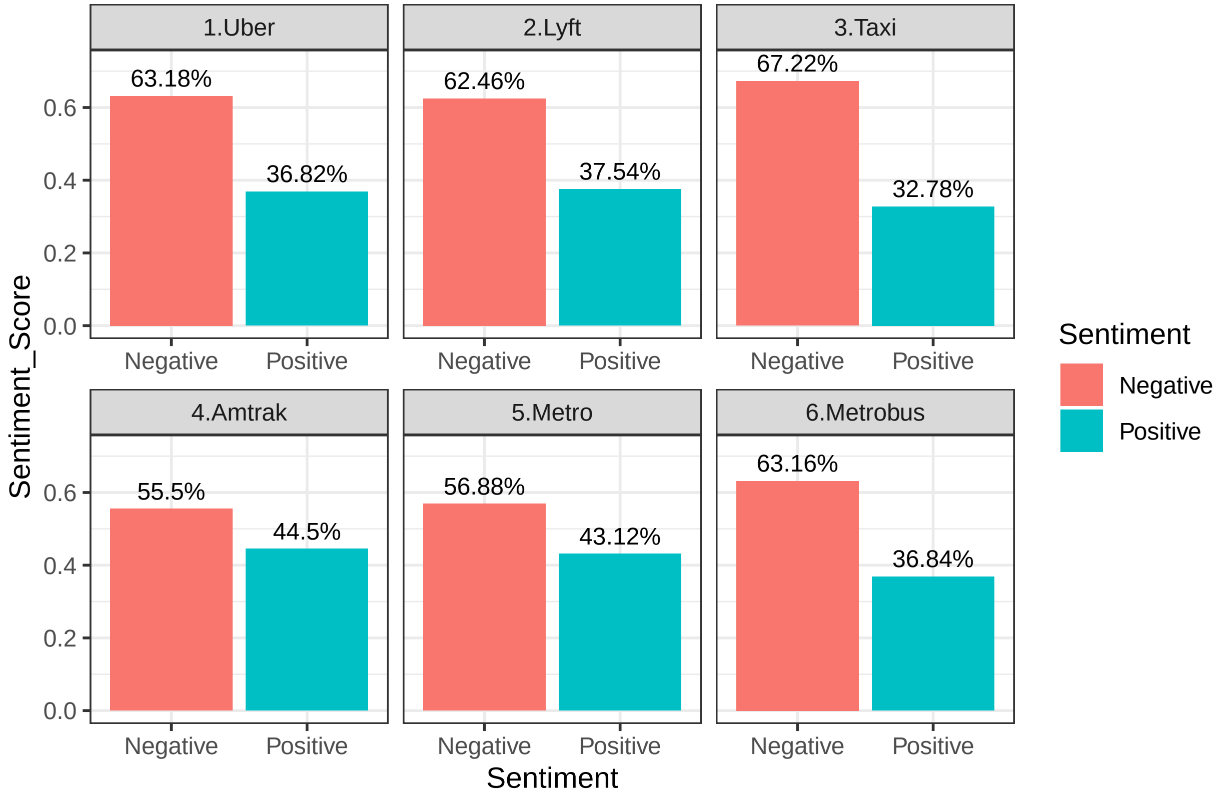

Figure 4. Sentiment Analysis at the word level (6 categories)

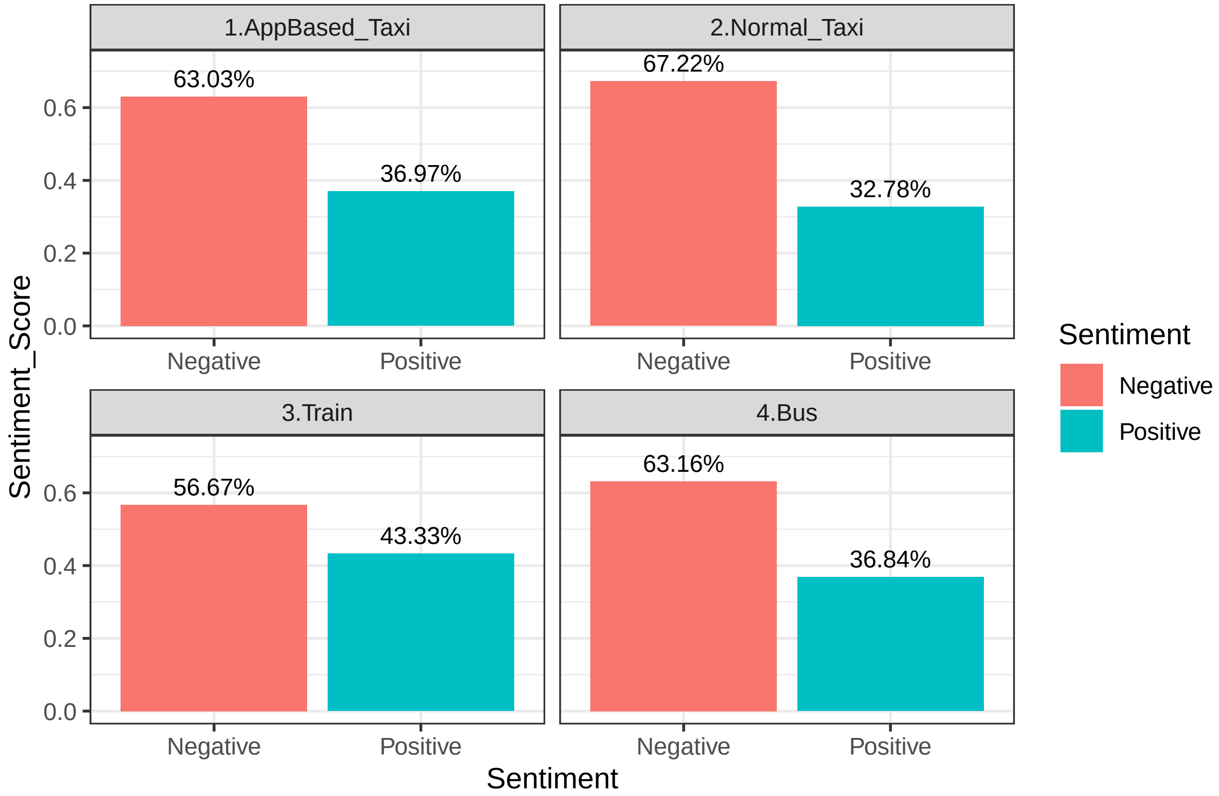

Figure 5. Sentiment Analysis at the word level (4 categories)

## Text Ploarity Analysis using sentimentR

For the same six groups and the four groups defined before. The sentiment analysis was also done at the whole text level using the (2021) package. The full text or the twitter post (after cleaning) was considered in the analysis instead of individual words. This can correct any inversion problem in the text. For example the bing sentiment would judge "I do not hate using Uber" as negative due the "hate" word, where in fact it is a positive opinion.

```r
Text_Polarity_Sentiment <- function(cleaned_data, service_name) {
  sentiment_by(cleaned_data$text) %>%
  mutate(Sentiment=case_when(
    ave_sentiment==0~"Natural",
    ave_sentiment<0~"Negative",
    ave_sentiment>0~"Positive"
  )) %>%
  select(ave_sentiment,Sentiment) %>%
  filter(Sentiment%in%c("Positive","Negative")) %>%
  group_by(Sentiment) %>%
  summarise(totals = sum(ave_sentiment), count = n())%>%
  mutate(Sentiment_Score = count/sum(count)) %>%
  select(Sentiment_Score, Sentiment) %>%
  mutate(Service = service_name)
}
```

The Figures 6 and 7 show the results from the sentiment analysis at the text level using sentimentr package for the six different services and the aggregated four groups.

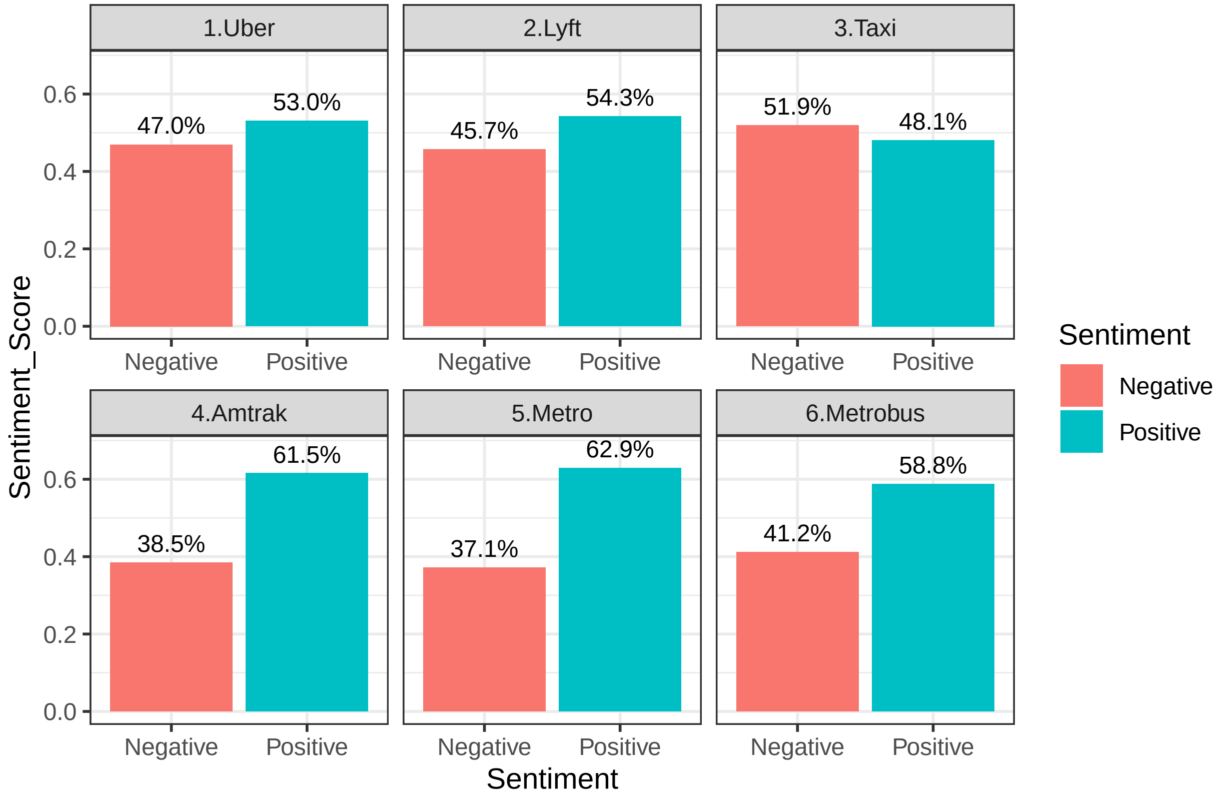

Figure 6. Sentiment Analysis per the whole text (6 categories)

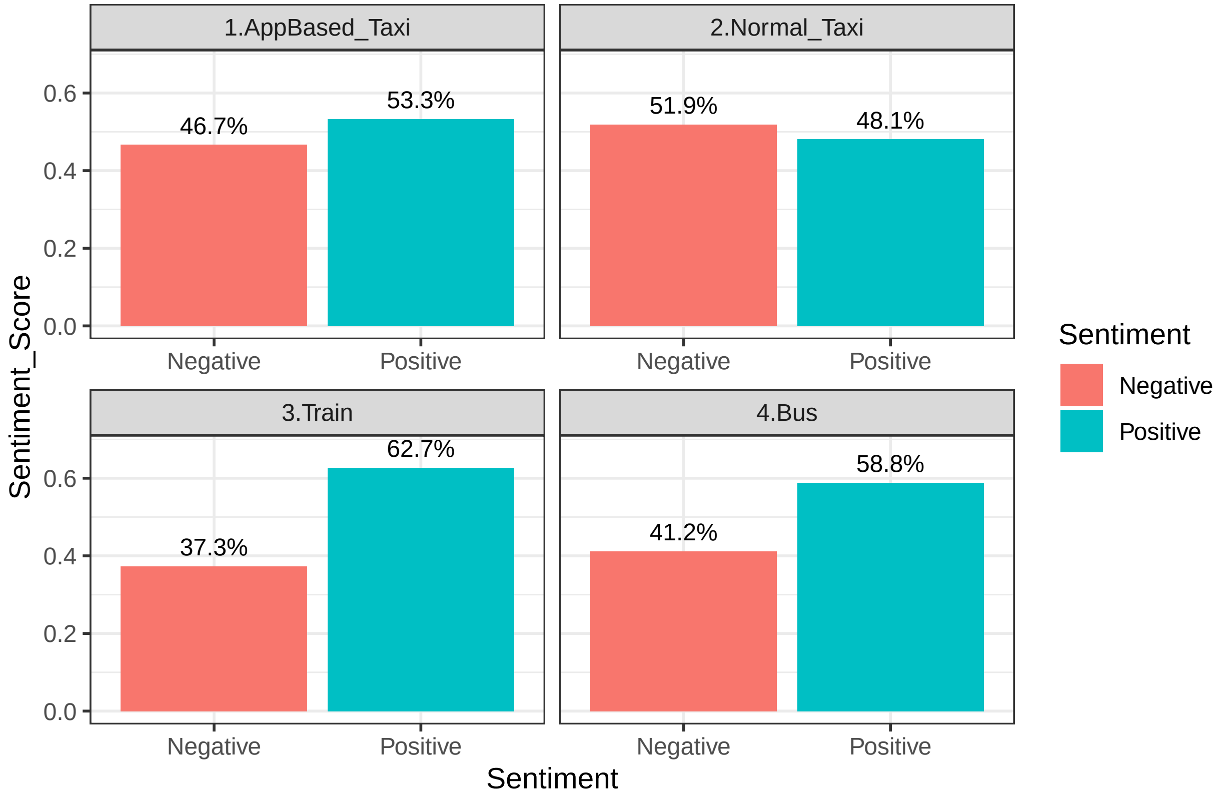

Figure 7. Sentiment Analysis per the whole text (4 categories)

# Results

In general, the sentiment analysis at the word level (Figures 4 and 5) did not show any strong opinion towrd any of the feelings (positive or negative) and had slightly differences in the percentages among the different services. However, the results showed that the App based services (Uber & Lyft) had more positive opinions than the normal cab taxi. Similarly, the Metro and Amtrak results had higher percentages of the positive feelings than the Metrobus. Moreoever, the results showed more negative opinions toward the provided services with no significant differences in the percentages among all the services.

On the other hand, the sentiment analysis at the word level showed that the twitter posts had more likely positive opinions toward these ride services expect for the normal cab taxi as shown in Figures 6. Overall, the relative comparisons between the services was noticed to be the same or very close between the two method of the sentiment analysis. For example, the normal cab taxi was always accompanied with the highest negative opinions and the lowest positive opinions. Therefore, the services were ranked from the most likely service to the lowest based on the positive percentages results from both methods. It was found that both of the two methods had captured very similar ranks for the selected services. Figures 8 and 9 summaries the ranks for each service or a group of service for each sentiment analysis method.

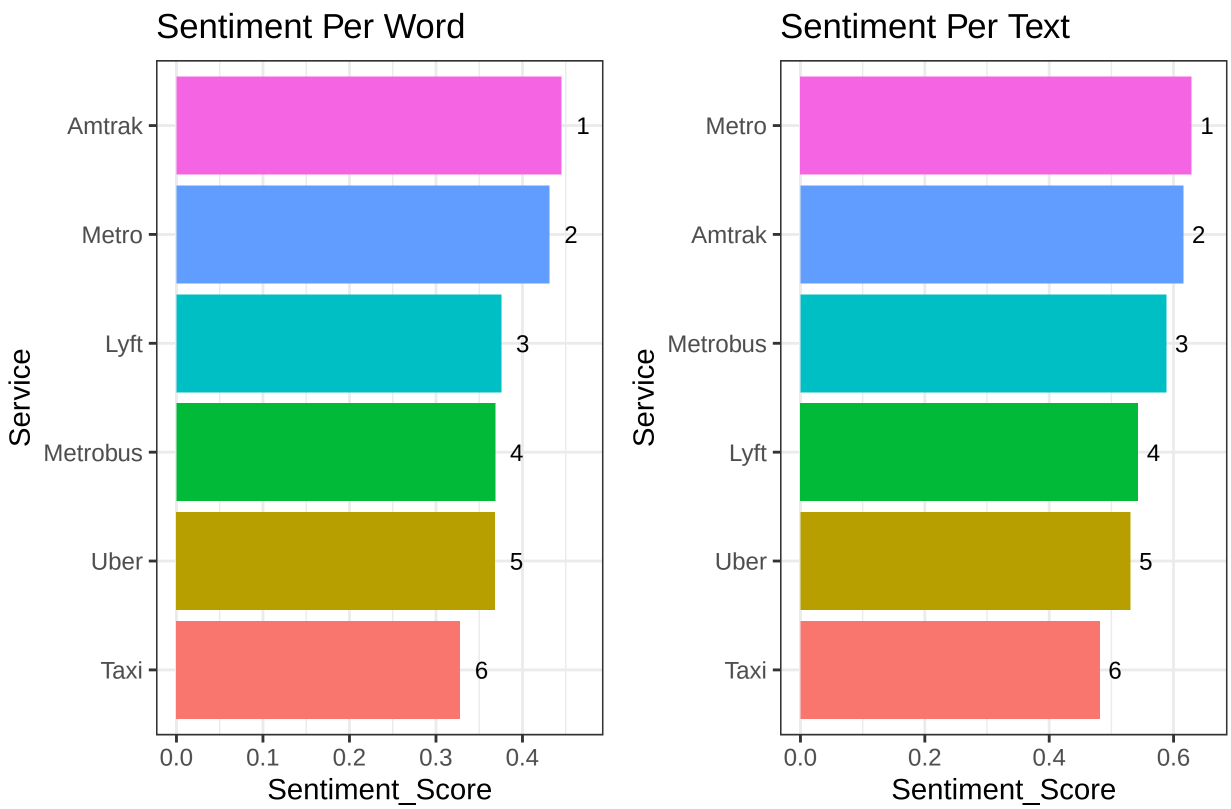

Figure 8. Ride services in D.C. from the most prefered to the least

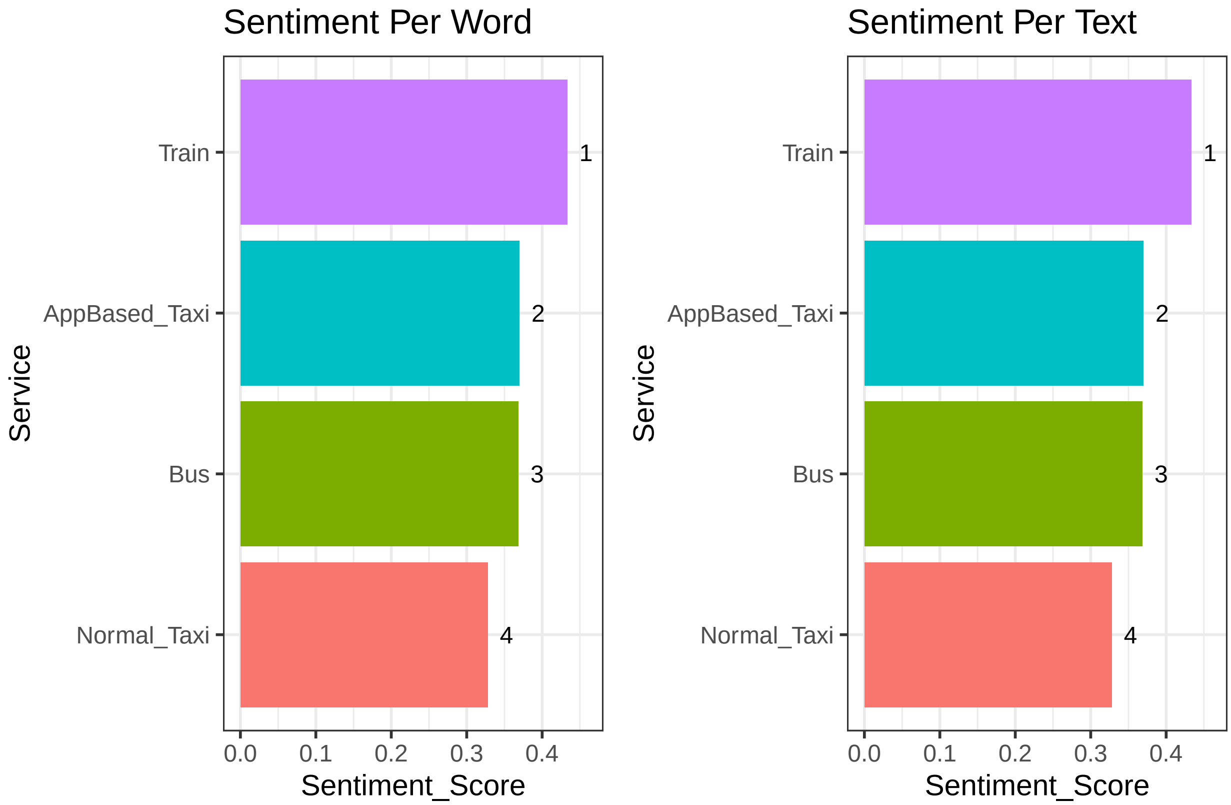

Figure 9. Ride service from the most prefered to the least

# Conclusion

This research study involves sentiment analysis towards the use of public transportation and ride-sharing services in the Washington D.C area. The data was collected through Twitter API for November 2021 (11-02-2021 to 12-02-2021). The collected data included the entire D.C. area and some parts of Baltimore since several people live there and work in D.C. The study included several public transportations such as Amtrak, Metrobus, and the Metro and some common ridesharing services including Uber, Lyft, and the normal cab-taxi. The sentiment analysis was done at the word level using the "tidytext" package and per the whole text using the "bing" dictionary. The results showed that both of the methods can be used to measure the same relative comparison between the service. However, the two methods had opposite opinions about each service.

# References

Rinker, Tyler W. 2021. *sentimentr: Calculate Text Polarity Sentiment*. Buffalo, New York. https://github.com/trinker/sentimentr.

Silge, Julia, and David Robinson. 2016. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in R." *JOSS* 1 (3). https://doi.org/10.21105/joss.00037.

Sonia Anastasia, Indra Budi. 2016. "Twitter Sentiment Analysis of Online Transportation Service Providers." Journal Article. https://ieeexplore.ieee.org/abstract/document/7872807.

Veronikha Effendy, Mira Kania Sabariah, Anita Novantirani. 2016. "Sentiment Analysis on Twitter About the Use of City Public Transportation Using Support Vector Machine Method." Journal Article. https://socj.telkomuniversity.ac.id/ojs/index.php/ijoict/article/view/85.