assessment

Yuta Hayashi

5/1/2019

Read the packages we use.

```
library(readr)
library(ggplot2)
library(dplyr)
library(lubridate)
library(plyr)
library(devtools)
library(tidyr)
library(quantmod)
```

Read the data sets we use.

```
#read SPY.csv
SPY <- read_csv("SMA Intern Test Set/SPY.csv")
#read TSLA.csv
TSLA <- read_csv("SMA Intern Test Set/TSLA.csv")
#read TSLA_Score
TSLA_Score <- read_csv("SMA Intern Test Set/TSLA_Score.csv")
```

Section 1: Calculate Control Series

1. Use the series 'SPY.csv' to calculate the cumulative daily open-to-close return for the timeperiod in the file. (Buy at open price, sell at close price, and accumulate these returns over time) Plot the results.

First, since date is stored in character type, it is necessary to convert it to date type. After that, we change the order of the time(oldest to latest).

```
#change date type from character to date
SPY$Date <- as.Date(SPY$Date,tryFormats = "%m/%d/%Y")
#Change the date (oldest to latest)
SPY <- SPY[order(as.Date(SPY$Date, "%m/%d/%Y"), decreasing = FALSE),]
```

Below is the formula of open-to-close return value and cumulative return value.

$$Return_n = \frac{AdjClose_n - AdjOpen_n}{AdjOpen_n} = r_n$$

$$CumulativeReturn = \prod_{i=1}^{n}(1 + r_n) - 1$$

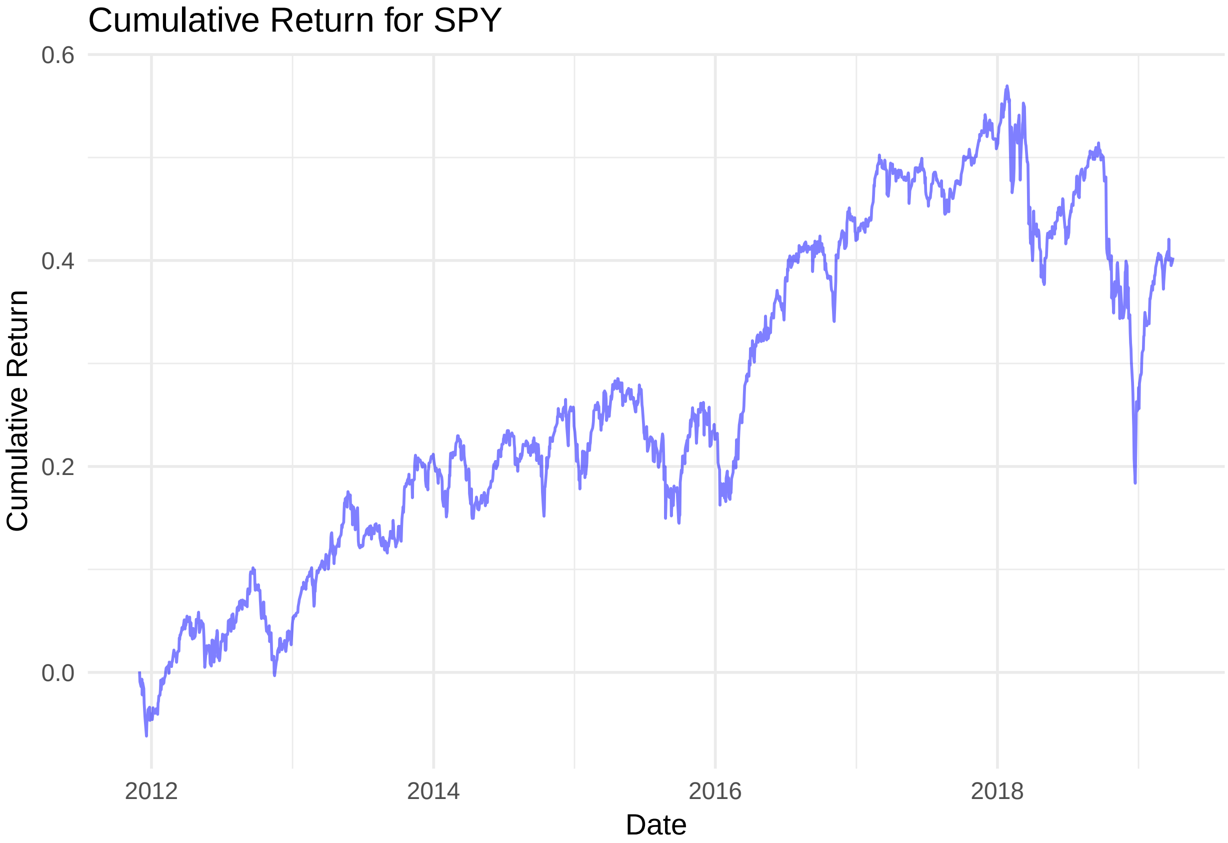
Cumulative return variable creation:

```
#daily open-to-close return
SPY$return <- (SPY$Adj_Close - SPY$Adj_Open) / SPY$Adj_Open
#open-to-close + 1 = rate
SPY$rate <- SPY$return + 1
```

```
#cumulative return using cumsum function put cum_return variable in SPY.csv
SPY$cum_return <- cumprod(SPY$rate)-1
```

Plot the results.

```
#as.numeric(SPY$cum_return)
#mean(SPY$cum_return)

#data visualization stored in p
p <- ggplot(data = SPY, aes(x = SPY$Date, y = SPY$cum_return))+
      geom_line(color = "blue", alpha=0.5) +
      ggtitle("Cumulative Return for SPY") +
      xlab("Date") +
      ylab("Cumulative Return") +
      theme_minimal()

print(p)
```



```
#save as the pdf
ggsave(file="1-1.pdf", width=15, height=7,dpi=400)
```

2. Repeat question 1 with 'TSLA.csv'. Plot the results.

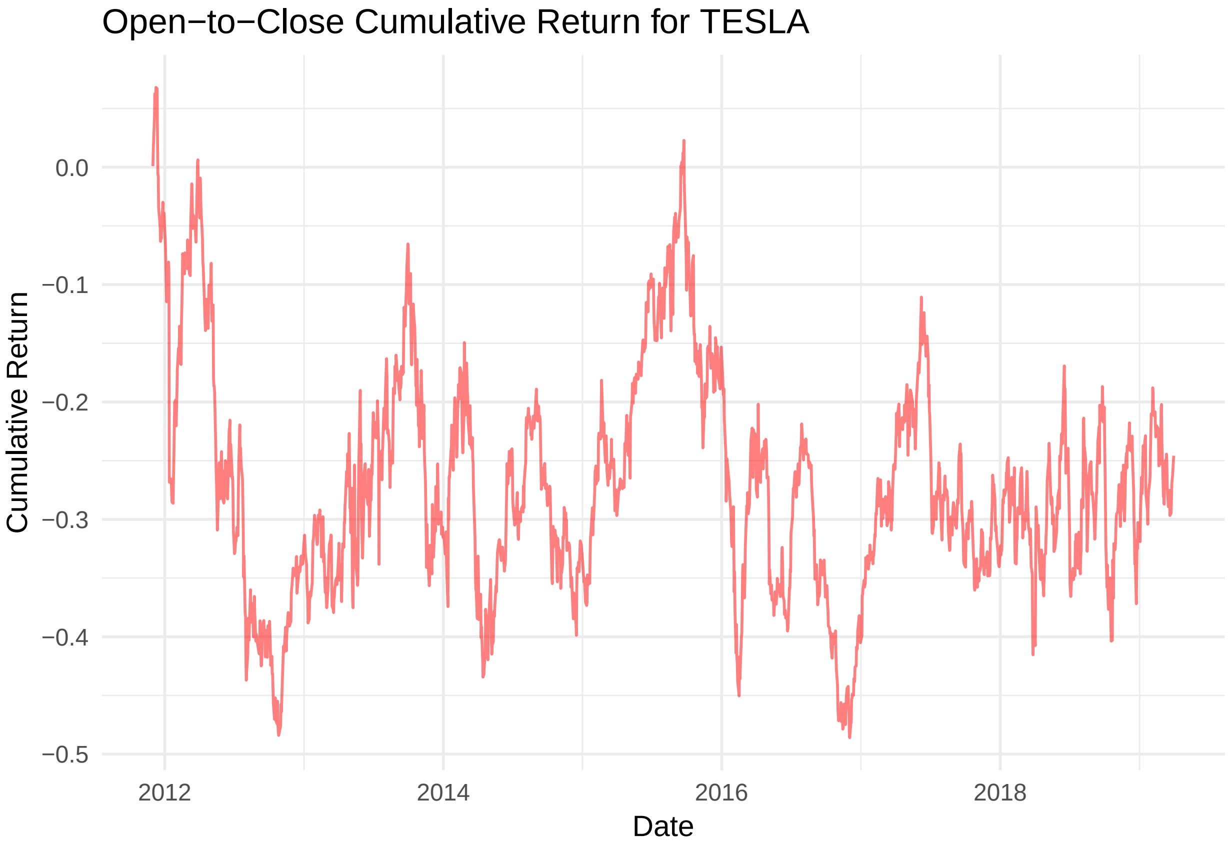
```
#change date type from character to date
TSLA$Date <- as.Date(TSLA$Date,tryFormats = "%m/%d/%Y")

#Change the date (oldest to latest)
```

```
TSLA <- TSLA[order(as.Date(TSLA$Date, "%m/%d/%Y"), decreasing = FALSE),]
```

Calculate open-to-close cumulative value just like SPY data set.

```
#daily open-to-close return
TSLA$return <- (TSLA$Adj_Close - TSLA$Adj_Open)/TSLA$Adj_Open
#open-to-close + 1 = rate
TSLA$rate <- TSLA$return + 1
#cumulative return using cumsum function put cum_return variable in SPY.csv
TSLA$cum_return <- cumprod(TSLA$rate)-1
```

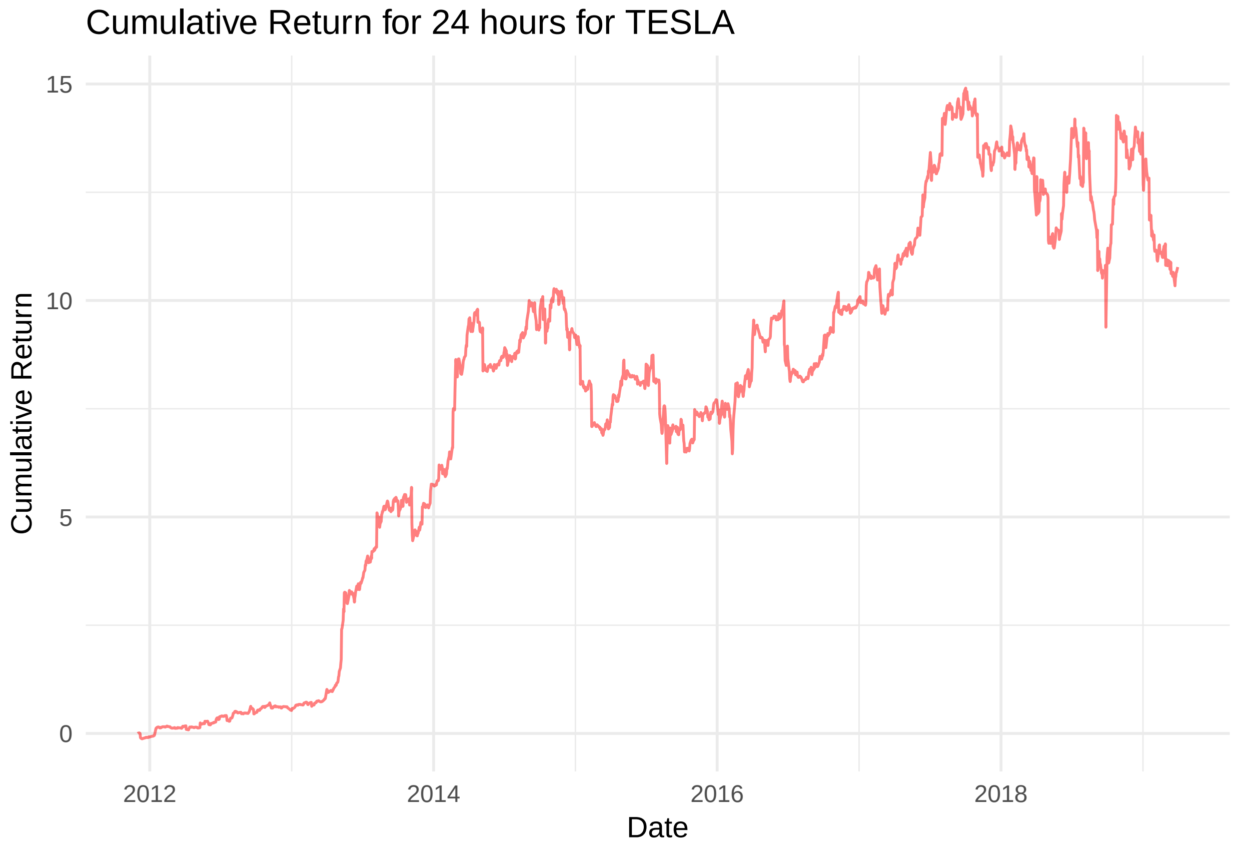
```
#data visualization stored in q
q1 <- ggplot(data = TSLA, aes(x = TSLA$Date, y = TSLA$cum_return)) +
       geom_line(color = "red", alpha=0.5) +
       ggtitle("Open-to-Close Cumulative Return for TESLA") +
       xlab("Date") +
       ylab("Cumulative Return") +
       theme_minimal()

print(q1)
```



```
#save as the pdf
ggsave(file="1-2-1.pdf", width=15, height= 7,dpi=400)
```

This Open-to-Close plot does not make any sense since according to YAHOO Finance, TSLA stock exponentially goes up from about 5/2013. So in order to attain similar graph as YAHOO Finance, we attain between a day return(Close-to-Close return).

```r
TSLA$prev_close <- c(NA, TSLA$Adj_Close[1:nrow(TSLA)-1])
TSLA$Return <- (TSLA$Adj_Open - TSLA$prev_close)/TSLA$prev_close
TSLA$Rate <- TSLA$Return + 1
TSLA$Cum_return <- cumprod(TSLA$Rate)-1
TSLA[1,8:10] <- 1
TSLA$Cum_return <- cumprod(TSLA$Rate)-1

#data visualization stored in q
q2 <- ggplot(data = TSLA, aes(x = TSLA$Date, y = TSLA$Cum_return)) +
       geom_line(color = "red", alpha=0.5) +
       ggtitle("Cumulative Return for 24 hours for TESLA") +
       xlab("Date") +
       ylab("Cumulative Return") +
       theme_minimal()

print(q2)
```

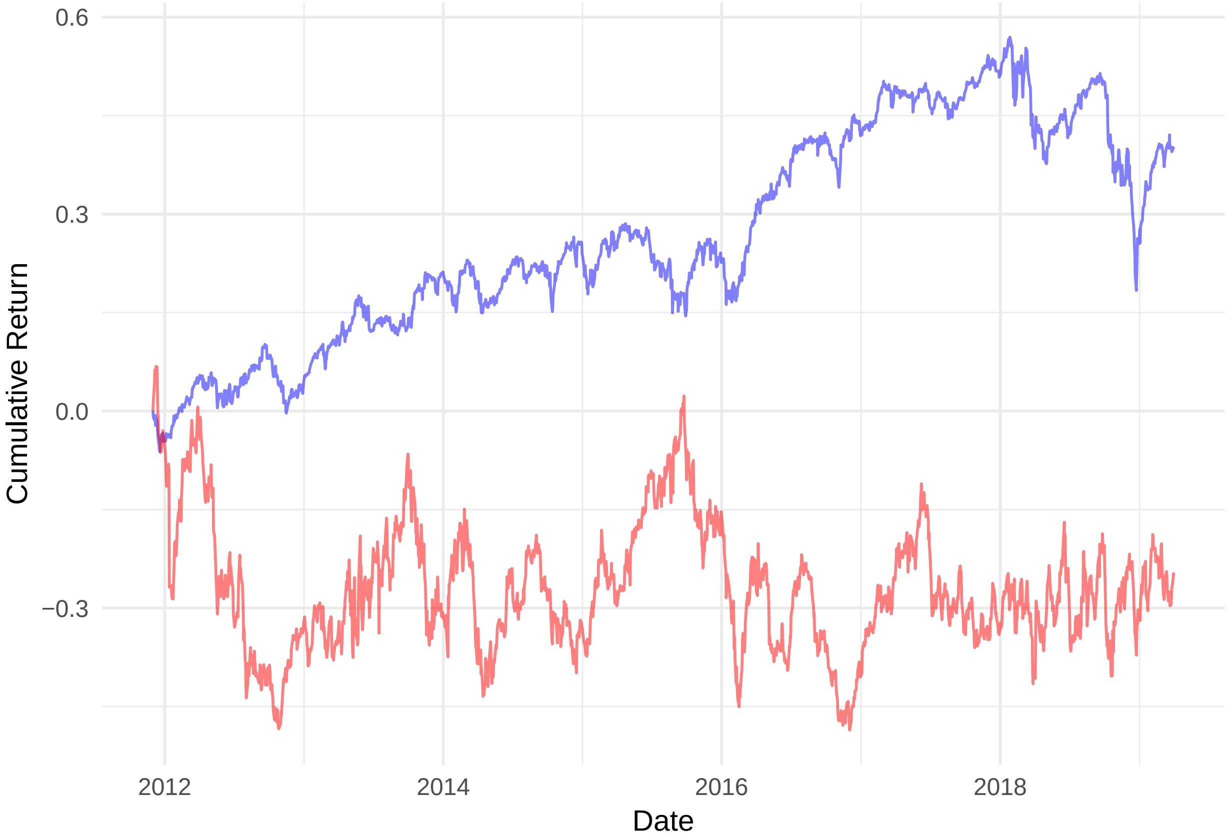


```r
#save as the pdf
ggsave(file="1-2-2.pdf", width=15, height= 7,dpi=400)
```

This is a lot similar to what we have in YAHOO Finance.

3. Plot results from questions 1 and 2 on the same graph to compare the returns of SPY and TSLA.

```
#plot two graph in one plot
r <- ggplot() +
      geom_line(data = SPY, aes(x = Date, y = cum_return), color = "blue", alpha=0.5) +
      geom_line(data = TSLA, aes(x = Date, y = cum_return), color = "red", alpha=0.5) +
      xlab("Date") +
      ylab("Cumulative Return") +
      theme_minimal()

print(r)
```



```
#save as the pdf
ggsave(file="1-3.pdf", width=15, height=7, dpi=400)
```

Summary

It makes sense to think that SPY has less volatility since 1) Adj_Open and Adj_Close prices are much bigger for SPY than those of TSLA so it is less likely to fluctuate compared to TSLA 2)since ETF is a mutual fund with 500 stock names in the U.S., even if one of the stock goes down, they offset the return all the time.

Tesla is a young company which was founded about 15 years ago, therefore, the market is fast-growing but not stable. In addition to that, since it is an automotive and energy company their stock price is heavily influenced by battery production in China, for example. Also they are influenced by announcement as well particularly by well-known CEO, Elon Musk.

Nevertheless, it started to stabilize at around 2018 for cumulative return for TSLA.

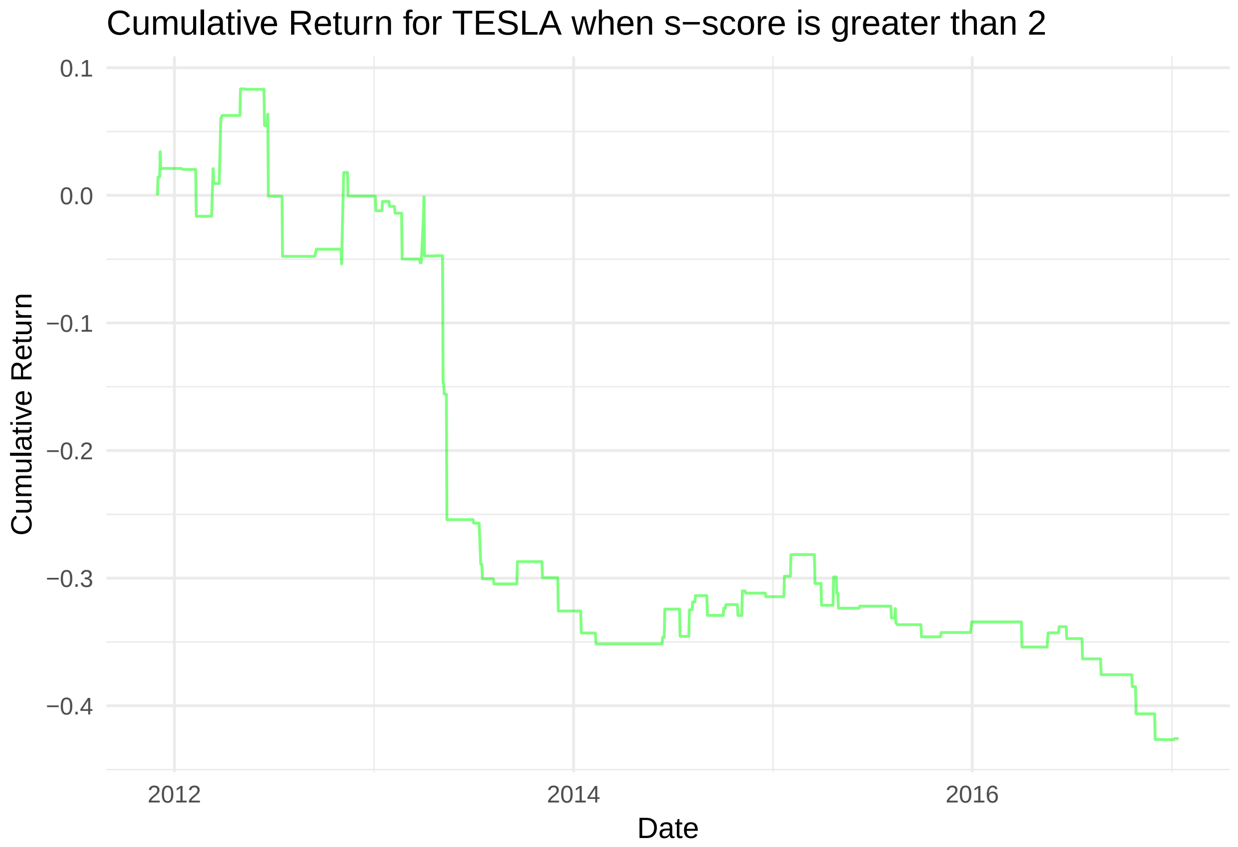
Section 2: Calculate Test Set

```
#change date type from character to date
TSLA_Score$date <- as.Date(TSLA_Score$date, tryFormats = "%m/%d/%Y %H:%M")

#Change the date (oldest to latest)
TSLA_Score <- TSLA_Score[order(as.Date(TSLA$Date, "%m/%d/%Y %H:%M"), decreasing = FALSE),]

#change the key column name from "date" to "Date"
colnames(TSLA_Score)[colnames(TSLA_Score)=="date"] <- "Date"

#inner join by date
data <- inner_join(TSLA_Score, TSLA, by="Date")
```

4. Use the 'TSLA__Score.csv' file to create an open-to-close cumulative return series for days with S-Socre greater than or equal to 2

filter the return with s score $\geqslant 2$

```
colnames(data)[colnames(data)=="s-score"] <- "sscore"

#return 0 to return value(which is same as returning 1 to rate)
#if s-score is greater than or equal to 2
data$rate_big_s <- ifelse(data$sscore>=2,data$rate,1)
data$cum_return_big_s <- cumprod(data$rate_big_s)-1

positive <- ggplot(data = data, aes(x = data$Date, y = data$cum_return_big_s)) +
               geom_line(color = "green", alpha=0.5) +
               ggtitle("Cumulative Return for TESLA when s-score is greater than 2") +
               xlab("Date") +
               ylab("Cumulative Return") +
               theme_minimal()

print(positive)
```

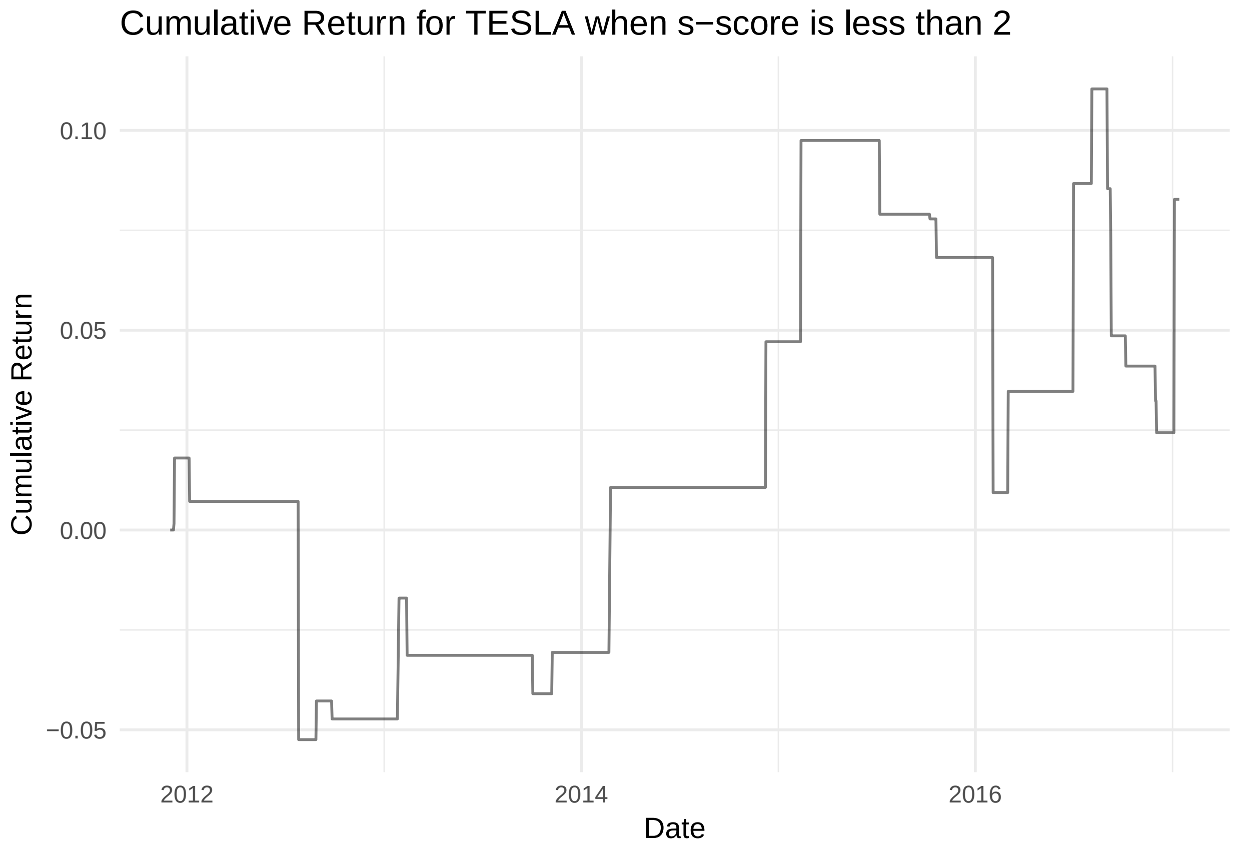


```
ggsave(file="2-4.pdf", width=15, height=7, dpi=400)
```

5. Repeat question 4 with the case of S-Score less than or equal to -2. Plot the results.

filter the return with s score ⩽ -2

```
#return 0 to return value(which is same as returning 1 to rate) if s-score is less than or equal to -2
data$rate_small_s <- ifelse(data$sscore<=-2,data$rate,1)
data$cum_return_small_s <- cumprod(data$rate_small_s)-1

negative <- ggplot(data = data, aes(x = data$Date, y = data$cum_return_small_s)) +
              geom_line(color = "black", alpha=0.5) +
              ggtitle("Cumulative Return for TESLA when s-score is less than 2") +
              xlab("Date") +
              ylab("Cumulative Return") +
              theme_minimal()

print(negative)
```

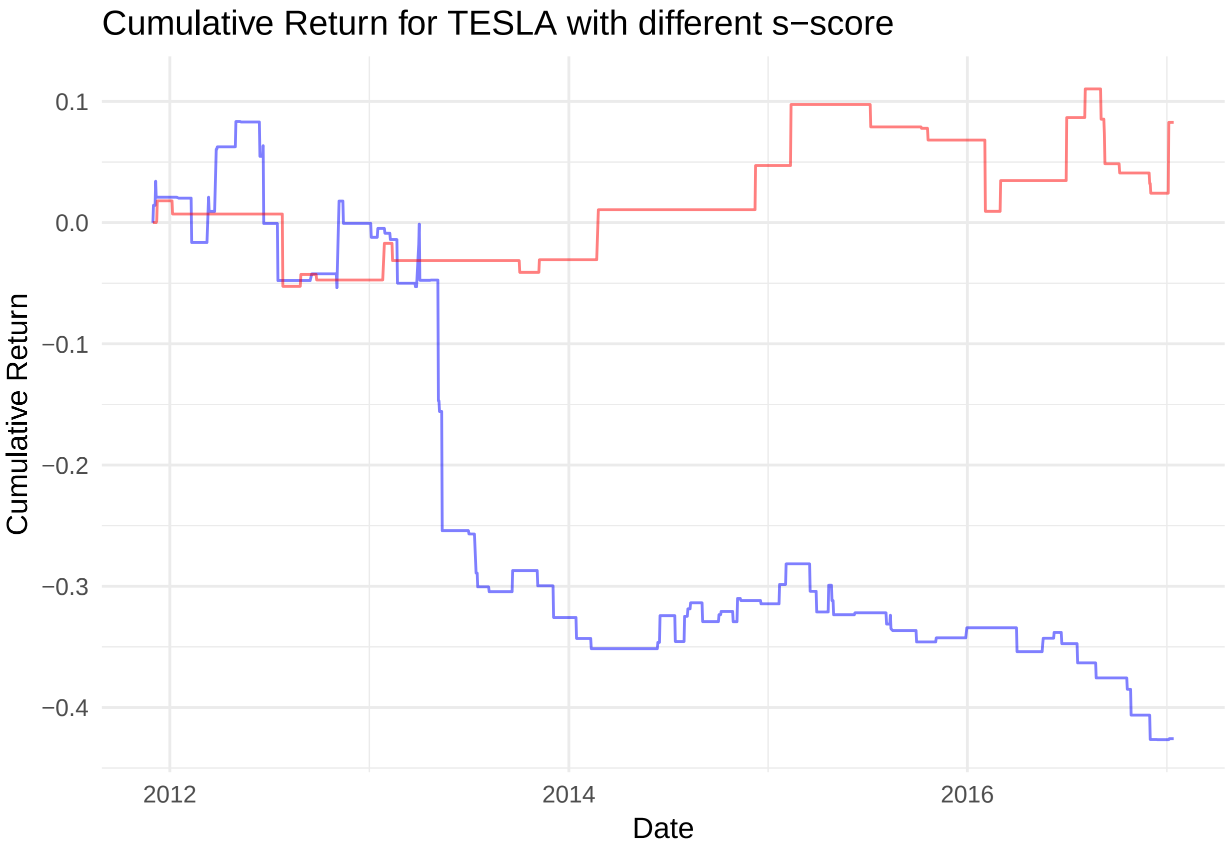


```
ggsave(file="2-5.pdf", width=15, height=7, dpi=400)
```

Section 3: Conclusions for Tesla S-Score

6. Plot results from questions 4 and 5 on the same graph. Comment on your findings.

```
pos_neg <- ggplot() +
      geom_line(data = data, aes(x = data$Date, y = data$cum_return_big_s), color = "blue",alpha=0.5)+
      geom_line(data = data, aes(x = data$Date, y = data$cum_return_small_s), color = "red", alpha=0.5)
      ggtitle("Cumulative Return for TESLA with different s-score") +
      xlab("Date") +
      ylab("Cumulative Return") +
      theme_minimal()

print(pos_neg)
```



```
ggsave(file="3-6.pdf", width=15, height=7, dpi=400)
```

This s-score index does not help us predict cumulative return plot in this given situation for the following reasons. First, there is a substantial drop in s-score and return value(if we look at Open-to-Close plot for TSLA). Because of this, daily cumulative return when s $\geqslant 2$ has been low after 2013. Additionally, according to the raw data set, all of s-score is gotten at 14:10. If this score is gotten after 14:10, it would produce different outcome to better predict daily return value since social media is supposed to get active during the day-time to night-time. Therefore, S-score would be based on more information about social media content of both positive or negative. Futher research would be how s-volume is correlated to s-score or even daily return value.

7. Present a histogram of S-Score values for TSLA along with its summary statistics. Comment on your findings.

Create S-score distribution

```
summary(data$sscore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -3.9280 -0.5517 -0.0060  0.1541  0.7370  4.1920
```

```
mean(data$sscore)
```

```
## [1] 0.1540926
```

```
sd(data$sscore)
```

```
## [1] 1.115704
```

```
pnorm(2.0, mean = mean(data$sscore), sd = sd(data$sscore), lower.tail = TRUE)
```

```
## [1] 0.9509848
```

```
qnorm(0.95,mean = mean(data$sscore), sd = sd(data$sscore), lower.tail = TRUE)
```

```
## [1] 1.989262
```

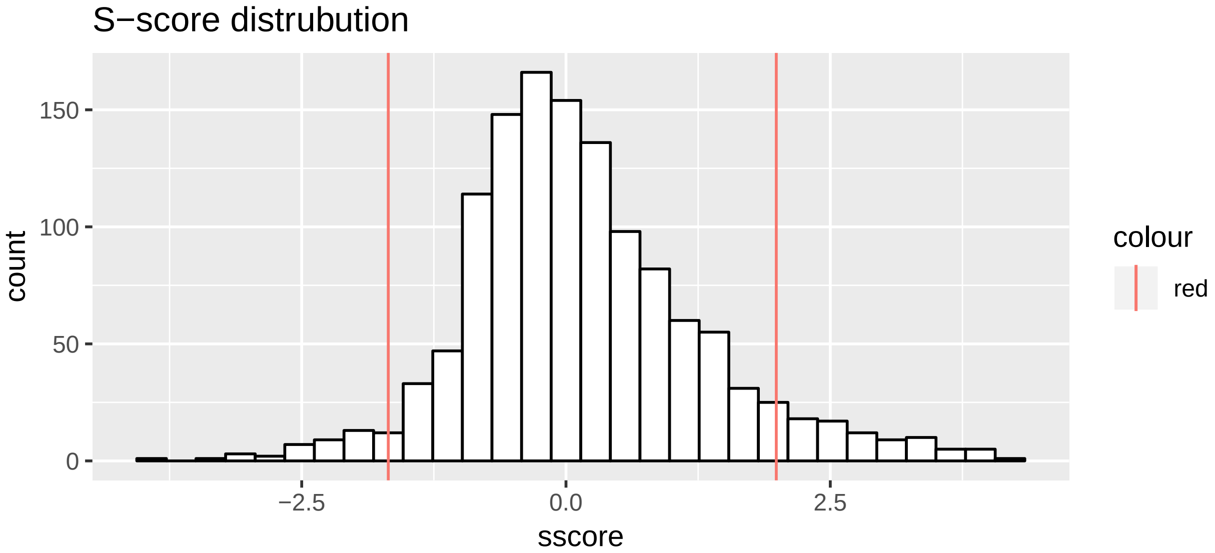
```
pnorm(-2.0, mean = mean(data$sscore), sd = sd(data$sscore), lower.tail = TRUE)
```

```
## [1] 0.02675988
```

```
qnorm(0.05,mean = mean(data$sscore), sd = sd(data$sscore), lower.tail = TRUE)
```

```
## [1] -1.681076
```

```
shist<-ggplot(data=data, aes(x=sscore)) +
       ggtitle("S-score distrubution") +
       geom_histogram(color="black", fill="white") +
       geom_vline(aes(xintercept=1.989262,col="red")) +
       geom_vline(aes(xintercept=-1.681076,col="red"))

shist
```



```
ggsave(file="3-7.pdf", width=15, height=7, dpi=400)
```

It is slightly skewed to the right but it is plausible to claim that s-score is normally distributed. With bootstrap or more samples, we could attain better normal distribution of S-score.

It has median at s-score = -0.006, mean at s-score = 0.1540926, and standard deviation σ = 0.9509848. We can treat just like Z-score. S-score with 2 means that current conversation is more positive than 95 % of prior conversation. S-score with -2 means that current conversation is more negative than 95% of prior conversation.

Section 4: Additional Analysis of Tesla S-Score

8. Create a weekly trading model for TSLA using S-Score. Accumulate and plot its return.

```
#read TSLA.csv
TSLA2 <- read_csv("SMA Intern Test Set/TSLA.csv")

#change date type from character to date
TSLA2$Date <- as.Date(TSLA2$Date,tryFormats = "%m/%d/%Y")

#Change the date (oldest to latest)
TSLA2 <- TSLA2[order(as.Date(TSLA2$Date, "%m/%d/%Y"), decreasing = FALSE),]

#get the week number
TSLA2$week_day <- format(TSLA2$Date, "%u")
TSLA2$which_week <- strftime(TSLA2$Date, "%Y_%V")


#make sure we do not have Saturday and Sunday
sum(TSLA2$week_day == 6)
```

```
## [1] 0
```

```
sum(TSLA2$week_day == 7)
```

```
## [1] 0
```

Therefore, this tells us that we do not incude Saturday and Sunday in this data set. Next step is lag the rows in the data frame by 5(we know this data set only includes market days).

```
TSLA2$lag_close <- c(NA, TSLA2$Adj_Close[1:nrow(TSLA2)-1])
TSLA2$lag_close[1] <- TSLA2$Adj_Close[1]
TSLA2$Return <- (TSLA2$Adj_Close - TSLA2$lag_close)/TSLA2$lag_close
TSLA2$Rate <- TSLA2$Return + 1

#data <- data %>% group_by(nweek) %>% mutate(wr = prod(rate))

#TSLA2 <- TSLA2 %>% group_by(which_week) %>% mutate(wreturn = prod(Rate))

#Data <- TSLA2 %>%
#mutate(open = Adj_Open, close = Adj_Close, lag.close = lag_close, rate = Rate, nweek = which_week)
#%>% select(Date, open, close, lag.close, rate, nweek)
#data_TSLA <- write.csv(Data, "TSLA.csv")
#TSLA4 <- read_csv("TSLA.csv")
```

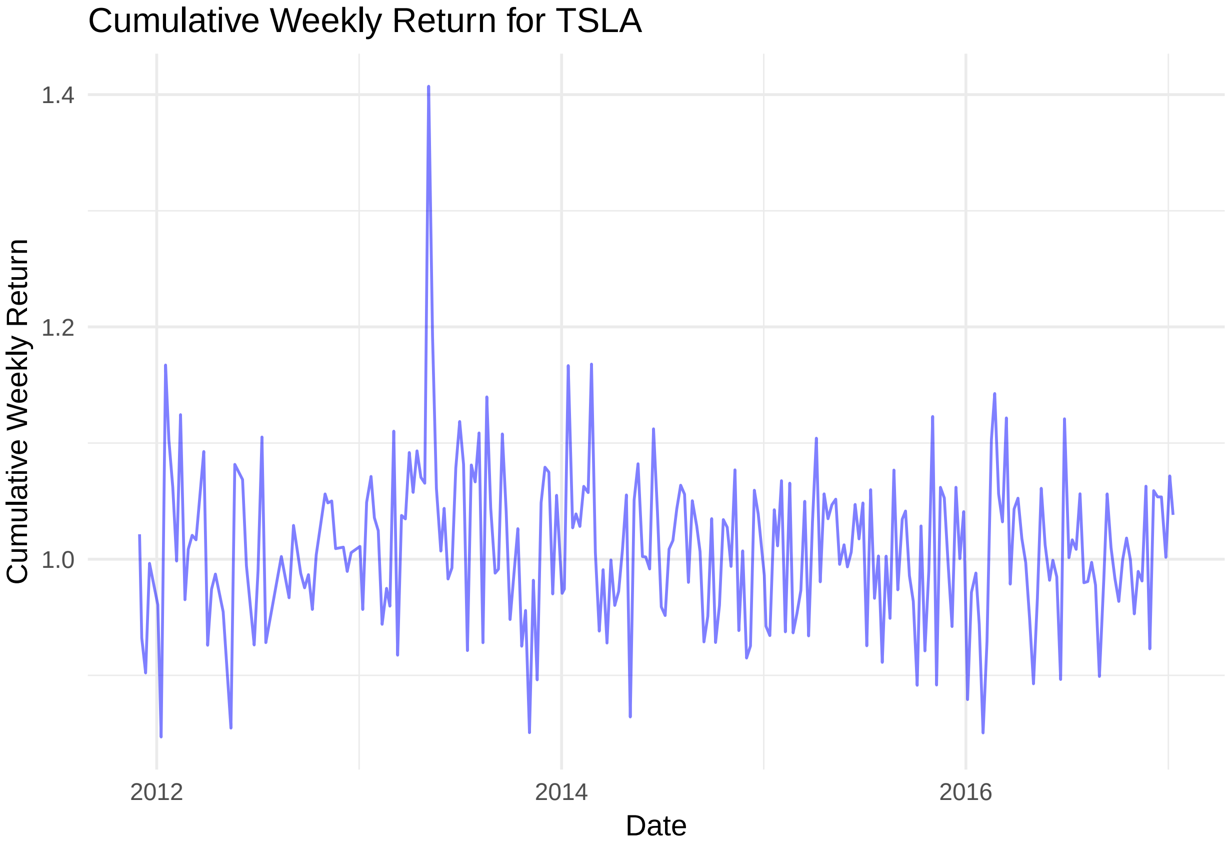
wr variable returns bizarre value and I spend a lot of time trouble shooting, but I could not figure it out.

Since I doubted my version of R or Rstudio technical problem, I use exactly the same code with my friend's Rstudio, and it worked out. Therefore, I used my friend's Rstudio with exactly the same code listed above with #line.

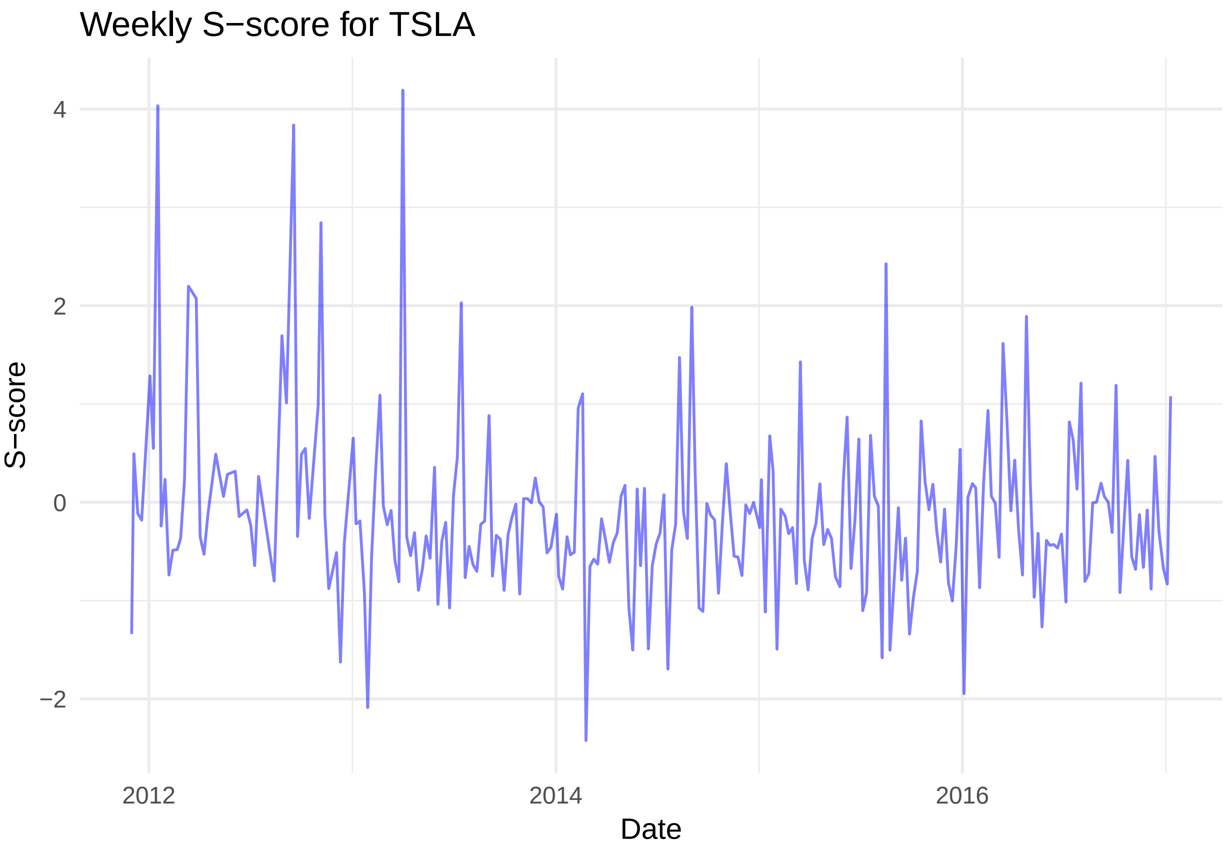
```
#import the previous data set
data5 <- read_csv("data_realgood.csv")
data5$X1 <- NULL
```

```
#only get one distinct value(since we got five consecutive return values Monday to Friday)
data5 <- subset(data5, duplicated(data5$nweek) != TRUE)
colnames(TSLA_Score)[colnames(TSLA_Score)=="s-score"] <- "sscore"

#inner join score data
d2 <- inner_join(TSLA_Score, data5, by="Date")
```

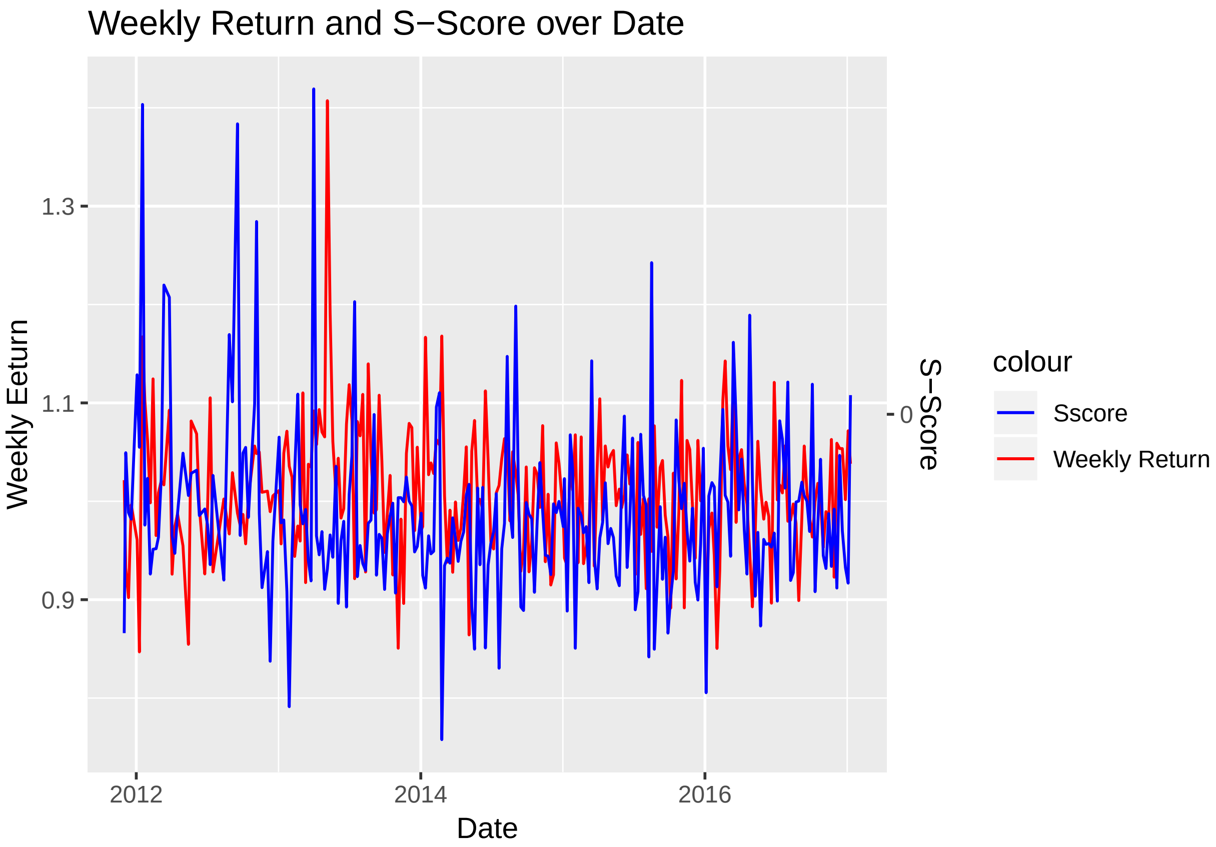
```
#plot cumulative weekly return
z <- ggplot(data = d2, aes(x = d2$Date, y = d2$wr))+
      geom_line(color = "blue", alpha=0.5) +
      ggtitle("Cumulative Weekly Return for TSLA") +
      xlab("Date") +
      ylab("Cumulative Weekly Return") +
      theme_minimal()

print(z)
```



```
ggsave(file="4-8-1.pdf", width=15, height=7, dpi=400)

#plot weekly S-Score
s_weekly <- ggplot(data = d2, aes(x = d2$Date, y = d2$sscore))+
            geom_line(color = "blue", alpha=0.5) +
            ggtitle("Weekly S-score for TSLA") +
            xlab("Date") +
            ylab("S-score") +
            theme_minimal()

print(s_weekly)
```



```
ggsave(file="4-8-2.pdf", width=15, height=7, dpi=400)

#plot them in the same graph
gg <- ggplot(d2, aes(x = d2$Date))+
  geom_line(aes(y = d2$wr, colour = "Weekly Return"))+
  geom_line(aes(y = 1+ d2$sscore/10, colour = "Sscore")) +
  ggtitle("Weekly Return and S-Score over Date") +
  scale_y_continuous(sec.axis = sec_axis(~.*0, name = "S-Score")) +
  scale_colour_manual(values = c("blue", "red")) +
  labs(y = "Weekly Eeturn", x = "Date")


print(gg)
```



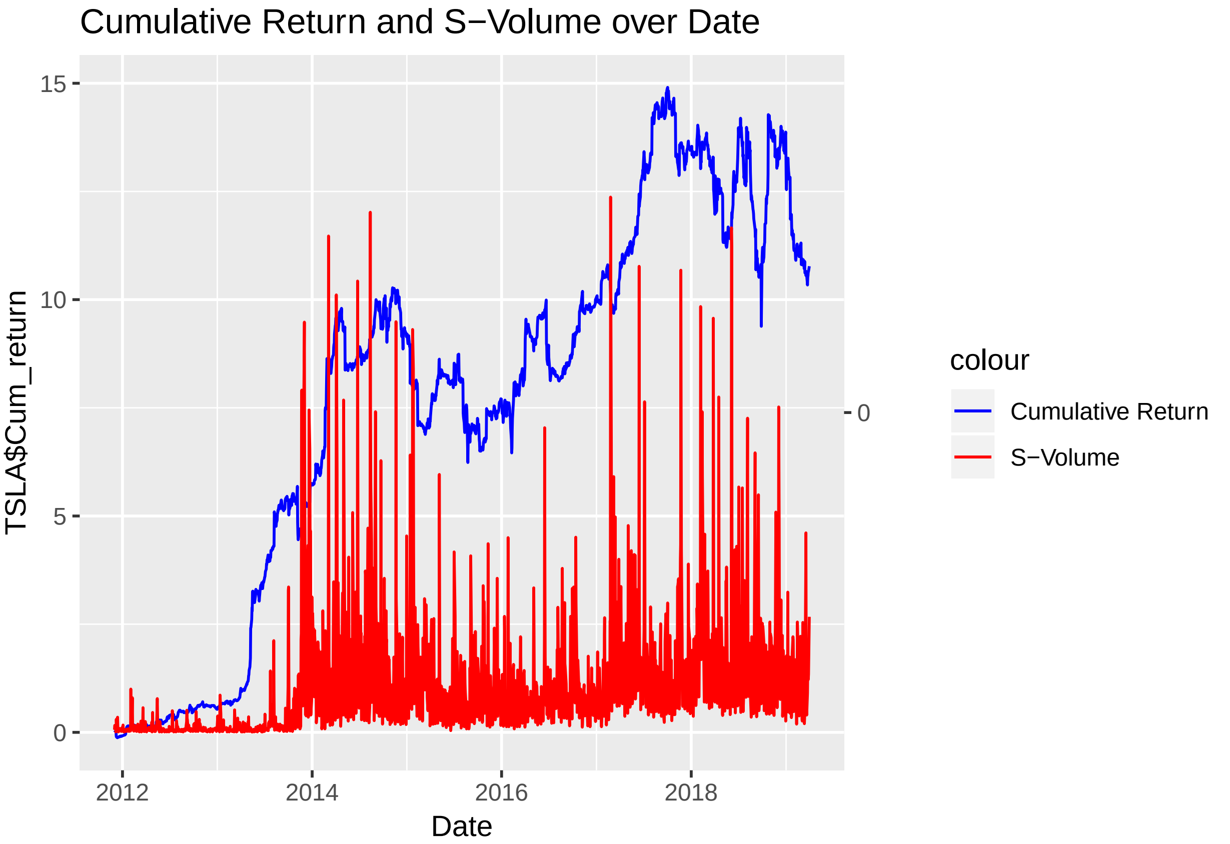
```
ggsave(file="4-8-3.pdf", width=15, height=7, dpi=400)
```

There is not clear correlation between S-Score and Weekly Return, however, there might be a high chance of correlation between change in S-Score and Weekly Return because when S-Score increases, we could see the big increase in Weekly Return right after(example is if we take a look at the beginning of 2013). Even in general, there is few time when Weekly Return got lower than 1 right after big S-Score. Because of this lag in Weekly Return increase, we suggest a trading model to buy stock when S-Score gets high($\geqslant 2$) and sell them the week after.

9. Combine 'S-Score' and another factor (S-Volume, SV-Score, or something else) to create a trading strategy. This trading strategy does not have to be profitable and can have a daily, weekly, or monthly holding period. Please explain your rationale behind choosing this strategy.

Since Tesla is a tech automotive company which utilizes the power of social media such as Twitter, we hypothesized that the hotness(social media activity) is a good indication to predict the return ratio. Therefore, the first graph is using S-Volume which is the number of indicative tweets.

```
colnames(TSLA_Score)[colnames(TSLA_Score)=="s-volume"] <- "svolume"
colnames(TSLA_Score)[colnames(TSLA_Score)=="sv-score"] <- "svscore"
colnames(TSLA_Score)[colnames(TSLA_Score)=="s-delta"] <- "sdelta"

gf <- ggplot(TSLA, aes(x = TSLA$Date))+
  geom_line(aes(y = TSLA$Cum_return, colour = "Cumulative Return"))+
  geom_line(aes(y = TSLA_Score$svolume/100, colour = "S-Volume")) +
  ggtitle("Cumulative Return and S-Volume over Date") +
  scale_y_continuous(sec.axis = sec_axis(~.*0, name = "")) +
  scale_colour_manual(values = c("blue", "red")) +
  labs(x = "Date")

print(gf)
```



```
ggsave(file="4-9-1.pdf", width=15, height=7, dpi=400)
```

S-Volume is useful to predict the cumulative return, however, since it only takes the positive values, we could also need to combine different indicators that takes both positive and negative values.

We also assumed that few number of S-volume indicates inactivitiy in social media and it is hard to use it for predicting cumulative return. Therefore, we filtered S-Volume less than 10.

```
TSLA5 <- subset(TSLA_Score, svolume >= 10)
```

```
#inner join by date
newTS <- inner_join(TSLA5, TSLA, by="Date")
```

Since we want to know what predicts the cumulative return better, we create the linear model using multiple interesting indicators. We also take logarithm to the value of S-Volume since it only takes positive values.

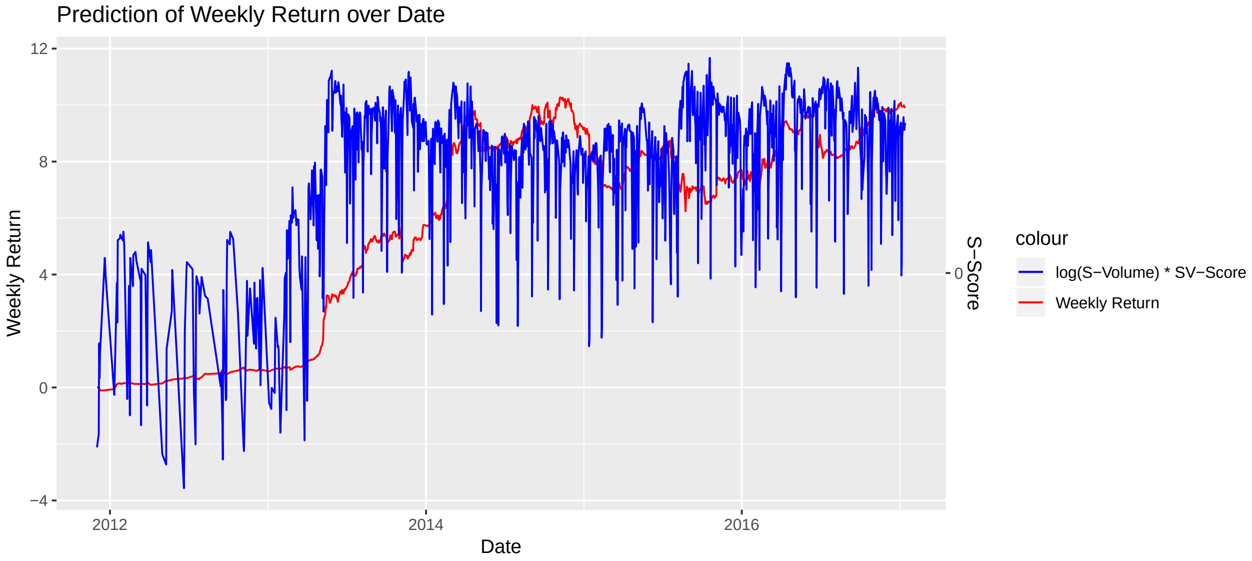
```
lm <- lm(data = newTS, Cum_return ~ sscore + log(svolume) + svscore + sdelta)
summary(lm)
```

```
##
## Call:
## lm(formula = Cum_return ~ sscore + log(svolume) + svscore + sdelta,
##     data = newTS)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4093 -1.7949  0.3093  1.9177  4.9772
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.68392    0.37370  -4.506 7.33e-06 ***
## sscore        0.03260    0.06876   0.474    0.635
## log(svolume)  1.89805    0.08016  23.678  < 2e-16 ***
## svscore      -1.25877    0.07573 -16.622  < 2e-16 ***
## sdelta        0.14129    0.36262   0.390    0.697
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.352 on 1070 degrees of freedom
## Multiple R-squared:  0.3756, Adjusted R-squared:  0.3733
## F-statistic: 160.9 on 4 and 1070 DF,  p-value: < 2.2e-16
```

As we could see S-Volume and SV-Score could better explain the whole cumulative return of TSLA.

```r
gg <- ggplot(newTS, aes(x = newTS$Date))+
geom_line(aes(y = newTS$Cum_return, colour = "Weekly Return"))+
geom_line(aes(y = 1.89*log(newTS$svolume) - 1.26*newTS$svscore /0.5, colour = "log(S-Volume) * SV-Score
ggtitle("Prediction of Weekly Return over Date") +
scale_y_continuous(sec.axis = sec_axis(~.*0, name = "S-Score")) +
scale_colour_manual(values = c("blue", "red")) +
labs(y = "Weekly Return", x = "Date")

print(gg)
```



```r
ggsave(file="4-9-2.pdf", width=15, height=7, dpi=400)
```

Blue line successfully explains both increase and decrease of cumulative return. However, this trading strategy is profitable when volatility of cumulative return is big(after 2013). When the value of cumulative return steadily increased in 2012, the value of blue line went back and forth.

In conlcusion, the trading model we suggest is to create a linear model with multiple variables in TSLA_Score to better predict return fluctuation.

Our arranged indicator is below:

$$S_{new} = 1.89log(S_{volume}) * 0.07SV_{score}$$

With this indication, we could establish a new trading strategy to better predict the price change.