JLedesma_607_FinalProject

2022-11-30

Twitter Sentiment vs Stock Price

Loading Necessary Packages

```
require(httr)
require(jsonlite)
require(dplyr)
require(rtweet)
require(ggplot2)
require(dplyr)
require(tm)
require(wordcloud)
require(clarifai)
```

```
## Loading required package: clarifai
```

```
require(twitteR)
require(tidytext)
require(stringr)
library(tibbletime)
library(dplyr)
library(lubridate)
library(alphavantager)
```

Getting AlphaVantage API access

```
av_api_key('6KXR3Y47EGL14D1')
#av_get('BYND')
```

Retrieving stock data from AlphaVantage based on stock abbreviation. The data is then sorted and plotted in the interval of Dec 28 to Dec 2. The code below is for Microsoft but 4 other companies were also analyzed.

Microsoft Stock Data

```r
Microsoft_Stock<-av_get(symbol = "MSFT", av_fun = "TIME_SERIES_INTRADAY_EXTENDED", interval = "60min", 

Microsoft_Stock$Day<-day(as.POSIXlt.POSIXct(Microsoft_Stock$time))
Microsoft_Stock<-  filter(Microsoft_Stock, Microsoft_Stock$Day==28 | Microsoft_Stock$Day==29 | Microsof
Microsoft_Stock<-as.data.frame(Microsoft_Stock)
Microsoft_Stock_Plot<- as.data.frame(unique(Microsoft_Stock$Day))

Microsoft_Stock_Plot$DayAverage <- 1.0
for(i in (1:nrow(Microsoft_Stock_Plot))){
  Microsoft_Stock_Plot[i,2] <- mean(Microsoft_Stock[Microsoft_Stock$Day == Microsoft_Stock_Plot[i,1], '
}

Microsoft_Stock_Plot %>% arrange(factor(`unique(Microsoft_Stock$Day)`,levels =c(29,30,1,2,3)))
Microsoft_Stock_Plot$day = c(1,2,3,4,5)
plot(Microsoft_Stock_Plot$DayAverage~Microsoft_Stock_Plot$day,type='l', xlab = "Day", ylab=("Average St
```

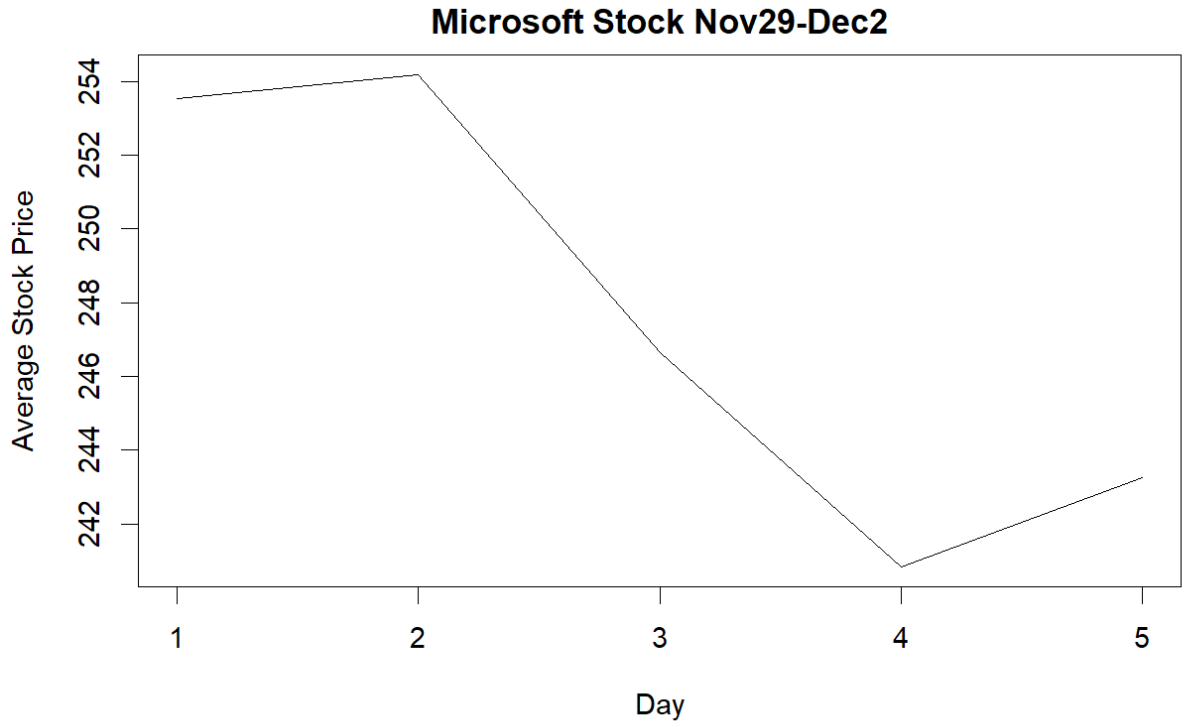

Figure 1: Microsoft

Tesla Stock Data

```r
#Tesla_Stock<-av_get(symbol = "TSLA", av_fun = "TIME_SERIES_INTRADAY_EXTENDED", interval = "60min", out

Tesla_Stock$Day<-day(as.POSIXlt.POSIXct(Tesla_Stock$time))
Tesla_Stock<-  filter(Tesla_Stock, Tesla_Stock$Day==28 | Tesla_Stock$Day==29 | Tesla_Stock$Day==30 | Tes
Tesla_Stock<-as.data.frame(Tesla_Stock)
```

```
Tesla_Stock_Plot<- as.data.frame(unique(Tesla_Stock$Day))

Tesla_Stock_Plot$DayAverage <- 1.0
for(i in (1:nrow(Tesla_Stock_Plot))){
  Tesla_Stock_Plot[i,2] <- mean(Tesla_Stock[Tesla_Stock$Day == Tesla_Stock_Plot[i,1], 'close'])
}

Tesla_Stock_Plot %>% arrange(factor(`unique(Tesla_Stock$Day)`,levels =c(29,30,1,2,3)))
Tesla_Stock_Plot$day = c(1,2,3,4,5)
plot(Tesla_Stock_Plot$DayAverage~Tesla_Stock_Plot$day,type='l', xlab = "Day", ylab=("Average Stock Price
```

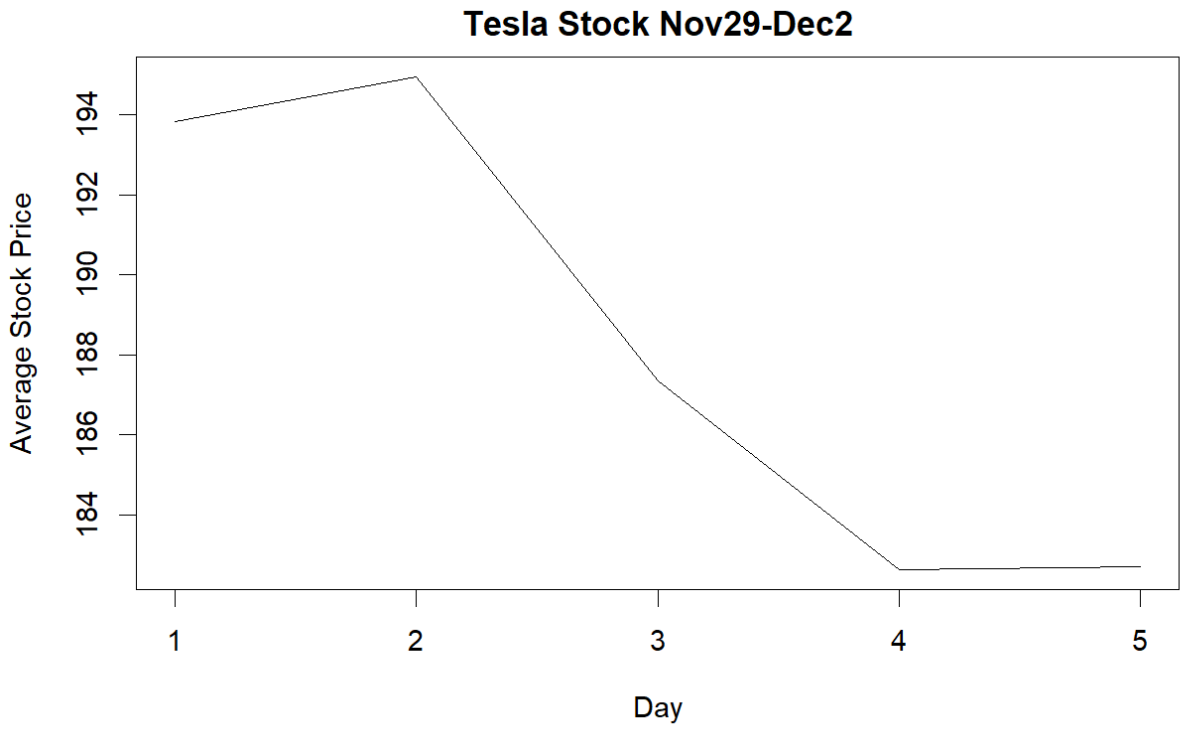

Figure 2: Tesla

Apple Stock Data

```
Apple_Stock<-av_get(symbol = "TSLA", av_fun = "TIME_SERIES_INTRADAY_EXTENDED", interval = "60min", outpu

Apple_Stock$Day<-day(as.POSIXlt.POSIXct(Apple_Stock$time))
Apple_Stock<-  filter(Apple_Stock, Apple_Stock$Day==28 | Apple_Stock$Day==29 | Apple_Stock$Day==30 | App
Apple_Stock<-as.data.frame(Apple_Stock)
Apple_Stock_Plot<- as.data.frame(unique(Apple_Stock$Day))

Apple_Stock_Plot$DayAverage <- 1.0
for(i in (1:nrow(Apple_Stock_Plot))){
  Apple_Stock_Plot[i,2] <- mean(Apple_Stock[Apple_Stock$Day == Apple_Stock_Plot[i,1], 'close'])
```

```
}

Apple_Stock_Plot %>% arrange(factor(`unique(Apple_Stock$Day)`,levels =c(29,30,1,2,3)))
Apple_Stock_Plot$day = c(1,2,3,4,5)
plot(Apple_Stock_Plot$DayAverage~Apple_Stock_Plot$day,type='l', xlab = "Day", ylab=("Average Stock Price
```

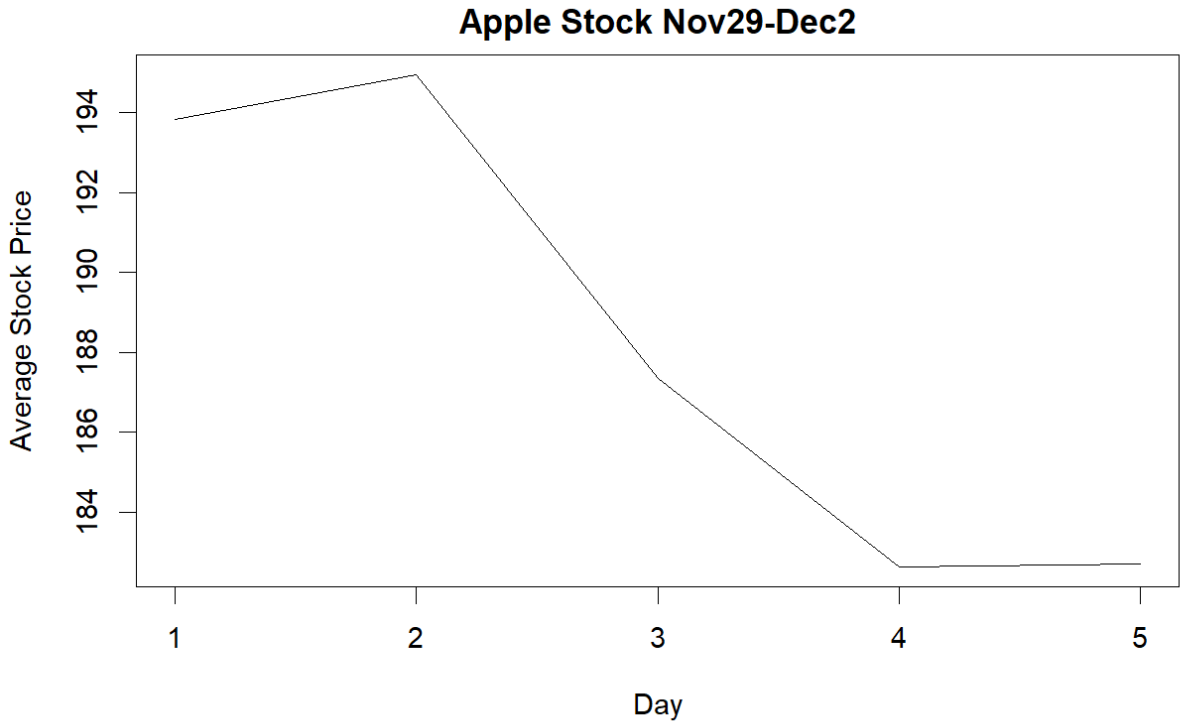

Figure 3: apple

Target Stock Data

```
Target_Stock<-av_get(symbol = "TGT", av_fun = "TIME_SERIES_INTRADAY_EXTENDED", interval = "60min", outpu

Target_Stock$Day<-day(as.POSIXlt.POSIXct(Target_Stock$time))
Target_Stock<-  filter(Target_Stock, Target_Stock$Day==28 | Target_Stock$Day==29 | Target_Stock$Day==30
Target_Stock<-as.data.frame(Target_Stock)
Target_Stock_Plot<- as.data.frame(unique(Target_Stock$Day))

Target_Stock_Plot$DayAverage <- 1.0
for(i in (1:nrow(Target_Stock_Plot))){
  Target_Stock_Plot[i,2] <- mean(Target_Stock[Target_Stock$Day == Target_Stock_Plot[i,1], 'close'])
}

Target_Stock_Plot %>% arrange(factor(`unique(Target_Stock$Day)`,levels =c(29,30,1,2,3)))
Target_Stock_Plot$day = c(1,2,3,4,5)
plot(Target_Stock_Plot$DayAverage~Target_Stock_Plot$day,type='l', xlab = "Day", ylab=("Average Stock Pri
```

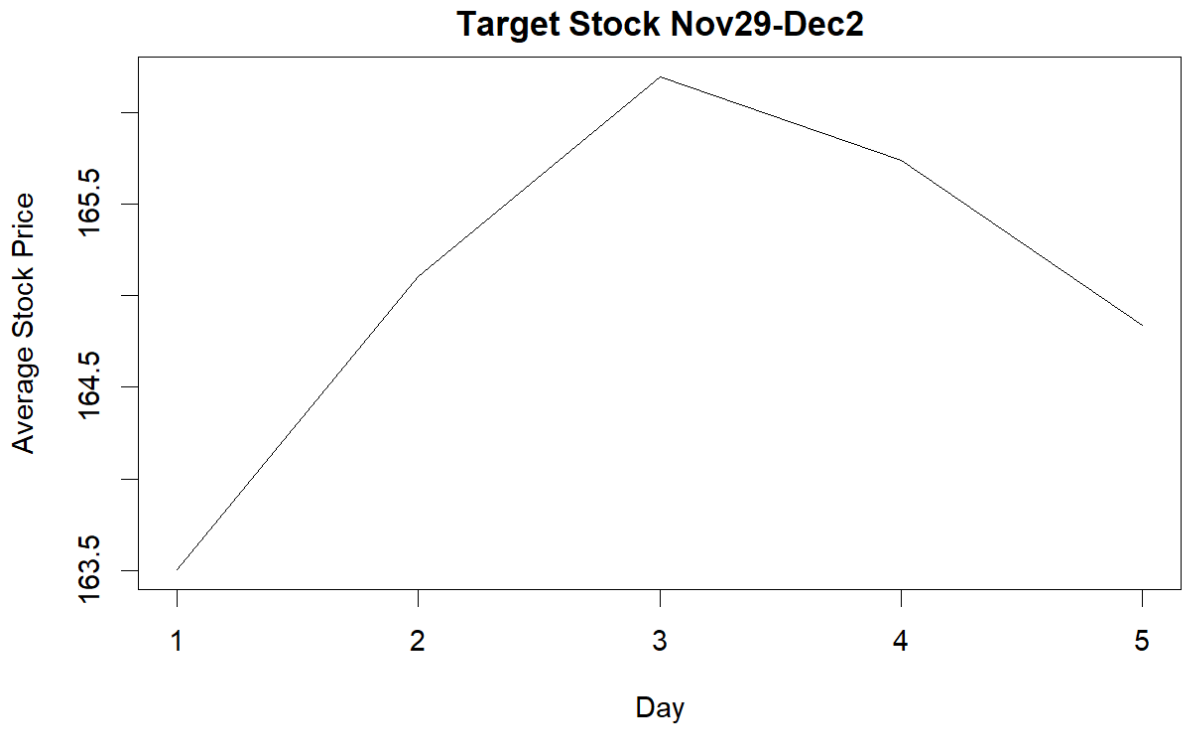

Figure 4: Target

Loading tweet data required me applying for a twitter developer account to which I was given access keys to the twitter API. I

```
bearer_token <- "AAAAAAAAAAAAAAAAAAAAAL2gjwEAAAAACuACbul%2BLX4asrb%2FwiPmOWf1trw%3DOOWhnmZqH6eJneMMDOvV
```

```
auth <- rtweet_app()
```

Reading raw tweets data

```
TSLA_Tweets_raw <- search_tweets(q = "#Doge", n = 40000, lang ="en")
```

```
Apple_Tweets_raw <- search_tweets(q="#AAPL",n =40000, lang="en")
```

```
Target_Tweets_raw <- search_tweets(q="#TGT",n =40000, lang="en")
```

```
MSFT_Tweets_raw <- search_tweets(q="#MSFT",n =40000, lang="en")
```

Function to clean tweets, using the afinn sentiment library and placing all data into a dataframe.

```
FindSentiment <- function(raw){

clean <- raw[,c(1,4)]
cleandf <- as.data.frame(clean)




cleandf$full_text <- removeNumbers(cleandf$full_text)
cleandf$full_text <- removePunctuation(cleandf$full_text)
cleandf$full_text <- tolower(cleandf$full_text)
cleandf$full_text <- removeWords(cleandf$full_text,stopwords())
cleandf$full_text <- gsub("[^\x01-\x7F]", "", cleandf$full_text)


one <- summarise(cleandf,text=paste(full_text,collapse = " "))
one <- strsplit(one[1,1], " ")[[1]]
one <- as.data.frame(one)
one <- one[nzchar(one$one),]
one <- as.data.frame(one)

names(one)[1]<-"word"

sentimentdf<-merge(x = one, y = afinn, by = c('word') )# Equivalent

sentimentScore <- sum(sentimentdf$value)


return(sentimentScore)
}
```

Raw data is read into the function, cleaned and found sentiments through the outside function call. The data is then placed in another data frame that is averaged by the 5 day interval of Nov 28 - Dec 2. A plot of this resulting data frame is then done.

```
MSFT_Sentiments <- MSFT_Tweets_raw[,c(1,4)]
MSFT_Sentiments[,3]= 0
names(MSFT_Sentiments)[3]<-"Sentiment"
MSFT_Sentiments$Day <- day(MSFT_Sentiments$created_at)

for (i in (1:nrow(MSFT_Sentiments))){
MSFT_Sentiments[i,3] <- FindSentiment(MSFT_Tweets_raw[i,])
}
```

```
MSFT_Sentiments_Plot <- as.data.frame(unique(MSFT_Sentiments$Day))
MSFT_Sentiments$DayAverage <- 1.0
MSFT_Sentiments_Plot<-as.data.frame(MSFT_Sentiments_Plot)
MSFT_Sentiments <- as.data.frame(MSFT_Sentiments)
for(i in (1:nrow(MSFT_Sentiments_Plot))){
  MSFT_Sentiments_Plot[i,2] <- mean(MSFT_Sentiments[MSFT_Sentiments$Day == MSFT_Sentiments_Plot[i,1], '
}

MSFT_Sentiments_Plot
MSFT_Sentiments_Plot<- arrange(MSFT_Sentiments_Plot,`unique(MSFT_Sentiments$Day)`)
MSFT_Sentiments_Plot<-head(MSFT_Sentiments_Plot,5)
plot(MSFT_Sentiments_Plot$V2~MSFT_Sentiments_Plot$`unique(MSFT_Sentiments$Day)`,type='l', xlab = "day",
```

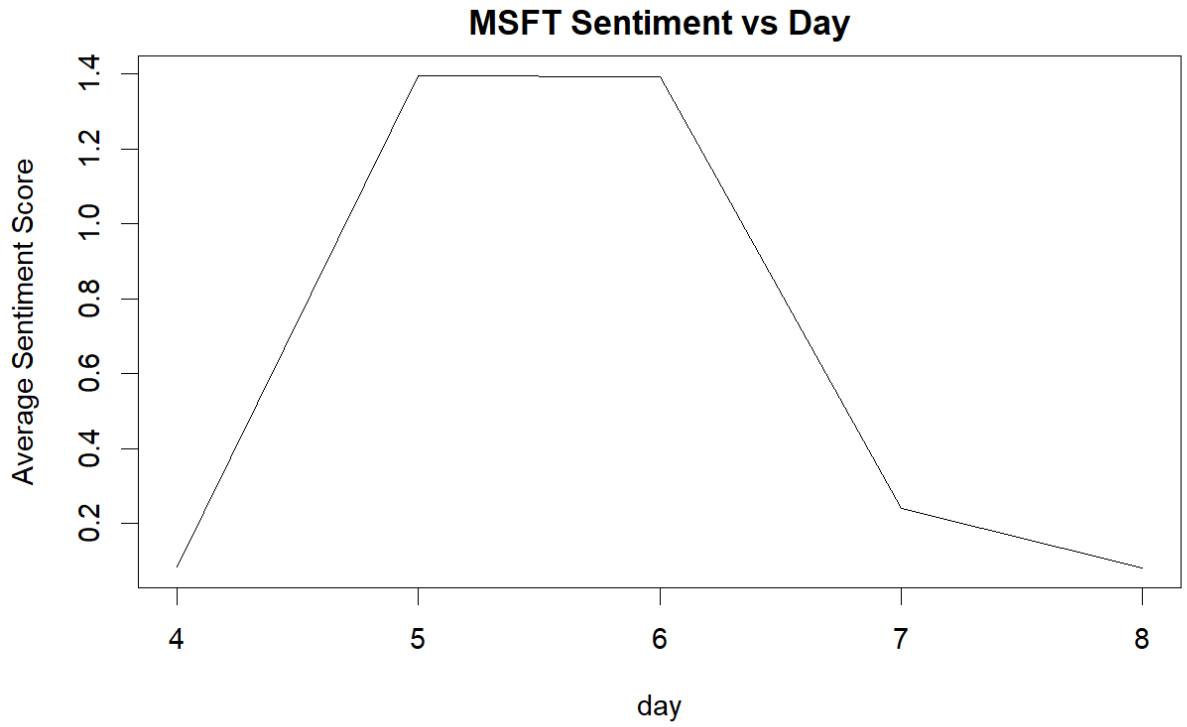

Figure 5: MSFT Sentiment

```
TSLA_Sentiments <- TSLA_Tweets_raw[,c(1,4)]
TSLA_Sentiments[,3]= 0
names(TSLA_Sentiments)[3]<-"Sentiment"
TSLA_Sentiments$Day <- day(TSLA_Sentiments$created_at)

for (i in (1:nrow(TSLA_Sentiments))){
TSLA_Sentiments[i,3] <- FindSentiment(TSLA_Tweets_raw[i,])
}

TSLA_Sentiments_Plot <- as.data.frame(unique(TSLA_Sentiments$Day))
TSLA_Sentiments$DayAverage <- 1.0
TSLA_Sentiments_Plot<-as.data.frame(TSLA_Sentiments_Plot)
```

```
TSLA_Sentiments <- as.data.frame(TSLA_Sentiments)
for(i in (1:nrow(TSLA_Sentiments_Plot))){
  TSLA_Sentiments_Plot[i,2] <- mean(TSLA_Sentiments[TSLA_Sentiments$Day == TSLA_Sentiments_Plot[i,1], '
}

TSLA_Sentiments_Plot
TSLA_Sentiments_Plot<- arrange(TSLA_Sentiments_Plot,`unique(TSLA_Sentiments$Day)`)
TSLA_Sentiments_Plot<-head(TSLA_Sentiments_Plot,5)
plot(TSLA_Sentiments_Plot$V2~TSLA_Sentiments_Plot$`unique(TSLA_Sentiments$Day)`,type='l', xlab = "day",
```

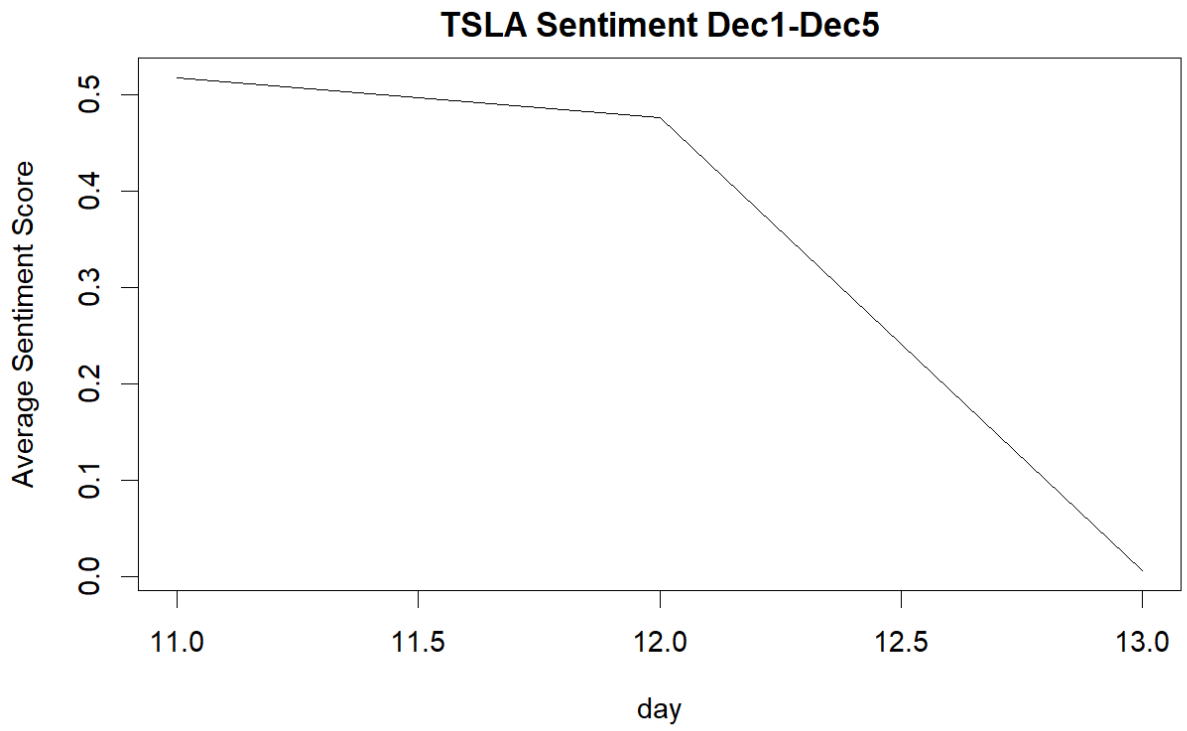

Figure 6: TSLA Sentiment

```
Target_Sentiments <- Target_Tweets_raw[,c(1,4)]
Target_Sentiments[,3]= 0
names(Target_Sentiments)[3]<-"Sentiment"
Target_Sentiments$Day <- day(Target_Sentiments$created_at)

for (i in (1:nrow(Target_Sentiments))){
Target_Sentiments[i,3] <- FindSentiment(Target_Tweets_raw[i,])
}

Target_Sentiments_Plot <- as.data.frame(unique(Target_Sentiments$Day))
Target_Sentiments$DayAverage <- 1.0


for(i in (1:nrow(Target_Sentiments_Plot))){
  Target_Sentiments_Plot[i,2] <- mean(Target_Sentiments$Day==Target_Sentiments_Plot[i,1])
```

```
}
print(unique(Target_Sentiments$Day))
Target_Sentiments_Plot
Target_Sentiments_Plot<- arrange(Target_Sentiments_Plot,`unique(Target_Sentiments$Day)`)
Target_Sentiments_Plot<-head(Target_Sentiments_Plot,5)
plot(Target_Sentiments_Plot$V2~Target_Sentiments_Plot$`unique(Target_Sentiments$Day)`,type='l', xlab =
```

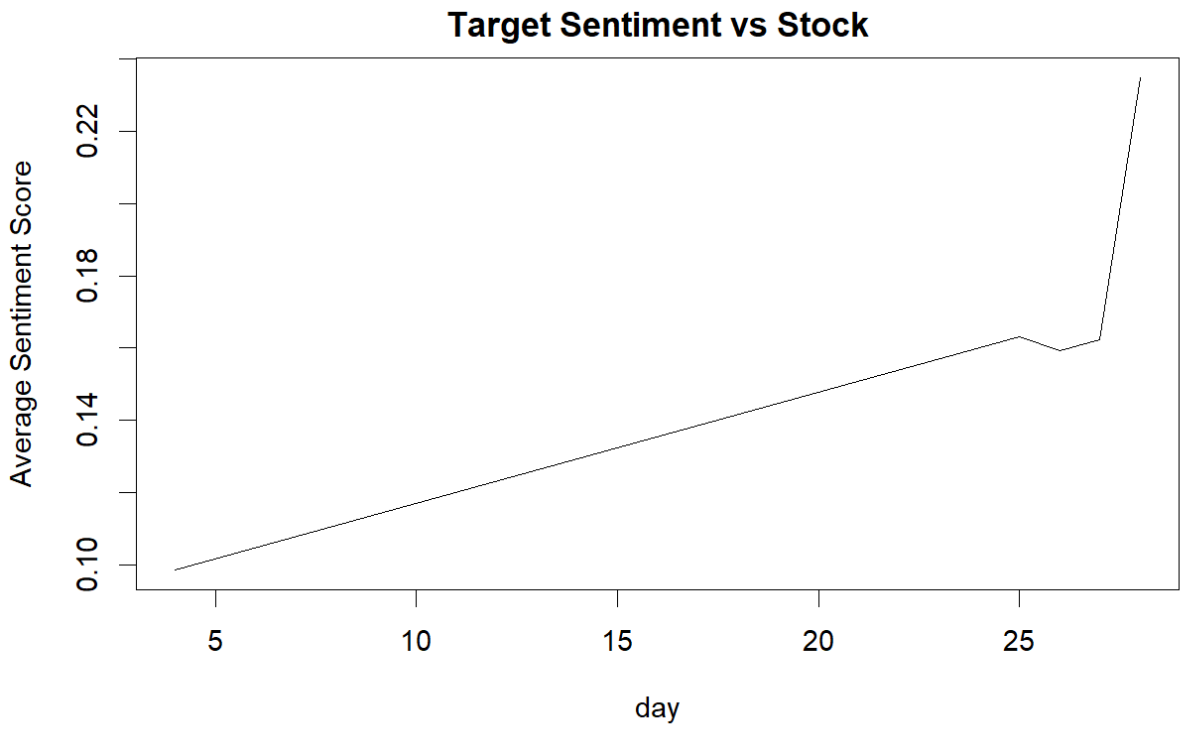

Figure 7: Target Sentiment

```
Apple_Sentiments <- Apple_Tweets_raw[,c(1,4)]
Apple_Sentiments[,3]= 0
names(Apple_Sentiments)[3]<-"Sentiment"
Apple_Sentiments$Day <- day(Apple_Sentiments$created_at)

for (i in (1:nrow(Apple_Sentiments))){
Apple_Sentiments[i,3] <- FindSentiment(Apple_Tweets_raw[i,])
}

Apple_Sentiments_Plot <- as.data.frame(unique(Apple_Sentiments$Day))
Apple_Sentiments$DayAverage <- 1.0


for(i in (1:nrow(Apple_Sentiments_Plot))){
  Apple_Sentiments_Plot[i,2] <- mean(Apple_Sentiments$Day==Apple_Sentiments_Plot[i,1])
}

Apple_Sentiments_Plot
```

```
Apple_Sentiments_Plot<- arrange(Apple_Sentiments_Plot,`unique(Apple_Sentiments$Day)`)
Apple_Sentiments_Plot<-head(Apple_Sentiments_Plot,5)
plot(Apple_Sentiments_Plot$V2~Apple_Sentiments_Plot$`unique(Apple_Sentiments$Day)`,type='l', xlab = "day
```

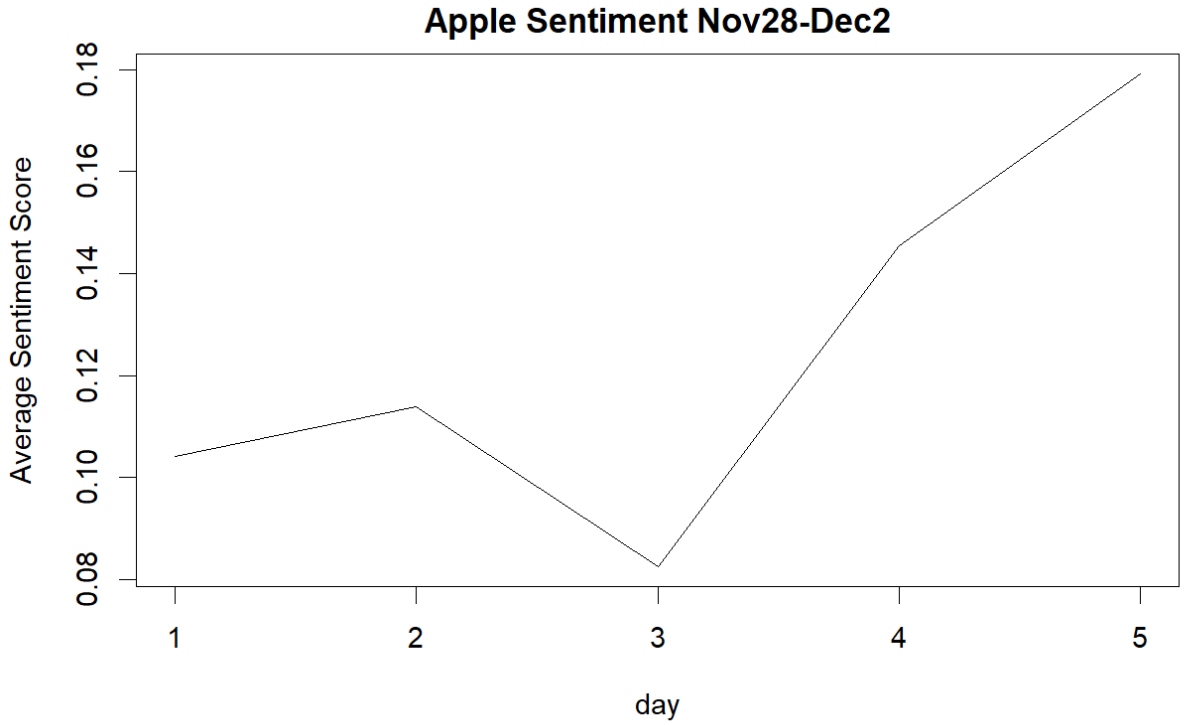

Figure 8: Apple Sentiment

In Conclusion, there was no correlation of stock price and sentiment score. Considering we were only limited to a five day interval based on api restrictions, more accurate results would maybe arise if the interval was longer and sentiment bins were placed in a weekly or monthly basis. Further data cleaning and fine tuning of the sentiment library could also be used when determining sentiment score.