Assignment 01, Question 3

Jeff Nguyen

February 5, 2021

University of Southern California
Marshall School of Business
FBE 506 Quantitative Method in Finance

Student Name: Ngoc Son (Jeff) Nguyen

Question 3

To test whether the pre-split changes in the stock price of TSLA was due to random fluctuations of the market due to the leak of stock split information, we complete an event study statistical analysis. Our assumption is that the stock market is efficient and thus, stock prices adjust quickly to news and information in the market. Steps:

1. Download data from Yahoo Finance. The following tickers are downloaded: TSLA, GSPC (representing the S&P 500).
2. Define two sets of data for Estimation Window and Event Window. The event happens on August 12, 2020.
3. Run a linear regression using Estimation Window.
4. Do an ex-post forecast of TSLA daily returns using data for the event period.
5. Compute the test statistics: abnormal return (AR) and the t-test statistics on the significance of AR
6. Concluding remarks

Event Study

1. Downloading data:

```
library(quantmod)
```

```
## Loading required package: xts

## Loading required package: zoo

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

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

## Loading required package: TTR

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

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
# Set start date and end date of data
start_date <- "2019-08-01"
end_date <- "2020-10-31"
event_date <- "2020-08-12"

# Get data
getSymbols("TSLA", src = "yahoo", from = start_date, to = end_date)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "TSLA"
```

```
getSymbols("^GSPC", src = "yahoo", , from = start_date, to = end_date) # S&P 500
```

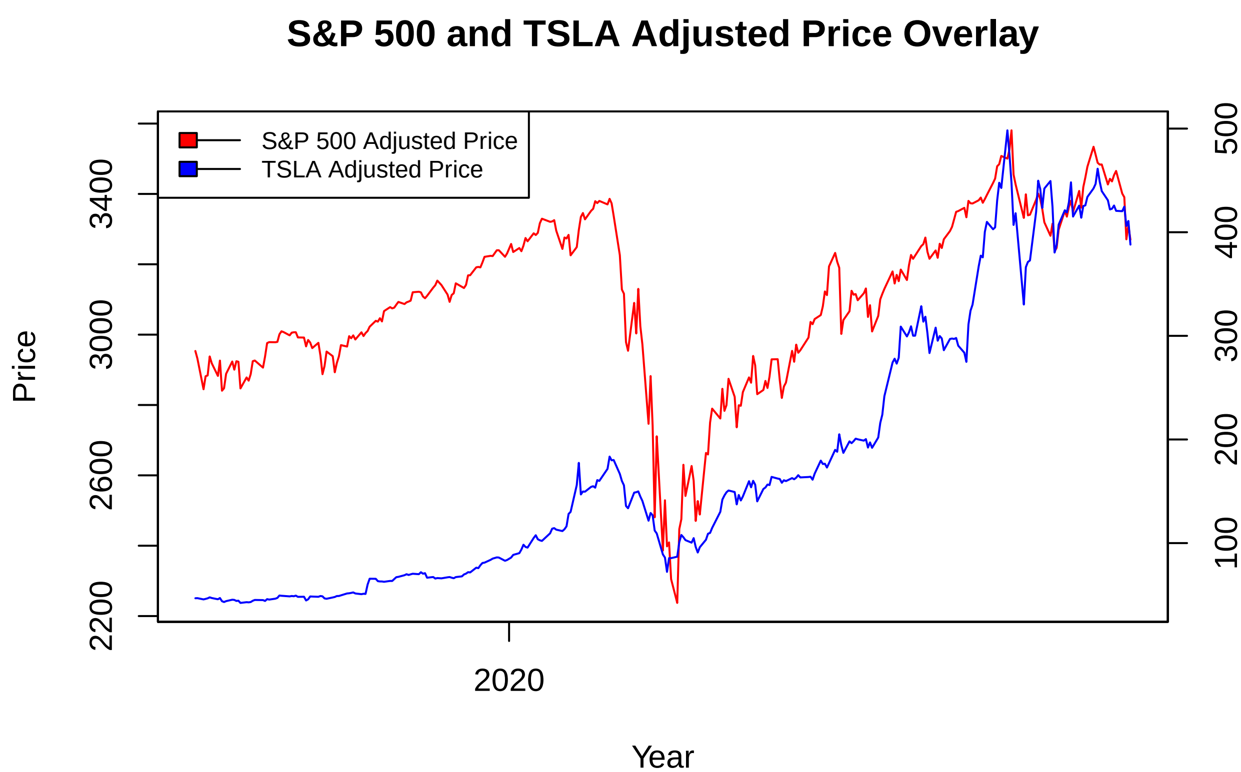
```
## [1] "^GSPC"
```

```
# Get adjusted returns data
rTSLA <- diff(log(TSLA$TSLA.Adjusted))
rGSPC <- diff(log(GSPC$GSPC.Adjusted))
```

Observe TSLA and S&P 500 for the period:

```
# Initialize xts objects contain adjusted price for S&P 500 and AAPL and merge
gspc_xts <- as.xts(GSPC[,"GSPC.Adjusted"])
tsla_xts <- as.xts(TSLA[,"TSLA.Adjusted"])
price_compare <- merge.xts(gspc_xts, tsla_xts)

# Graph monthly AAPL and monthly S&P500 on one coordinate system
# Plot S&P 500
plot(as.zoo(price_compare[, "GSPC.Adjusted"]), screens = 1, main = "S&P 500 and TSLA Adjusted Price Ove
     xlab = "Year", ylab = "Price", col = "Red")
```

```
# Keep working on the same plot
par(new = TRUE)

# Plot AAPL and suppress axis value
plot(as.zoo(price_compare[, "TSLA.Adjusted"]), screens = 1, xaxt = "n", yaxt = "n", xlab = "", ylab = "

# Add right-handed axis to display AAPL price
axis(4)

# Add legend
legend("topleft", c("S&P 500 Adjusted Price", "TSLA Adjusted Price"), lty = 1:1, cex = 0.75, fill = c("
```



For the period, we can observe TSLA price trend tracks that of the S&P 500 closely with S&P500 experience more volatility at the March 2020 global pandemic shutdown.

2. Define two set of data:

```
# Data File
data_ret <- data.frame(rTSLA, rGSPC)
colnames(data_ret) <- c('rTSLA', 'rGSPC')

# Define two set of data
data_ret_estimation <- data_ret[4:255,]
cat("The number of trading day in estimation data set is", nrow(data_ret_estimation))
```

```
## The number of trading day in estimation data set is 252
```

```r
data_ret_test <- data_ret[256:266,]
cat("The number of trading day in test data set is", nrow(data_ret_test))
```

```
## The number of trading day in test data set is 11
```

```r
data_ret_test
```

```
##                   rTSLA         rGSPC
## 2020-08-05 -0.001332414  0.0064091605
## 2020-08-06  0.003065917  0.0064071258
## 2020-08-07 -0.025063401  0.0006328292
## 2020-08-10 -0.023781467  0.0027384655
## 2020-08-11 -0.031639320 -0.0080010556
## 2020-08-12  0.123311216  0.0138995025
## 2020-08-13  0.041722100 -0.0020492727
## 2020-08-14  0.018162213 -0.0001718974
## 2020-08-17  0.106187678  0.0027061773
## 2020-08-18  0.027642787  0.0023007404
## 2020-08-19 -0.004546417 -0.0044141232
```

3. Run a linear regression using Estimation Window:

We attempt to fit an equation of a line: $Y = \alpha_{Jensen} + \beta * X + \epsilon$

Hypothesis for regression:

$$
\begin{aligned}
H_0 : \alpha &= 0 \\
H_a : \alpha &\neq 0 \\
&\text{and} \\
H_0 : \beta &= 0 \\
H_a : \beta &\neq 0
\end{aligned} \tag{1}
$$

```r
reg <- lm(rTSLA~rGSPC, data=data_ret_estimation)

summary(reg)
```

```
##
## Call:
## lm(formula = rTSLA ~ rGSPC, data = data_ret_estimation)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.208404 -0.018885 -0.003262  0.019352  0.166171
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.006730   0.002728   2.467   0.0143 *
## rGSPC       1.181994   0.128159   9.223   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04329 on 250 degrees of freedom
## Multiple R-squared:  0.2539, Adjusted R-squared:  0.2509
## F-statistic: 85.06 on 1 and 250 DF,  p-value: < 2.2e-16
```

The estimated equation is $rTSLA = .006730 + 1.181994 * rGSPC$, where the p_{value} for the intercept $.0143 < .05$.
Therefore, we reject the null hypothesis at 95% confidence level that the intercept α_{Jensen} statistically is no different from 0. Thus, we reject the null hypothesis $H_0 : \alpha = 0$ and accept the null hypothesis $H_a : \alpha \neq 0$.

Similarly, the p_{value} for β is $2e - 16 < .05$, implying that the coefficient β statistically is significant at 95% or more, and we reject the null hypothesis $H_0 : \beta = 0$ and accept the alternative hypothesis $H_a : \beta \neq 0$.

Goodness of Fit:
Through inspection of the linear regression result, we observe the $R^2 = .2539$ value to not be close to 1 at all. $R^2 = .2539$ implies that 25.39% of the variations in daily returns of TSLA is explained by that or the S&P 500.

Standard Error of Regression:
We can see that the Standard Error of Regression is $S.E. = .04329$.

We have the mean daily returns of TSLA for the period to be $mean(na.omit(rTSLA))$.

From this, we can calculate the forecasting efficiency statistic to be:

$$\begin{aligned} \frac{S.E.}{\overline{Y}} =& \frac{.04329}{0.0066958} \\ =& 646.53\% > 10\% \end{aligned} \tag{2}$$

This statistic implies that the linear model is not a good forecasting model.

Thus, upon exploring the goodness of fit and standard error of regression, we can see that daily returns of TSLA and S&P500 do not follow a linear model.

4. Do an ex-post forecast of TSLA daily returns using data for the event period.

```
library(forecast)

pred <- predict(reg, newdata=data_ret_test, se.fit = TRUE)
pred
```

```
## $fit
##   2020-08-05   2020-08-06   2020-08-07   2020-08-10   2020-08-11   2020-08-12
##  0.014305628  0.014303223  0.007478041  0.009966890 -0.002727156  0.023159165
##   2020-08-13   2020-08-14   2020-08-17   2020-08-18   2020-08-19
##  0.004307814  0.006526860  0.009928726  0.009449502  0.001512576
##
## $se.fit
##  2020-08-05  2020-08-06  2020-08-07  2020-08-10  2020-08-11  2020-08-12
## 0.002826811 0.002826742 0.002726905 0.002740679 0.002941122 0.003215977
##  2020-08-13  2020-08-14  2020-08-17  2020-08-18  2020-08-19
## 0.002747909 0.002728681 0.002740268 0.002735630 0.002801504
##
## $df
```

```
## [1] 250
##
## $residual.scale
## [1] 0.04328822
```

5. Compute the test statistics: abnormal return (AR) and the t-test statistics on the significance of AR

Calculate abnormal return statistics:

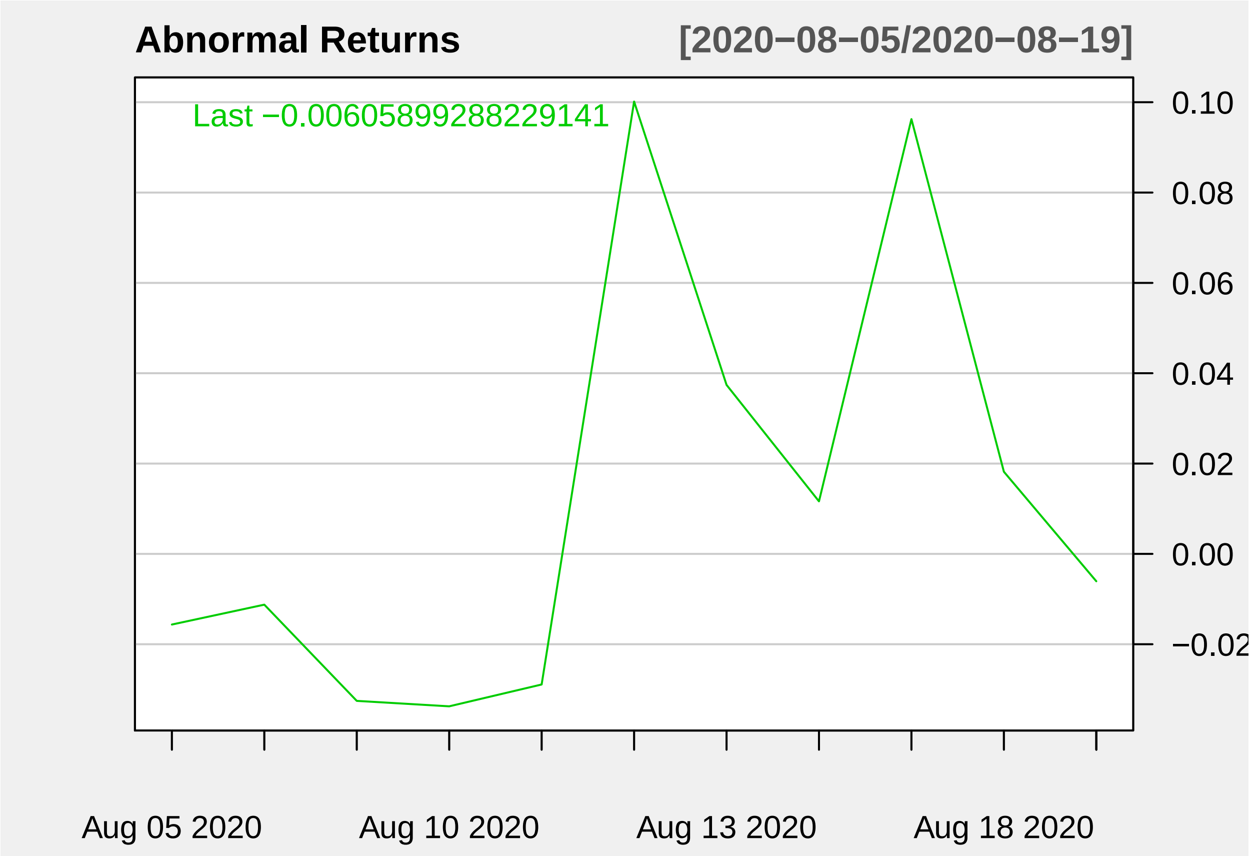
```r
#Find Abnormal Return
ar_data <- cbind(data_ret_test$rTSLA,  pred$fit)
ar_data
```

```
##                    [,1]         [,2]
## 2020-08-05 -0.001332414  0.014305628
## 2020-08-06  0.003065917  0.014303223
## 2020-08-07 -0.025063401  0.007478041
## 2020-08-10 -0.023781467  0.009966890
## 2020-08-11 -0.031639320 -0.002727156
## 2020-08-12  0.123311216  0.023159165
## 2020-08-13  0.041722100  0.004307814
## 2020-08-14  0.018162213  0.006526860
## 2020-08-17  0.106187678  0.009928726
## 2020-08-18  0.027642787  0.009449502
## 2020-08-19 -0.004546417  0.001512576
```

```r
ar <- data_ret_test$rTSLA - pred$fit
ar
```

```
##   2020-08-05   2020-08-06   2020-08-07   2020-08-10   2020-08-11   2020-08-12
## -0.015638042 -0.011237306 -0.032541442 -0.033748357 -0.028912164  0.100152051
##   2020-08-13   2020-08-14   2020-08-17   2020-08-18   2020-08-19
##  0.037414286  0.011635353  0.096258952  0.018193286 -0.006058993
```

```r
ar_plot <- data.frame(ar)
colnames(ar_plot) <- "AR"
chartSeries(ar_plot, name="Abnormal Returns", type='line', theme=chartTheme("white"))
```



Do a t-test for the significance of the abnormal returns:

```
T_Data <- cbind(ar, pred$se.fit)
T_test <- data.frame(ar/pred$se.fit)
T_test
```

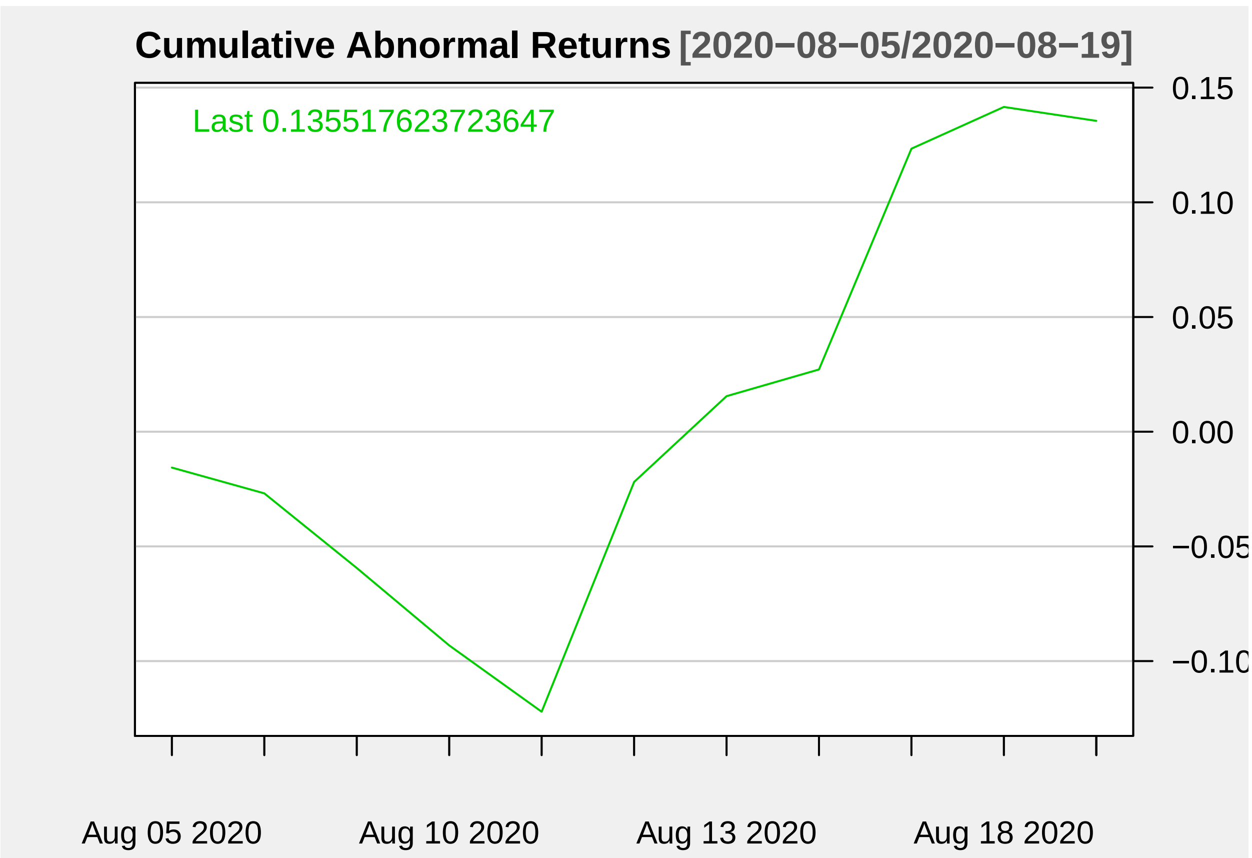
```
##            ar.pred.se.fit
## 2020-08-05      -5.532044
## 2020-08-06      -3.975356
## 2020-08-07     -11.933469
## 2020-08-10     -12.313868
## 2020-08-11      -9.830319
## 2020-08-12      31.142030
## 2020-08-13      13.615548
## 2020-08-14       4.264094
## 2020-08-17      35.127574
## 2020-08-18       6.650491
## 2020-08-19      -2.162764
```

Plot the cummulative abnormal return statistics to examine the impact of the event and the efficiency of the market:

```
cum <- data.frame(cumsum(ar))
colnames(cum) <- "CAR"
cum
```

```
##                      CAR
## 2020-08-05 -0.01563804
## 2020-08-06 -0.02687535
## 2020-08-07 -0.05941679
## 2020-08-10 -0.09316515
## 2020-08-11 -0.12207731
## 2020-08-12 -0.02192526
## 2020-08-13  0.01548903
## 2020-08-14  0.02712438
## 2020-08-17  0.12338333
## 2020-08-18  0.14157662
## 2020-08-19  0.13551762
```

```
chartSeries(cum, name="Cumulative Abnormal Returns", type='line', theme=chartTheme("white"))
```



6. Concluding Remarks

The event happens on August 12, 2020. We can observe that the event was a surprise to the market as TSLA sold off prior to the event, from Aug 5 to Aug 11 and increased in price substantially after the announcement. Thus, we conclude that the pre-split changes in the stock price was due to random fluctuations of the market and not because of the leak of the stock split information.