Business analysis

R Markdown

Business Problem : The business question are from the perspective of a investment firm looking to invest in the stock of Apple inc.

libraries required

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.1.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.1.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(fBasics)
```

```
## Warning: package 'fBasics' was built under R version 4.1.2
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
## Warning: package 'timeSeries' was built under R version 4.1.2
```

```
##
## Attaching package: 'timeSeries'
```

```
## The following object is masked from 'package:zoo':
##
##     time<-
```

```
library(rugarch)
```

```
## Warning: package 'rugarch' was built under R version 4.1.2
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'rugarch'
```

```
## The following object is masked from 'package:stats':
##
##     sigma
```

```
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:timeSeries':
##
##     filter
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

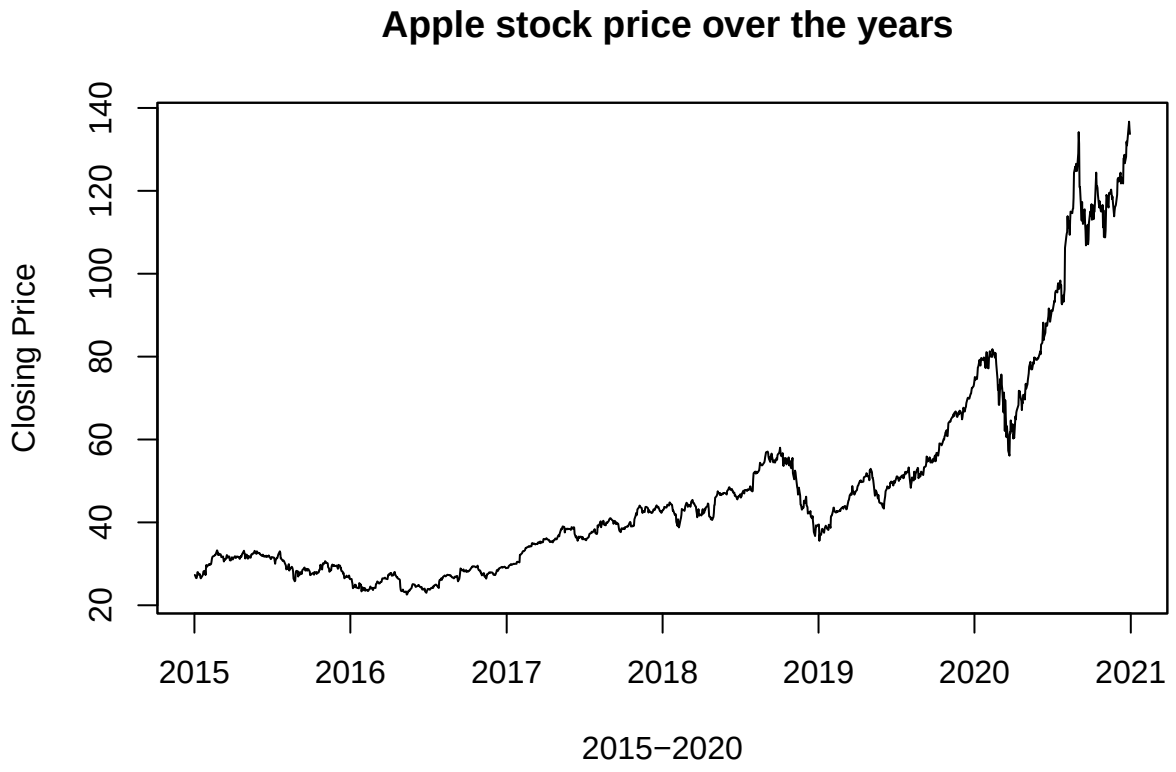
Probablity questions

Question 1 : Lets understand the previous price trends of the Apple inc ticker?

```
#reading the dataset
apple_price <- read.csv("AAPL.csv")

apple_price$Date <- as.Date(apple_price$Date,"%m/%d/%Y")

#subsetting data according to relevant dates
apple_price_relevant_before <- apple_price[apple_price$Date > "2014-12-31" & apple_price$Date < "2020-1


#creating a timely ordered data set to perform time series regression
apple_price_relevant = zoo(apple_price_relevant_before$Close, apple_price_relevant_before$Date)
head(apple_price_relevant)
```

```
## 2015-01-02 2015-01-05 2015-01-06 2015-01-07 2015-01-08 2015-01-09
##    27.3325    26.5625    26.5650    26.9375    27.9725    28.0025
```

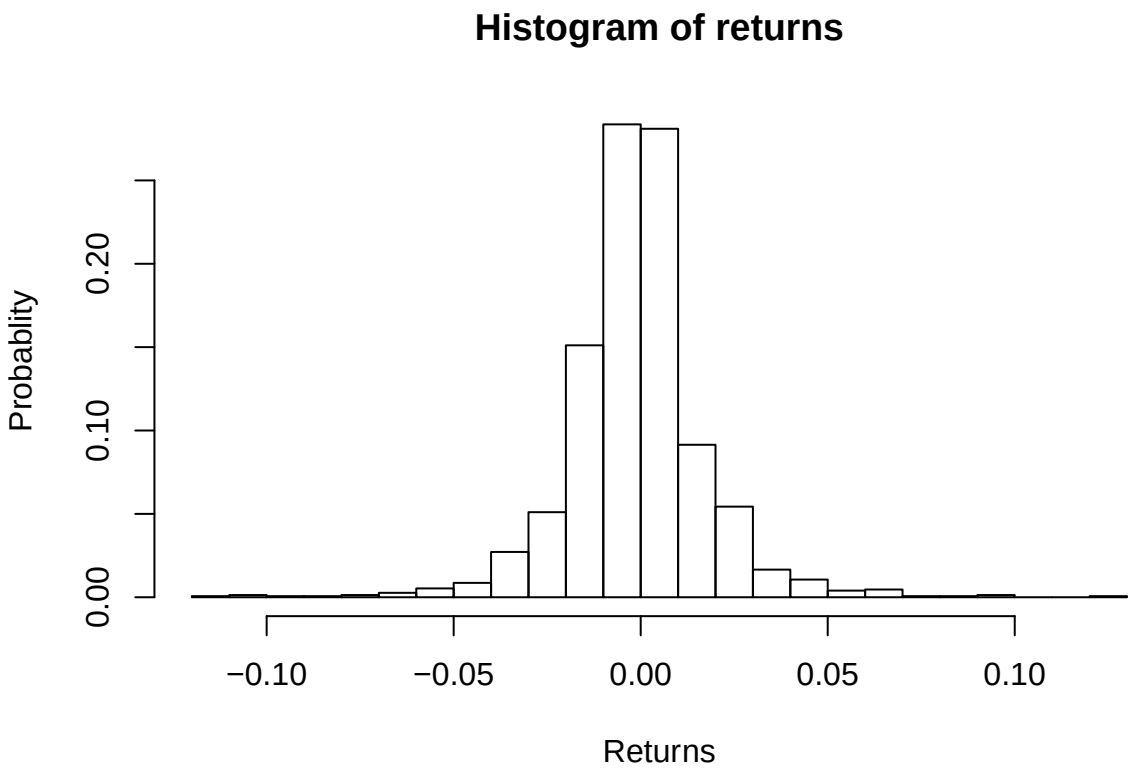
```
#The apple stock price between the
par(mfrow=c(1,1))
plot(apple_price_relevant, type='l', ylab = "Closing Price", xlab = '2015-2020' , main=" Apple stock pr
```



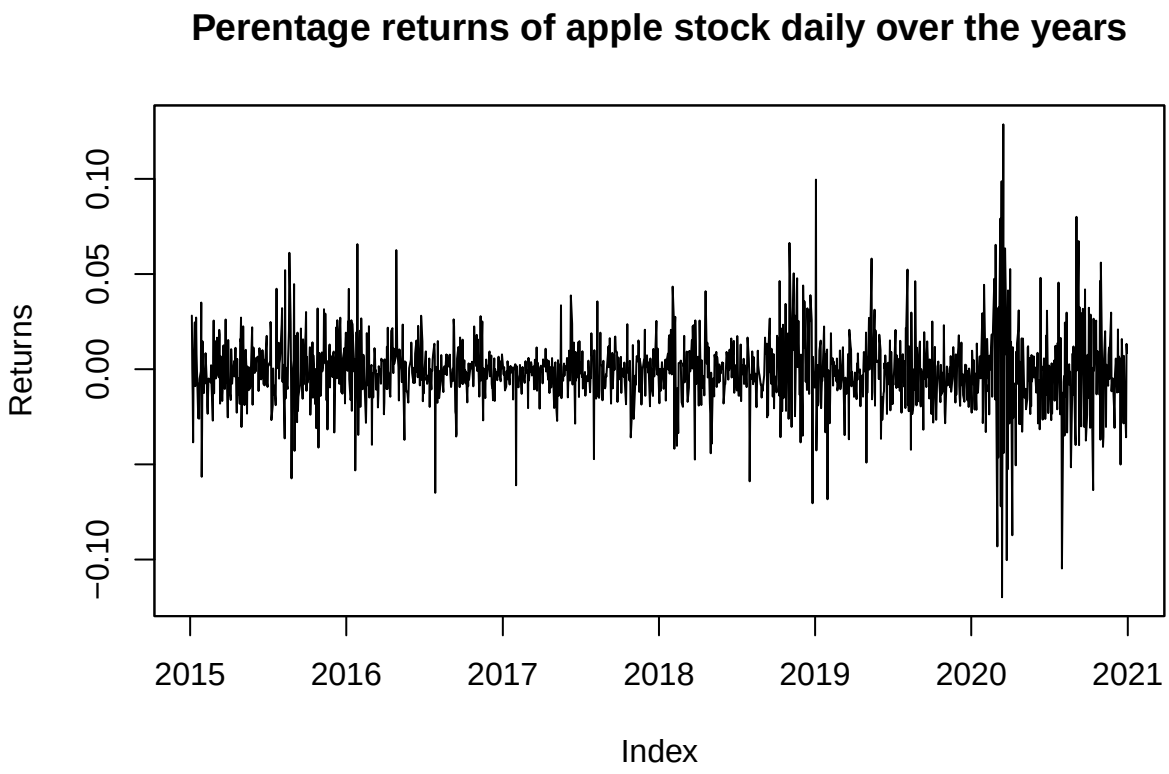
#Conclusion : As you can see from the above representation the apple stock price has historically compounded over the years. Historically apple stock have given an compounded annual return of around 33%. The stock had fallen recently due to the corona crisis to $55 in March of 2019 and recovered to the new high levels of 130's by December of 2020. This is the summary of the Apple stock price over the years.

Question 2 : How have the daily returns of Apple's stock been between 2015-2020?

```
#to calculate abs returns for each day
returns = 1- (apple_price_relevant/lag(apple_price_relevant, -1))

h <- hist(returns, breaks = 20, plot=FALSE)
h$counts=h$counts/sum(h$counts)
plot(h,xlab='Returns',ylab="Probablity")
```




```
#displaying the absolute returns over the years
par(mfrow=c(1,1))
a <- plot(returns, type='l', ylab = "Returns", main="Perentage returns of apple stock daily over the yea
```



Conclusion :

The graph above gives us the probability distribution of daily returns given by Apple Inc stock between the years 2015 and 2020. Above distribution shows that that Apple stock is not a highly volatile stock with almost 50 percent of all the daily returns lying in between -1% / +1% range.

Question 3 : Can a long/short position in apple stock generate more than a 5% return in day(considering absolute returns)?

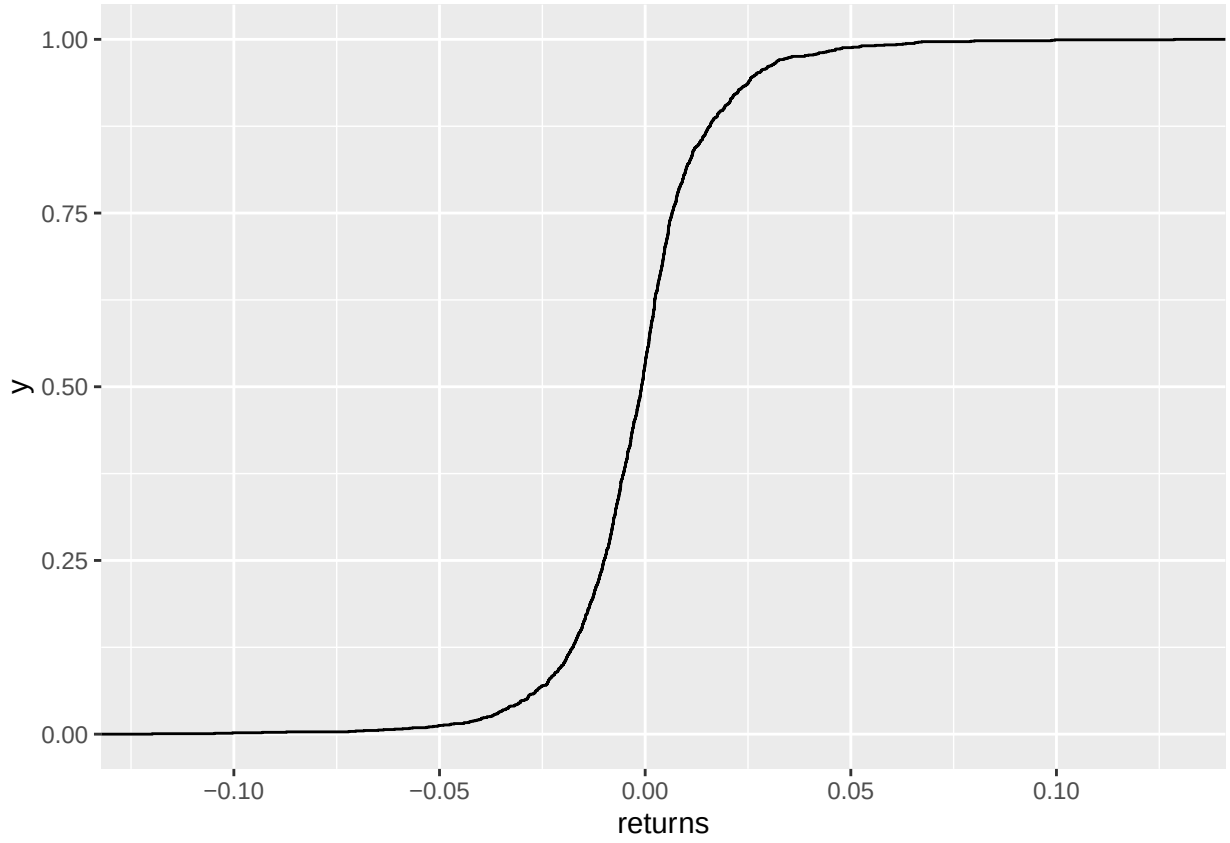
```
print(quantile(returns,1:80/80))
```

```
##         1.25%          2.5%         3.75%           5%         6.25%
## -0.0492362924 -0.0377879885 -0.0332398208 -0.0288431718 -0.0264176087
##          7.5%         8.75%          10%        11.25%         12.5%
## -0.0235913496 -0.0219014575 -0.0198662718 -0.0188620509 -0.0175513902
##        13.75%          15%        16.25%         17.5%        18.75%
## -0.0166471859 -0.0155056042 -0.0148740021 -0.0140258121 -0.0133491657
##           20%        21.25%         22.5%        23.75%          25%
## -0.0125170434 -0.0119165864 -0.0111387788 -0.0104965623 -0.0100170492
##        26.25%         27.5%        28.75%          30%        31.25%
## -0.0093747668 -0.0087408877 -0.0083075705 -0.0079255993 -0.0074619544
##         32.5%        33.75%          35%        36.25%         37.5%
```

```
## -0.0070806177 -0.0065900265 -0.0061312162 -0.0058919212 -0.0052639969
##        38.75%           40%        41.25%         42.5%        43.75%
## -0.0047963007 -0.0042986313 -0.0037996921 -0.0034621941 -0.0030989284
##           45%        46.25%         47.5%        48.75%           50%
## -0.0026749905 -0.0020463813 -0.0016500715 -0.0011967839 -0.0008932886
##        51.25%         52.5%        53.75%           55%        56.25%
## -0.0004926635 -0.0002219613  0.0000941117  0.0004398384  0.0008832756
##         57.5%        58.75%           60%        61.25%         62.5%
##  0.0011195653  0.0014489716  0.0018734563  0.0022001140  0.0023415033
##        63.75%           65%        66.25%         67.5%        68.75%
##  0.0028653531  0.0031934879  0.0036842991  0.0041338879  0.0044577835
##           70%        71.25%         72.5%        73.75%           75%
##  0.0047719982  0.0052949811  0.0056349215  0.0059241400  0.0065276879
##        76.25%         77.5%        78.75%           80%        81.25%
##  0.0072512627  0.0077623130  0.0084278646  0.0092658455  0.0098685630
##         82.5%        83.75%           85%        86.25%         87.5%
##  0.0108151029  0.0115743858  0.0131148413  0.0144946597  0.0157330804
##        88.75%           90%        91.25%         92.5%        93.75%
##  0.0171054329  0.0190040821  0.0207088545  0.0225979790  0.0250405942
##           95%        96.25%         97.5%        98.75%          100%
##  0.0270243351  0.0306380897  0.0356442816  0.0478127973  0.1286469475
```

```
po <- ggplot(data = returns, aes(x=returns))
po +stat_ecdf()
```

```
## Don't know how to automatically pick scale for object of type zoo. Defaulting to continuous.
```



Conclusion :

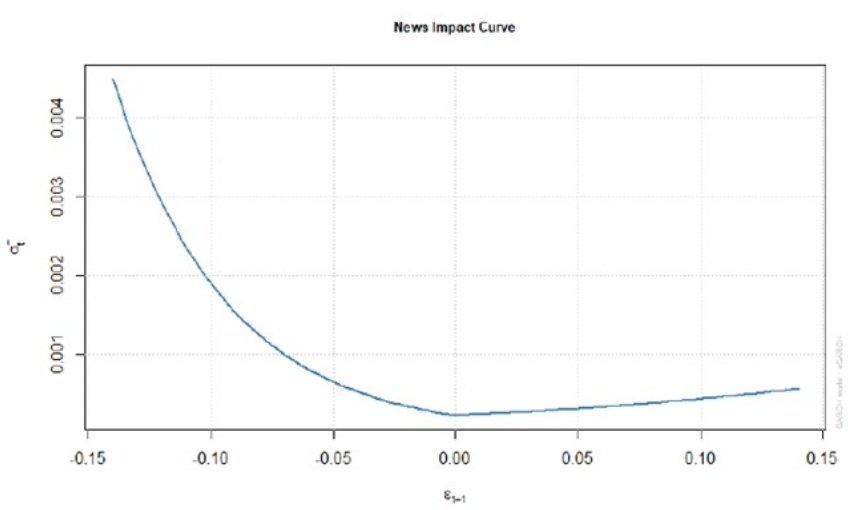
As you can see for the above return cdf and percentile data, historically about 1.25% of the times the stock has fallen more than 5% between adjacent days closing and about 1% of the time the stock price has increased more than 5% between adjacent days closing. This shows us that the probability of apple stock generating returns greater than 5 percent given a long/short position (if returns are in absolute investment amounts without leverage) is less than 2.5%. This means the apple stock barely reacts drastically to the market conditions/news and hence you have a very less probability of making more than five percent in a given day, by trading Apple stock.

Time series Analysis

Question 4 :

Can we make a strategy based on the volatility of the stock based on previous stock price data?

```
# creates time plot of log returns
returns_log = log(apple_price_relevant/lag(apple_price_relevant, -1))

egarch11.t.spec=ugarchspec(variance.model=list(model = "eGARCH", garchOrder=c(1,1)), mean.model=list(ar
#estimate model
egarch11.t.fit=ugarchfit(spec=egarch11.t.spec, data=returns_log)
#plot(egarch11.t.fit)
#12
```



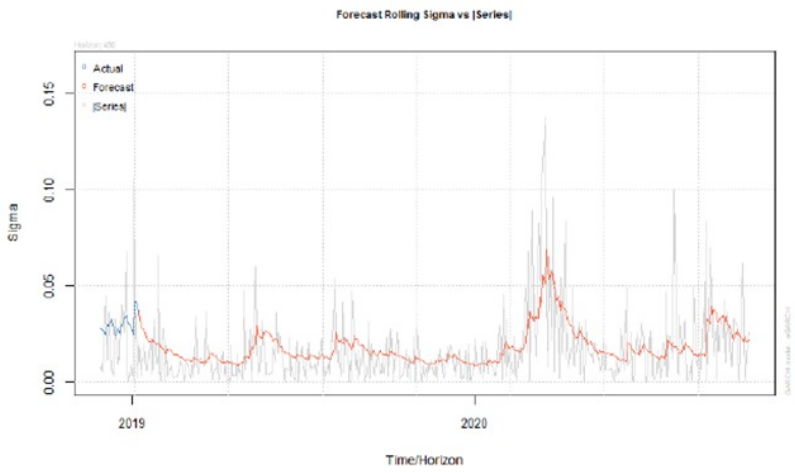
Conclusion :

While trying to find out the variation of price change percentage while performing time series prediction using GARCH model. We found out that the volatility of percentage price change is much higher when the price is falling as compared to the volatility of percentage price change when the price is increasing. Using this nature of the ticker we can use strategies which make profit based on the volatility of the stock price. The trader goes both long and short on a stock at the same time creating a band on both the positive and negative side of the current price within which the trader is at loss. Whenever the ticker exceeds the price the trader will earn a profit. This News Impact curve helps us understand that this can likely be done in case of the apple stock when the price starts falling as the price volatility of the stock drastically increases.

Question 5 :

Continuing on the above question can we find a model to predict the volatility of the stock price to build our investment models?

```
rff=ugarchfit(spec=egarch11.t.spec, data=returns_log, out.sample=500)
rf=ugarchforecast(rff, n.ahead=20, n.roll=450)
#plot(rf)
```



Conclusion :

We were able to adequately predict the volatility or the sigma of the stock price with adequate effect. This model can be fine tuned with better data sets of stock prices with each transaction being reflected in the data set instead of just daily data. These kind of datasets are hard to access and need to be bought from brokers on whose exchange the transactions are taking place and hence, they are very expensive. The regression model looks adequate in predicting volatility given our data sets.

Text analysis and clustering ##question

Question 6: Is the market price of Apple stock related to the sentiment of that stock amongst inverstors? (we will use twitte'rs tweet data to find out the sentiment of the stock amongst investors)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:timeSeries':
##
##     filter, lag
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## Warning: package 'forcats' was built under R version 4.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks plotly::filter(), timeSeries::filter(), stats::filter()
## x dplyr::lag()    masks timeSeries::lag(), stats::lag()
## x purrr::reduce() masks rugarch::reduce()
```

```
library(stringr)
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.1.2
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.1.2
```

```
library(tokenizers)
```

```
## Warning: package 'tokenizers' was built under R version 4.1.2
```

```
library(stopwords)
```

```
## Warning: package 'stopwords' was built under R version 4.1.2
```

```
##
## Attaching package: 'stopwords'
```

```
## The following object is masked from 'package:tm':
##
##     stopwords
```

```
require(plyr)
```

```
## Loading required package: plyr
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:plotly':
##
##     arrange, mutate, rename, summarise
```

```
library(syuzhet)
```

```
## Warning: package 'syuzhet' was built under R version 4.1.2
```

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.1.2
```

```
## Loading required package: RColorBrewer
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.1.2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(cluster)
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 4.1.2
```

```
aapl_tweets <- read_csv("aapl_tweets_cleaned.csv")
```

```
## New names:
## * `` -> ...1
```

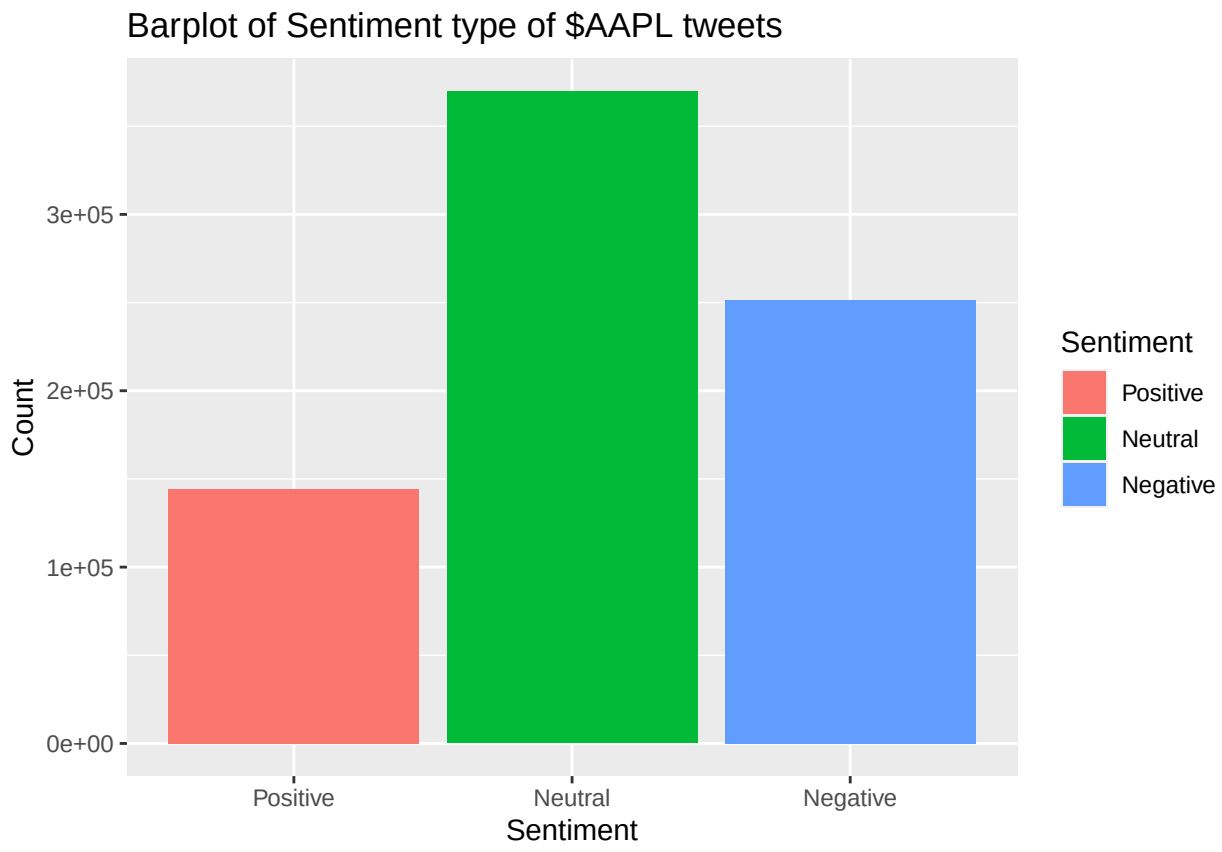
```
## Rows: 1181911 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): unique.aapl_tweets.body.
## dbl (1): ...1
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Obtain sentiment of the Tweet based on the positive and the negative words that is being read
aapl_tweets$unique.aapl_tweets.body. <- str_replace_all(aapl_tweets$unique.aapl_tweets.body.,"[^[:graph
tweet_sentiment <- get_sentiment(aapl_tweets$unique.aapl_tweets.body.)
# Drop N/A and NULL values
aapl_tweets <- aapl_tweets %>%
  drop_na()
# Histogram of sentiment of tweets
neutral <- length(which(tweet_sentiment == 0))
positive <- length(which(tweet_sentiment > 1))
negative <- length(which(tweet_sentiment < 0))
Sentiment <- c("Positive","Neutral","Negative")
Count <- c(positive,neutral,negative)
output <- data.frame(Sentiment,Count)
```

```
output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
output_plot <- ggplot(output, aes(x=Sentiment,y=Count))+
  geom_bar(stat = "identity", aes(fill = Sentiment))+
  ggtitle("Barplot of Sentiment type of $AAPL tweets")

output_plot
```



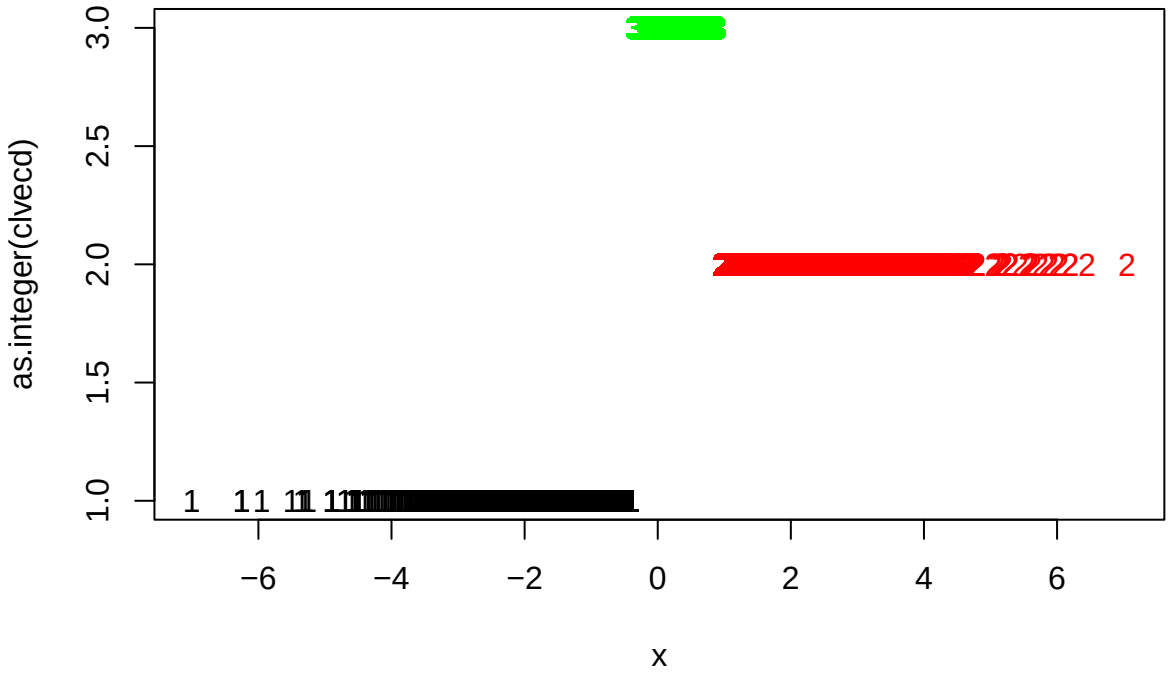
Conclusion :

As shown by our results for all the years above. We can see that the sentiment for the apple stock does not reflect the apple stock price. The stock price of apple has increased more than three hundred percent over the years whereas the general sentiment of apple stock over the years has been largely negative or neutral based on our tweets data. Let us further study the relationship of market sentiment and share price to further establish the relationship between the both of them using an market events in the next question.

Question 7: Can we make a clustering algorithm to determine the sentiment of the market based on tweets data to make an investment strategy based on the same?

```
# Number of positive and negative tweets about $AAPL for the month of October, 2018
tweets <- read.csv("tweets/Tweet.csv")
```

```
october_data <- data.frame(post_date = tweets$post_date, tweet = tweets$body)
october_data$post_date <- as_datetime(october_data$post_date)
october_data$post_date <- format(october_data$post_date, "%Y-%m-%d")
october_data$post_date <- as.Date(october_data$post_date)
subset_dates <- interval(start = "2018-10-01", end = "2018-10-31")
october_data <- october_data[which(october_data$post_date %within% subset_dates), ]
october_data$sentiment <- get_sentiment(october_data$tweet)
october_kmeans_sentiment <- kmeans(data.frame(october_data$sentiment), centers = 3, nstart = 20)
cluster_plot <- plotcluster(october_data$sentiment, october_kmeans_sentiment$cluster)
```



Conclusion : It can be noted that the clustering of tweets yield the following results: For the month of October 2018, our clustering model shows that the sentiment of the market was largely negative to neutral with over 66% of tweets in this range. The sentiment analysis model can be further used to make an investment strategy as our model's results for the market sentiment correlates with $AAPL's stock price during this time period. It can also be noted that the variance of the positive tweets is more when compared to the other two.

Question 8: What does the word cloud for the tweets based on $AAPL's stock look like?

```
# Word Cloud
#text_corpus <- Corpus(VectorSource(october_data$tweet))
#tdm <- TermDocumentMatrix(text_corpus)
#tdm <- as.matrix(tdm)
```

```
#tdm <- sort(rowSums(tdm), decreasing = TRUE)
#tdm <- data.frame(word = names(tdm), freq = tdm)
#set.seed(123)
#word_cloud <- wordcloud(text_corpus, min.freq = 1, max.words = 100, scale = c(2.2,1),
#          colors=brewer.pal(8, "Dark2"), random.color = T, random.order = F)

#word_cloud
```

#Conclusion : **Note**: The code above has been commented since we were facing memory constraint issues as this particular case requires a vector of 90.5GB to be allocated, which was not possible while knitting the file. However, you can run the above piece of code as a standalone *.R file and you will get the results as shown below. As we can see in the image, Apple's products such as iPhone, iPad are talked about at large. People also talk about the effect China has on Apple since they is known to hire a lot of cheap labor from China to assemble their products. Also, we can conclude from the wordcloud that the general consensus is to compare Apple's stock to other tech giants like Amazon($AMZN$), $Google$(GOOG) and Microsoft($MSFT).